Technische Universität München
TUM School of Computation, Information and Technology

Adaptive Task Execution for Flexible Robotic Assembly Systems Using Intrinsic Tactile Sensing

Korbinian Nottensteiner

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. habil. Alois Christian Knoll

Prüfer*innen der Dissertation:

1. Prof. Dr.-Ing. Alin Albu-Schäffer
2. Prof. Dr. Aleš Ude

Die Dissertation wurde am 16.11.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 17.04.2023 angenommen.

Abstract

Today's challenges in production motivate the development of flexible production systems. Larger product varieties are to be manufactured in a shorter time while using resources more efficiently. The use of flexible robotic assembly systems is a promising part of the solution to meet these requirements. In this context, robotic assembly systems have to deal with the uncertainty about which products are demanded in the future and the uncertainties during the actual execution of the physical manufacturing process. In this work, a planning and execution framework for such a flexible robotic assembly system is proposed, which reduces manual effort throughout the overall process chain and enables the efficient manufacturing of individual products. The framework accepts specifications of desired products as input and manufactures them without any further manual steps. An assembly planner is integrated, which finds suitable task sequences. In order to solve the tasks, robotic skills are developed that can be mapped to particular tasks in the sequences with a task classification approach. These skills are implemented in a reusable and object-centric fashion such that they can be applied in multiple situations. The manufacturing of one-of-a-kind products is successfully demonstrated for structures made out of aluminum profiles with a varying number of parts and various geometric configurations.

In general, the fact that each product has individual properties makes it impossible to optimize for all possible cases beforehand. Therefore, the skills need to be able to deal online with the present uncertainties. Here, the impedance-based compliant control of the considered lightweight robot enables a robust and reliable assembly in the presence of uncertainties. Additionally, in this work, it is demonstrated how intrinsic tactile sensing using joint torque measurements enables an improved observation of the execution. In particular, a particle filtering method is implemented with a focus on part localization during insertion tasks. It is shown how heuristics inspired by probabilistic roadmap planning can improve observation performance in such contact-rich tasks. The approach uses explicit models of the kinematics and the contacts and thereby is suitable for industrial settings in which structured information is available. A contact model known from haptic rendering is integrated, which is able to deal with complex and non-convex geometries. Furthermore, methods for tracking moving objects are proposed and evaluated, namely a constraint-based propagation model and a constant velocity model. The developed observation methods allow the accurate online estimation of uncertainties and thus enable the adaption of the task execution. For this purpose, an adaptive motion generation algorithm is developed, which can be used to execute an assembly strategy with respect to the currently estimated part poses. The motion generator is demonstrated with an object-centric peg-in-hole skill. A further method developed in this work takes into account the local shape of the relative configuration space between the parts to select the next best motion toward the task goal. Overall, the methods for adaptive task execution developed in this work enable reusing skills in various situations with uncertainties as required for flexible robotic assembly.

Zusammenfassung

Die heutigen Herausforderungen in der Produktion motivieren die Entwicklung flexibler Produktionssysteme. Eine größere Produktvielfalt soll in kürzerer Zeit und mit effizienterem Ressourceneinsatz hergestellt werden. Der Einsatz von flexiblen Robotermontagesystemen ist ein vielversprechender Teil der Lösung um diese Anforderungen zu erfüllen. In diesem Zusammenhang müssen Robotermontagesysteme mit der Unsicherheit umgehen können, welche Produkte in der Zukunft nachgefragt werden, und mit Unsicherheiten bei der tatsächlichen Ausführung des physischen Fertigungsprozesses. In dieser Arbeit wird ein Framework zur Planung- und Ausführung für ein solches flexibles Robotermontagesystem vorgeschlagen, welches den manuellen Aufwand in der gesamten Prozesskette reduziert und die effiziente Herstellung individueller Produkte ermöglicht. Das Framework akzeptiert die Spezifikationen der gewünschten Produkte als Eingabe und fertigt diese ohne weitere manuelle Schritte zu erfordern. Es ist ein Montageplaner integriert, der geeignete Reihenfolgen der Aufgabenausführung ermittelt. Zur Lösung der Aufgaben werden Roboterfertigkeiten entwickelt, die mit einem Klassifizierungsansatz auf bestimmte Aufgaben in den Sequenzen zugewiesen werden können. Diese Fertigkeiten sind wiederverwendbar und objektzentriert implementiert, so dass sie in verschiedenen Situationen eingesetzt werden können. Die Herstellung individueller Produkte wird erfolgreich für Strukturen aus Aluminiumprofilen mit einer unterschiedlichen Anzahl von Teilen und verschiedenen geometrischen Konfigurationen demonstriert.

Die Tatsache, dass jedes Produkt individuelle Eigenschaften besitzt, macht es im Allgemeinen unmöglich, eine Optimierung für alle möglichen Fälle in Voraus durchzuführen. Daher müssen die Fertigkeiten in der Lage sein, während der Ausführung mit den vorhandenen Unsicherheiten umzugehen. Hier ermöglicht die impedanzbasierte Nachgiebigkeitsregelung des verwendeten Leichtbauroboters eine robuste und zuverlässige Montage in Gegenwart von Unsicherheiten. Darüber hinaus wird in dieser Arbeit gezeigt, wie eine intrinsische taktile Sensorik, realisiert durch Drehmomentmessungen in den Gelenken, eine verbesserte Beobachtung der Ausführung ermöglicht. Insbesondere wird eine Methode mit einem Partikelfilter implementiert, die sich auf die Lokalisierung von Bauteilen bei Fügeaufgaben konzentriert. Es wird gezeigt, wie Heuristiken, die von der probabilistischen Roadmap-Planung inspiriert sind, die Beobachtung bei solchen kontaktreichen Aufgaben verbessern können. Der Ansatz verwendet explizite Modelle der Kinematik und der Kontakte und ist damit für industrielle Einsatzumgebungen geeignet, in denen strukturierte Informationen verfügbar sind. Hierbei wird ein Kontaktmodell verwendet, welches aus dem haptischen Rendering bekannt ist, und das in der Lage ist, mit komplexen und nicht-konvexen Geometrien umzugehen. Darüber hinaus werden Methoden zum Tracking von bewegten Objekten vorgeschlagen und evaluiert, nämlich ein Systemmodell welches geometrische Randbedingungen einbezieht und ein Modell, welches konstante Geschwindigkeiten in der Bewegung annimmt. Die entwickelten Beobachtungsmethoden erlauben die genaue Online-Schätzung von Unsicherheiten und ermöglichen so die Anpassung der Aufgabenausführung. Zu diesem Zweck wird ein adaptiver Algorithmus

zur Bewegungsgenerierung entwickelt. Dieser kann verwendet werden um eine Montagestrategie unter Berücksichtigung der aktuell geschätzten Bauteilpositionen auszuführen. Der Bewegungsgenerator wird anhand einer objektzentrierten Fertigkeit für die Ausführung von Fügeaufgaben demonstriert. Eine weitere in dieser Arbeit entwickelte Methode verwendet die lokale Repräsentation des relativen Konfigurationsraums zwischen den Bauteilen, um die nächstbeste Bewegung zur Zielerreichung der Aufgabe auszuwählen. Insgesamt ermöglichen die in dieser Arbeit entwickelten Methoden zur adaptiven Aufgabenausführung die Wiederverwendung von Fertigkeiten in verschiedenen Situationen mit Unsicherheiten, wie sie für die flexible Robotermontage erforderlich sind.

Acknowledgment

This dissertation was written in parallel to my work at the Institute of Robotics and Mechatronics (RM) of the German Aerospace Center (DLR) in Oberpfaffenhofen. I would like to thank the head of the institute and supervisor of my dissertation Prof. Dr.-Ing. Alin Albu-Schäffer, the former head of autonomy and teleoperation Christoph Borst, and Dr. Freek Stulp, head of cognitive robotics, for giving me this opportunity and supporting me throughout all phases of the graduation. Furthermore, I would like to thank Prof. Dr. Aleš Ude as the second examiner, and Prof. Dr.-Ing. Alois Knoll for chairing the examination committee.

My first steps at the institute were guided by Andreas Stemmer, who introduced me to impedance-based robotics in assembly applications and provided significant input, as well as precise and analytical feedback, during the development of the topic of this thesis for which I am very grateful as this formed the basis for all further achievements. In the same line, I would like to thank Dr.-Ing. Katharina Hertkorn, who helped me formulate and concretize the concepts during the first years. Furthermore, I enjoyed very much the always constructive and inspiring exchange with Michael Kaßecker, who provided me often the motivation to go one step further in research and various project activities.

The idea related to flexible assembly of one-of-a-kind products was initially discussed and developed in the EU project *SMErobotics*. The activities at RM were coordinated by Prof. Dr.-Ing. Ulrike Thomas, Dr.-Ing. Tim Bodenmüller and Dr. Maximo A. Roa. Together with the project team, consisting of Andreas Stemmer, Daniel Seidel, Dr. Theodoros Stouraitits, Michael Kaßecker, Arne Sachtler and Andrea Schwier, the first demonstrator of the robotic assembly system had been developed, which is at the core of this dissertation and I am very thankful for having done this pioneering work together.

After completion of the project, the *SME-system* was developed further by the team of autonomous assembly, for which I got the chance to take over the lead. I would like to thank all who contributed significantly to this evolution, in particular, Ismael Rodríguez, Arne Sachtler, Michael Kaßecker, Timo Bachmann, Jan Cremer, Stefan Schneyer, and Jean-Pascal Lutze, for the great sessions and fun in the lab, developing new features together, preparing the demos, and contributing successfully to the research on flexible robotic assembly with multiple publications. Also I would like to thank Florian Schmidt, Peter Lehner, Dr.-Ing. Daniel Leidner, Sebastian Brunner, Rico Belder, Dr. Franz Steinmetz, Anja Hellings and Sergey Tarassenko for providing important components of this system.

A highlight for our team in recent years was the application of our methods in the field of space robotics, which was strongly pushed by the visions of Dr. Gerhard Grunwald and Dr. Maximo A. Roa about using robots for in-space assembly, and gave the team the possiblity to evaluate the concepts in a very interesting future field of application.

Over the years, we had the opportunity to discuss and develop the concepts for flexible robotic assembly within the application domain *Future Manufacturing* of RM. This helped to build up a common vision beyond the individual works and systems, and together we were able to detail the overall concepts including modular and flexible robotic systems, human robot collaboration, autonomous assembly and mobile manipulation. Here, I would like to thank the project and team leaders of the domain at that time, Dr.-Ing. Roman Weitschat, Oliver Eiberger, Andreas Dömel, Dr. Franz Steinmetz for the constructive discussions and setting of shared strategic goals. Also, I am looking forward to following up on many ideas as a speaker of the domain, together with the teams now coordinated by Oliver Eiberger, Peter Lehner and Thomas Eiband.

A big thank you goes to all colleagues who are working on the project *Factory of the Future* for the valuable contributions and the great team spirit during the development of the systems and the preparation of demos and events. And also many thanks to all colleagues who are always open to new concepts and project ideas in the context of the manufacturing domain and who help to successfully implement the initiatives from administration to public relations.

A special thanks goes to Florian Schmidt, Thomas Gumpert, Werner Friedl, and Christian Schöttler for their assistance in all situations where special solutions in soft- and hardware were required. Furthermore, I would like to thank Dr.-Ing. Mikel Sagardia for the support during the implementation of the contact model, which I used in this dissertation, and Dr.-Ing. Ingo Kossyk, Dr. Zoltan-Csaba Marton and Michael Neumann for the exciting excurse of using structure-borne sound as additional modality in contact sensing.

There are many colleagues whom I would like to thank for their support and who ensure high quality in research and developement through their work on the overall infrastructur, in particular with respect to software development Dr.-Ing. Tim Bodenmüller, Daniel Seth, Martin Stelzer, Jan Cremer and the IT of the institute. Beyond software, I would also like to thank the colleagues from the mechanical workshop and the electronic lab, who always found time to support the requests from our team.

In addition to all the great collaboration in research and development work, there are many colleagues I would like to thank for their friendship, great times at work, and numerous activities in our free time. These are in particular my office mates from the first years Bao-Anh, Theo, Nico and Can, and furthermore: Ribin, Rico, Philipp L., Annette and Roman, Mathilde, Markus N., Christian N., Marco, Alessandro, Hrishik, Philipp S., Jörn, Seba, Julian, Basti D. and all I had the opportunity to meet and get to know outside of the daily routines.

Finally, I would like to thank again all who supported me during finalization of the dissertation and the final presentation. In particular, I would like to thank Dr. Ribin Balachandran, Arne Sachtler, and Dr. Freek Stulp for proof-reading and commenting parts of the work.

Last but not least, many thanks to all friends and family! All this would not have been possible without the enduring support of my parents, who gave me strength and confidence to achieve my goals.

Hofhegnenberg, May 2023

Korbinian Nottensteiner

Contents

List of Acronyms

APOS	advanced parts orientation system
ASP	assembly sequence planning
CAD	computer-aided design
CF	contact formation
CV	constant velocity model
DFA	design for assembly
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.
EAS	evolvable assembly system
ESS	effective sample size
FAA	flexible assembly automation
FMS	flexible manufacturing system
HMM	hidden Markov model
HRI	human robot interaction
iTaSC	constraint-based task specification
LN	links and nodes
LWR	light-weight robot
MC	mass customization
MFD	modular function deployment
PbD	programming by demonstration
PCC	polyhedral convex cone
PRM	probabilistic roadmap planning
RAFCON	RMC advanced flow control
RL	reinforcement learning
RMC	Robotics and Mechatronics Center
RMPL	Robot Motion Planning Library
RMS	reconfigurable manufacturing system
SIR	sampling and importance resampling
SIS	sequential importance sampling
SMC	sequential Monte Carlo
SVD	singular value decomposition
SVM	support vector machine
TFF	task frame formalism
VAE	variational auto-encoder
VCM	virtual contact manipulator
VPS	voxelmap-pointshell algorithm

1 Introduction

Customized and one-of-a-kind products are everywhere. Examples are highly specialized machines and devices, such as satellites and spacecraft. Typically, they are required only for a single use or mission and are individually designed and manufactured. Even in the high-volume car industry, the variants and available feature combinations are constantly rising, making each car a unique product (Fig. 1).

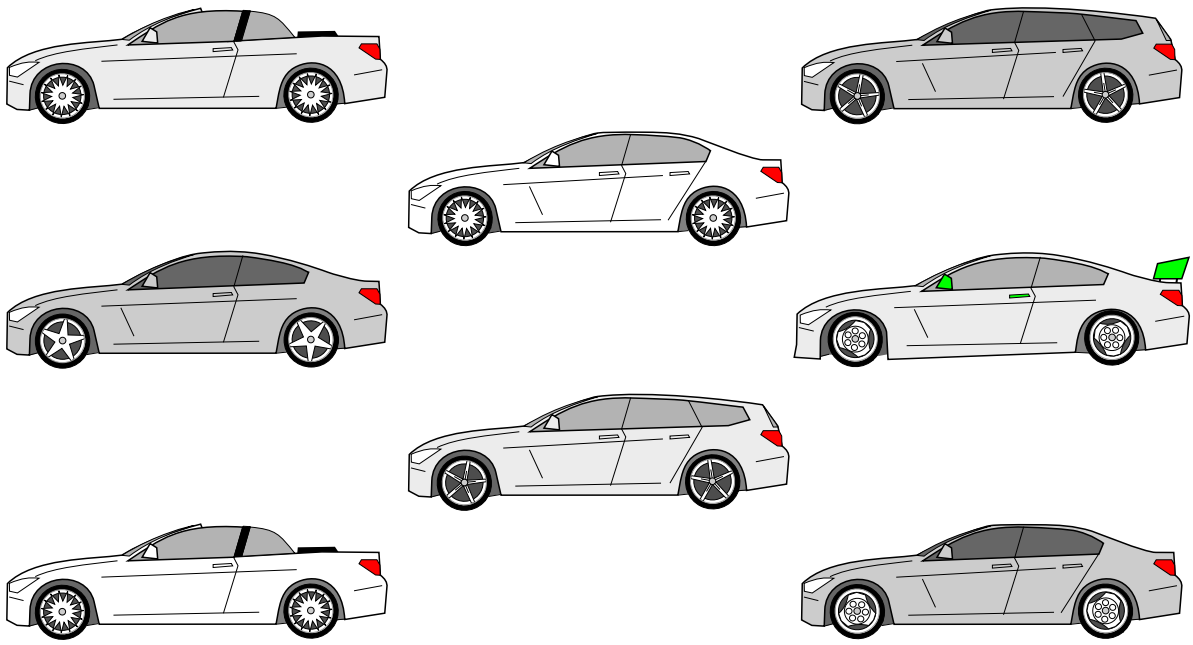


Figure 1: Product variants and a custom choice of available equipment features make each individual car a unique product.

Although a high rate of automation has been reached in specific processes, the production of custom products mainly relies on human labor as the effort to automate infrequent processes or processes under varying conditions exceeds the effort needed for manual execution. Consequently, in this work, a framework for flexible robotic assembly is developed and evaluated, together with methods to make robotic assembly adaptable to various situations, and thereby reducing the manual effort in the process chain.

1.1 Motivation

Until now, industrial robots have mainly been used in mass production. Typical examples are the automotive and the consumer electronics market, in which standardized products are offered for low prices in high quantities. The traditional assembly line concept, established from the early 20th century on by Ransom E. Olds and Henry Ford in the automotive sector [May77, p. 192; For23, p. 81], is specialized in producing single types of products, which are then offered for low prices to the customers. Thus, it became affordable for many people to own complex products like cars and electronic devices and, therefore, to benefit from the technical developments. In simple terms, increasing production volumes and standardization enabled a cost-effective production. However, there are general trends that require more complex and elaborate solutions, which are discussed in the following.

1.1.1 Challenges in Production

Modern societies express individuality in many facets, and therefore the demand for customized products has increased over the last century. The term mass customization (MC) was promoted to describe a concept that combines the demand for individual goods with the efficiency of a high-volume production [Dav87; Pin93]. One solution at the product level is the modularization of the product portfolio, for example, by applying product design methods such as modular function deployment (MFD) [Eri98]. Nevertheless, as a consequence of MC, the facilities and production methods need to be adapted as well in order to stay economically feasible while addressing the individuality of a product. Having solutions applicable for multiple use cases allows the production to become flexible with respect to the variety of products manufactured in the same facility. Flexibility[1] is therefore an important requirement for future production systems.

Besides product variety, there are further factors that require greater flexibility in production. In the automotive sector, the trend has been observed for a long time that, in parallel to the rising number of model types, product life is being shortened (e.g., [NKW10, p.5; Cac+09]). The increasing speed at which products change, new products are envisioned and brought to the market has a major impact on production. In the field of consumer electronics, as well, new models are released at an increasingly frequent rate. Consider, for example, the model evolution of the Apple iPhone starting in 2007 with a new model every year at the beginning, and reaching a number of five models just in 2020 [Car21]. In order to keep up with these trends, the implementation of new manufacturing processes needs to be simplified and provide solutions that can be adapted to new tasks efficiently in a short amount of time.

Furthermore, due to the shorter lifespan and the limited availability of resources, it is required that sustainable value chains are established. Therefore, the producers will have more

[1] See, e.g., Rogalski [Rog11] for general definitions and methods to evaluate flexibility in production systems.

responsibility not only to produce efficiently according to the market desires, but also to provide economical and ecological solutions for the full life cycle of a product.

In summary, the trends require that more product variants are manufactured in less time while using resources more efficiently. As a natural consequence, overall productivity must be raised significantly as the effort required for a single product tends to increase when its entire life cycle is taken into account. Automation and robotics technology could then play a key role in solving these challenges by reducing manual effort in multiple steps of the process chain.

1.1.2 Productivity and Assembly Automation

In today's factories, human labor is of central importance due to the high flexibility, dexterity, and general abilities of people to adapt to new situations and find efficient solutions for complex and delicate tasks. For example, the automation proportion in final assembly is estimated at only 10-15 percent.[2] Modern production facilities such as the *Daimler Factory 56* put the "focus on people" in the design of the plant [Dai21]. However, in addition to the trends mentioned above, factors like demographic change will reduce the availability of human labor for certain tasks and increase the need for enhanced productivity even more.

While the positive effect of information technology on productivity was not always clearly visible in the data, which is discussed under the term *productivity paradox*, it seems that robot technology improves productivity [MA15]. In their recent study, Graetz and Michaels [GM18, p. 766] conclude after analyzing the development of industries in several countries from 1993 to 2007 that "increased use of industrial robots is associated with increase in labor productivity". In the long term, robot technology could therefore help to compensate for the expected shortage of skilled human workers and to implement sustainable and flexible production systems.

These circumstances were already investigated in the early eighties by Geoffrey Boothroyd and colleagues, who conducted an economic comparison of various assembly systems by computing the assembly cost per part [Boo82; BCM82, p.240ff]. Factors considered were the number of parts per assembly, the number of different products, styles of each product and occurring design changes. For high volumes of single products, dedicated machines were most economical as expected. Whereas for different products, the assembly line with human workers was the best choice even for high volumes. Only the hypothetical *universal assembly center* could be competitive, especially for a high number of different products [BCM82, p. 254]. The concept of these universal assembly centers consists of two robotic arms, which can handle all the necessary assembly tasks for different products. This can be achieved by a close collaboration with the product design; for example, it is ensured that parts are graspable with the same universal gripper and insertable from above [BCM82, p. 241]. The optimization

[2] Compare, e.g., the percentages given by [BRS18, p.554; SM21, p.41].

of product design to ease assembly described here is the core idea of methods summarized under the term *design for assembly (DFA)* and were increasingly developed from the eigthies on [BA92].[3]

In line with these concepts, a concrete example for a flexible assembly system is the SMART (Sony Multi-Assembly Robot Technology) system, which is optimized for small lot sizes and aims to be adjustable to new products by using *multi-function machines* instead of specialized stations for single tasks [Hit88]. Similarly, the MARK III flexible assembly automation (FAA) cell [EGO94; Lan98; OG98] demonstrated the assembly of about 120 known variants of electronic connectors, and the applicability to three other use cases [ELO98]. With these and other FAA systems[4] it was demonstrated that large product varieties can be assembled automatically on the same shop floor enabled through a close interaction of product design and automation engineering.

1.1.3 Robotic Autonomy for Flexible Assembly

However, the manual effort did not vanish completly, but was shifted to the design phase to optimize the products and the production for specific applications and expected variants. Finally, the concept of creating a “multi-purpose machine that tries to cover a wide range of functionalities” at the center of flexible manufacturing system (FMS) showed to be inefficient as there are too many uncertainties on future requirements, as for example Onori et al. [OAH11, p.5] conclude. Instead, Onori et al. [OAH11, p.5] suggest implementing evolvable assembly systems (EASs) using “truly re-configurable modules with embedded control that evidence autonomous behavior to react to disturbances” (compare also: [Ono02; OBF06]). As a consequence of the downsides of FMS, concepts for reconfigurable and changeable manufacturing systems were developed providing solutions that can deal better with uncertainties about future requirements and react faster on market changes, for example, reconfigurable manufacturing systems (RMSs) [Kor+99; ElM06; Wie+07], cognitive [Zäh+09] and agile factories [SVB15].

Although these concepts and frameworks exist, there is still a gap between theory and practice when it comes to robotic solutions providing sufficient capabilities to fulfill their role in such a rapidly changing production environment. A major factor is still the actual implementation effort. To this day, the implementation of robot applications in industry is done manually by robotic experts and specifically for every new use case. Of course, there have recently been some great improvements through rich offline planning tools, intuitive programming, programming by demonstration (PbD), and human robot interaction (HRI) in general that make it significantly easier and faster to implement robotic solutions for specific use cases.[5]

[3] Compare also [Esk01] with a stronger focus on design for *automatic* assembly.

[4] See Chapter 4 of Johansson [Joh02] for an extensive overview of FAA systems from this period.

[5] Offline planning tools are, e.g., Visual Components, Dassault Systèmes DELMIA, Siemens Tecnomatix Process Simulate etc.; see, e.g., [Vil+18] for a survey on HRI for robot programming in industrial applications.

However, for long-term development, it is required that the robotic systems in such dynamic manufacturing environments provide more capabilities to adapt flexibly to new tasks without increasing manual efforts. This can only be achieved if the robotic systems find solutions autonomously on multiple stages and execute them reliably under varying conditions without the need for human intervention. Therefore, in this thesis, it will be investigated how individual products can be manufactured using robotic systems that can plan and execute autonomously the given assembly tasks.

1.2 Problem Statement

The implementation of an assembly application goes typically through multiple stages. First, there is a design and planning phase in which possible assembly sequences need to be defined. The major factors of the physical assembly process to consider here are the mechanical properties of the parts and the constraints of the desired connections. Then in a next phase, it is required to map identified assembly steps to available resources in the production facilities, which provide the capabilities to solve these tasks. Before the execution can finally be started, it is necessary to prepare the facilities, provide instructions, and implement machine programs for an automated execution. The effort needed to realize the assembled product accumulates along this process chain and iterations in the development add additional delays. Consequently, an important goal for flexible assembly of individual products is to eliminate unnecessary manual effort and support the process steps with automation methods for an efficient development chain with fewer manual iterations.

In this thesis, it will therefore be investigated how a flexible robotic assembly system can be realized that reduces manual effort in the overall implementation pipeline from product design to assembly execution. Accordingly, the goals of the thesis can be assigned to three major research questions as will be described in the following.

1.2.1 Flexible Robotic Assembly System

Research question 1
How can complex individual products be assembled with less manual effort using robots equipped with advanced sensing and control capabilities?

Ideally, the designer of an individual product only needs to specify the parts and the constraints under which they are connected. The role of the designer could be taken by an engineer in the company or even by a customer using dedicated product configuration tools (Fig. 2).

Figure 2: By using a dedicated product configurator with an intuitive interface, customers are able to design individual products without considering the required production processes. Here, the products are aluminum structures composed of parts from a modular construction set. Source: Courtesy of DLR (left)

Given such a product specification, a robotic assembly system with sufficient planning capabilities could then decompose the assembly into tasks and assign appropriate resources. Then the necessary actions could be planned and executed in a reliable and efficient manner. Tasks to be solved by the robotic system would be, for example, sequencing of assembly operations, selection of appropriate grasps to manipulate the parts, planning motions and finally a robust execution of the assembly process.

In this work, steps toward this ultimate vision of mass customization will be investigated and implemented, with a special focus on handling the physical assembly process using the sensing and control capabilities of robotic systems. Note that flexibility of production systems needs to be implemented on multiple levels of the manufacturing process. In this work, mainly the robotic perspective is taken on a workstation level to realize the actual process. Furthermore, the scope of this work is a fully automated system, in contrast to hybrid systems, which also incorporate human labor for execution of tasks.[6]

1.2.2 Contact Sensing

Research question 2
How can assembly tasks be observed using intrinsic tactile sensing with joint torque sensors?

The assembly process itself is a transition of a part from a free configuration to a typically very constrained configuration at its goal pose. Contacts are obviously both a natural and, at

[6] Although a focus is solely put on methods for automated execution, it is the author's strong belief that future production facilities will be collaborative environments shared by human workers and robotic agents.

the same time, critical factor for the success of the process. For example, Fig. 3 shows the insertion of an aluminium profile into an assembly fixture, which requires a specific sequence of contacts.

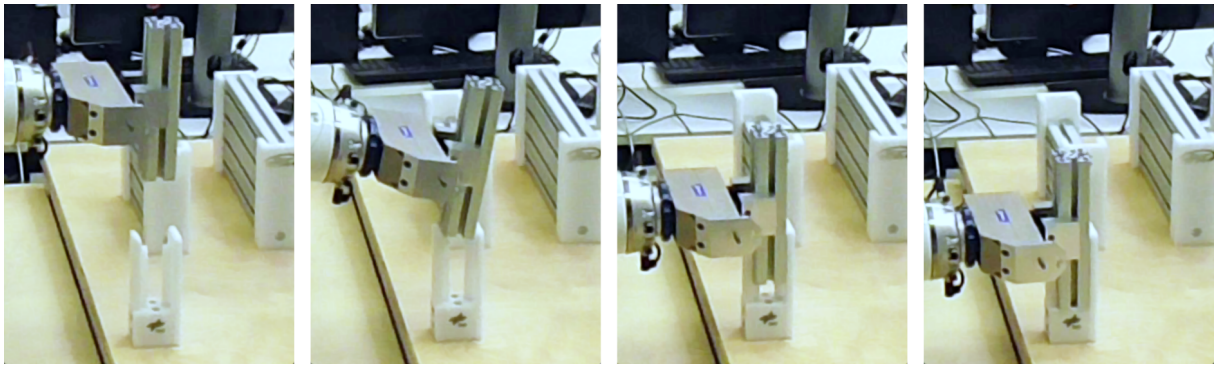


Figure 3: Insertion of an aluminum profile into an assembly fixture. A specific sequence of robot motions and contacts is required to align the geometric features until the part reaches the specified goal pose.

The *sense of touch* provides feedback about contacts and helps humans execute such tasks under difficult and varying conditions. Especially, it complements sensory feedback when vision fails due to occlusions or unsuitable lighting conditions. In robotics, the sense of touch can be classified according the location of the sensor: *extrinsic* and *intrinsic tactile sensing*.[7] Extrinsic tactile sensing aims to measure the effects of contact directly in the contact interface (e.g., with tactile skins), and intrinsic tactile sensing from measurements of internal joint and force/torque sensors [Dah+10, p. 3].

Intrinsic tactile sensing is especially relevant for industrial applications as the used grippers typically provide no (or only incomplete) tactile information. Furthermore, extrinsic tactile sensing directly in the contact interfaces of assembly parts is, in general, physically not possible. Therefore, it will be investigated how intrinsic tactile sensing can be used for the observation of assembly tasks. It is expected that feedback from the observation will allow the implementation of flexible and adaptive assembly operations that can deal autonomously with present uncertainties.

In the considered assembly cases, it can furthermore be assumed that the parts are grasped stably with a known relative pose between parts and gripper.[8] This is a feasible assumption in many industrial scenarios, where parts are provided to the assembly system in well-oriented configurations on transport trays.[9] Under this assumption, it is expected that intrinsic tactile

[7] The term *intrinsic tactile sensing* was introduced by Bicci and Dario in contact sensing with humanoid hands measuring the resulting force and torque with sensors inside the mechanical structure and not measuring the pressure distribution directly at the surface [BD88; BSD89; BSB93; MDR21].

[8] The *in-hand localization* of objects inside a gripper or humanoid hand is a related problem, but will not be investigated here. Compare, e.g., the approaches of [CRP13; Pfa+18].

[9] This can be achieved by special devices like the advanced parts orientation system (APOS) [Hit88] or by the combination of perception and grasp planning in the field of *bin picking*, e.g., [Kle+20].

sensing provides sufficient information about the occurring contacts. In particular, the joint torque sensors of a light-weight robot (LWR) provide information about external contacts and will be used in this work to observe uncertainties during the assembly task execution.

1.2.3 Adaptive Task Execution

Research question 3
How can contact information from online observation be used for a flexible and adaptive execution of assembly tasks?

Since the development of force control and compliant assembly devices and methods, it is technically feasible to assemble parts with robots, even with tight fittings and low tolerances.[10] Using impedance control [Hog84; AOH07] and an appropriate motion strategy, it was successfully demonstrated by Stemmer et al. [Ste+06] how complex shaped parts can be mated under the presence of uncertainties in the part poses.

The major challenge on process level is nowadays to reduce the implementation and programming efforts for individual assembly tasks. A programming paradigm on *task-level* would reduce the complexity of programming and focus on the requirements of the process. Combined with a library of modular and reusable robotic operations, which are capable of solving the process under the given constraints, a flexible robotic assembly system could be created. These *robotic skills* could then be selected manually by a user or automatically by a task-level based planning system and thereby reduce the implementation effort drastically.

In order to achieve high reusability, it is necessary that the skills support a wide range of application cases and can adapt to the present constraints autonomously. Consider, for example, the application of a skill for insertion tasks with parts of varying geometric properties as shown in Fig. 4, but also cases in which the location of execution is different and part geometries are less similar.

In this work, it will therefore be investigated how adaptive task execution can be achieved for assembly tasks. Existing solutions often do not actively use sensory feedback during the insertion or have only limited capabilities to react to the present situations. Therefore, it will be investigated how the feedback from tactile sensing can be incorporated to deal with geometric uncertainties in the part poses. This is of special interest for flexible scenarios in which the need for accurate calibrations of workcell components should be reduced or less specialized fixtures should be used to have fewer constraints in the product variability. A special focus in this work lies on insertion tasks, as they serve as a critical benchmark for a large variety of assembly tasks.

[10] See, e.g., [Ino74; Dra78; WN79; GTI80; Mas81]; [MF12, Section 5] for an overview of force control methods used for robotic assembly.

Figure 4: Insertion of aluminum profiles of varying length into an assembly fixture. The execution should adapt autonomously to the individual constraints of the tasks without generating manual programming effort.

1.3 Related Work

The goal of developing an autonomous robotic assembly system capable of producing a large variety of product variants can be found in related works. A brief overview of flexible and autonomous robotic assembly systems is provided in the first part of this section. Further on, existing methods for contact sensing in robotic assembly are described in short and approaches for task execution under the presence of uncertainty are presented.

1.3.1 Flexible Robotic Assembly System

As stated above in the motivation for this work (Section 1.1), the FAA systems [Joh02] provided a high grade of flexibility, but still required a substantial design and implementation effort, and as a consequence it was suggested to employ modules with autonomous behavior to circumvent the drawbacks (e.g., [OAH11]).

The concept of *autonomy* itself was introduced in the field of robotics early in the sixties by the development of the SHAKEY robot [Nil69; Kui+17]. In general, autonomy in robotics is understood as the “ability to perform intended tasks based on current state and sensing, without human intervention” [ISO12, p.1]. Accordingly, also in the manufacturing domain, the development of systems has begun that could plan and execute sequences of assembly tasks autonomously, and thus realize “manufacturing plants of great flexibility” [Rem88, p.598].

One of the first autonomous robotic assembly systems was described by Ejiri et al. [Eji+72]. It was able to understand a production goal given as assembly drawing, and plan and execute necessary steps autonomously. Another example is the Karlsruhe Autonomous Mobile Robot (KAMRO) [Rem88; HR91; LR94], which also provided the capabilities to autonomously assemble products. Problems to be solved are, on the one hand, sequencing of actions related to the (automatic) assembly sequence planning (ASP) and, on the other hand, dealing with uncertainties in the environment during the execution using perception and control

strategies, and the integration of these functionalities in an appropriate robotic platform and software framework.

In recent years, multiple autonomous and flexible robotic assembly systems have been presented for various application scenarios, for example: the assembly platform MARVIN using a three-level cognitive system [Sav+18] evaluated with the Cranfield Benchmark [CPR85], the flexible assembly of hydraulic valve sections [LSW16], IKEA chair assembly using bi-manual robotic motions for handling larger parts [SZP18], the assembly of modules on hot-top rails of control cabinets using an hierarchical planning approach [Kas+19], and the assembly of LEGO constructions [May+11; Näg+20]. Furthermore, corporate research departments in automation industry have identified autonomous assembly as a core technology for future production (e.g., Siemens [Zis17]).

It can be stated that higher levels of autonomy show huge potential in reducing manual planning effort and relaxing requirements on the workcell and production line design. Nevertheless, drawbacks of multiple proposed approaches are still the partially unreliable task execution, simplified and non-industrial scenarios, the low complexity of the considered assembly processes, as well as the limited performance compared to manual execution.

1.3.2 Contact Sensing

For the execution of assembly tasks, which are typicaly *contact-rich* manipulation tasks, it is required to maintain and establish contacts between the parts in a particular manner in order to have a successfull execution. Contact sensing enables the robotic system to gather information about the present contact situation, and thus provides a basis to adapt the execution accordingly, which is especially important under the presence of uncertainty. The observation of contacts by force measurements was pioneered by the work of Salisbury in the context of robotic hands using extrinsic tactile sensing [Sal84; Sal85; De +99, p. 1164]. Since then, various approaches have been investigated that deal with mainly two problems: recognition of contact states and estimation of uncertainties.[11]

The recognition of semantic *contact states*, which are associated with particular types of contacts, is often motivated by necessary switches of the active control law or the motion strategy according to the current contact situation. For example, the method by Hirai and Iwata [HI92] models the space of possible contact wrenches for a certain contact with polyhedral convex cones (PCCs) [GT56], which can be derived from the geometries of the parts in contact. The PCCs are then used as discriminant functions for classifying contact states. Following similar model-based principles, multiple approaches were proposed, for example by [DV89; Xia93; MA93; MF94; DDH04; LBD05]. Further, methods from machine learning were investigated for contact state recognition using, for example, stochastic gradient boosting

[11] See [Lef+05] for a survey about (traditional) methods in the field of uncertainty estimation and contact state recognition for compliant motions (as well as for related planning and control methods).

[CCS10] and support vector machines (SVMs) [JPH11; YR18]. The early works used rather simple part geometries; later, algorithms were developed for automatic generation of so-called *contact state graphs* of complex shaped and curved 3D geometries [XJ01; Tan07; TX08], which were used as prior knowledge about possible states and transitions for online recognition [Mee+04; Mee+07; Her+12; LLT15]. However, the methods for contact state recognition were mostly demonstrated only with simple shaped geometries. In addition to the modelling of states for complex geometries, another difficulty is the clear matching between measured forces and discretized contact states, which makes it challenging to handle complex geometries.

Besides recognition of discrete or semantic contact states, estimation of (geometric) uncertainties is useful to adapt the execution according to the actual situation, for example according to the location of contact surfaces or part dimensions. Especially, Bayesian state estimation methods were investigated for this purpose, which provide the possibility to include prior model knowledge and update it through observation using sensory data. In particular, the (extended) Kalman filter was used in multiple approaches (e.g., [BB87; De +99]). Those approaches provided accurate estimates, but were assuming particular contact states and required filter switching for new contact situations [De +99, p.1172]. Then, Gadeyne et al. [GLB05] proposed a hybrid framework based on a *particle filter* to simultaneously estimate contact states and geometric parameters, which was generalized by Meeussen et al. [Mee+07] and applied for compliant motion control [Mee+08]. Particle filters, which are Bayesian estimation algorithms using sequential Monte Carlo (SMC) methods [DFG01; CGM07; DJ11], have become popular in robotics, at first, for localization of mobile robotics [TBF05]. Advantages of SMC methods are that they can handle nonlinear and non-Gaussian models [DFG01], which makes them suitable for observing contacts which are strongly nonlinear due to the contact transitions. Consequently, multiple approaches propose to estimate geometric uncertainties during execution of assembly tasks with particle filters [GLB05; CB05; Mee+07; Tho+07; TMO10; AJT16; Wir+19].[12]

1.3.3 Adaptive Task Execution

Especially in a fast changing setup, it is of major importance that implemented solutions can be reused under varying conditions and without requiring new manual effort. Therefore, one prerequisite for reusing assembly solutions is that they need to be able to compensate for uncertainties that might arise in novel situations. In the early seventies, Inoue [Ino74] described in detail two different types of “stereotyped actions” for shaft-bearing assembly, one for a loose and the other for a close fit with small tolerances. Both contain a sequence of dedicated shift and tilt motions to compensate for pose uncertainty. Such object-centric and feature-based behavior was later formalized, for example, using the task frame formalism (TFF) [Mas81; BD96], task functions [SEL91], or the constraint-based task specification

[12] Similary, these methods found application in grasping and in-hand localization, e.g., [ZT12] and [CRP13].

(iTaSC) methodology [De +07]; and furthermore got embedded in the concept of robotic skills [HST92; Tho+03; Bjö+11; Bøg+12; Tho+13; WKW14; Wah+15; SW16]. Although there are slight differences in the detailed definitions, skills act in overall as an intermediate layer between task description and robot control to improve reusability and provide implemented strategies for execution of certain classes of tasks.[13]

In order to implement adaptive behavior on motion level and to deal with uncertainties during assembly, various approaches have been proposed in the past that typically rely on compliance and force control [VD16], and use dedicated strategies to improve success rates. For example, force guided strategies [SP94] and methods to (passively) align geometric features of complex shaped parts were developed [Ste+06; SAH07; SKS16]. For larger uncertainties, so-called *blind-search* strategies were proposed that cover relevant regions with random or systematic search movements, as suggested for example by [Bad+91; New+99; CB01; Jok+16].[14] These strategies often suffer from a weak performance due to the time spent in the search phase and have only limited capabilities to interpret sensor measurements. Through the use of dedicated planning methods for fine motions, for example, [LMT84; Don86; Erd86; RBS05; SAH07; Wir+18], the aim is to precompute optimal and reliable movements even under the presence of uncertainties. However, these plans are typically computed offline and do not incorporate reasoning about the actual state in the workcell.

The early work of Bolles and Paul [BP73] already demonstrated how sensory feedback could be used directly for uncertainty reduction in automated assembly. Accordingly, a common strategy is to use sensors to localize parts as accurately as possible before the actual execution of the assembly task starts. In particular, vision systems are appropriate for determining an initial pose of an object (e.g., in [Ste+06]), and work for moving parts [Jör+00; Che+09; LSH10]. Nevertheless, lighting conditions and the setup need to be carefully configured, and the remaining error is usually so large that compliant motions are still required during the assembly process.

Besides visual sensing, passive alignment, and search strategies mentioned above, it is therefore also of importance that robotic assembly skills can actively adapt using a richer interpretation of the feedback from contact sensing. The interpretation of sensed contacts can be used, for example, to recover from failures due to undesired contacts using motion replanning [XV89; Xia90], to build a complete framework for planning and execution of compliant motions [Mee+08], and to switch between certain phases of the process and adapt the controllers accordingly (e.g., [AVK16; Sto15]). In particular, feedback from contact sensing enables implementing object-centric assembly strategies that actively adapt to the present contact situation.

[13] In this context, the concept of skills is mainly related to robot motion control. A generalization of it is possible, in which a skill could be seen as an agent behavior that allows to reach arbitrary desired goal states in multiple situations and under varying conditions.

[14] Compare also [MBF18, Section 3] for a review of common motion and search strategies in robotic assembly.

1.4 Approach and Contributions

In this work, a flexible robotic assembly system is developed, which is capable of producing a large number of product variants by only providing the specification of the desired product. Furthermore, methods for contact observation are proposed and embedded in reusable and adaptive robotic assembly skills. An overview of the contributions of this thesis is given in this section sorted according to the three main research questions.

1.4.1 Flexible Robotic Assembly System

In **Publication 1** [Not+16a] of this dissertation, a flexible assembly system is presented that is capable of assembling unique products with parts from a modular aluminum profile construction set (see Fig. 5, left). It was developed within the EU project SMErobotics [Per+19] to showcase a fully automated pipeline from intuitive product specification to robot execution.

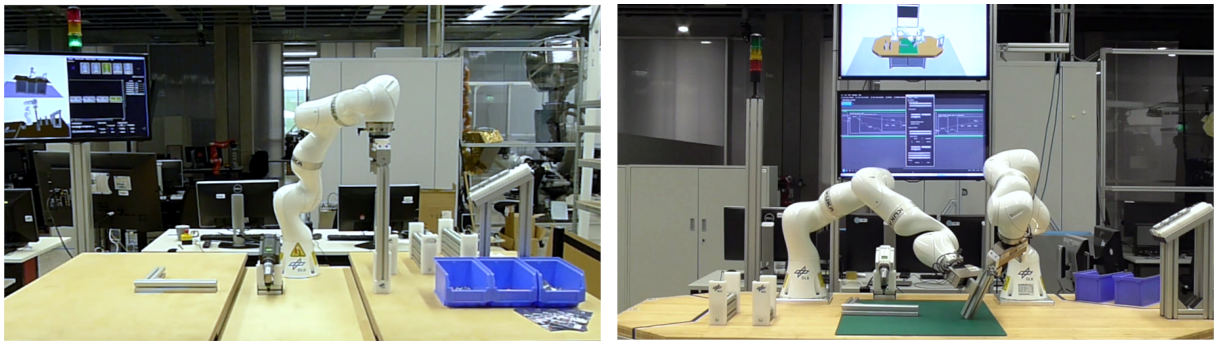


Figure 5: Single-arm robotic assembly system as part of this work (left). Extension to a dual-arm robotic system (right).

The main features of this system and contributions of the work are:

- Integration of an automatic assembly sequence planner on task-level using CAD data
- Mapping of tasks to skills using a task classification approach
- Implementation of a skill library for assembly tasks with multiple abstraction levels
- Integration of grasp and path planning components for solving geometric subproblems
- Simulation of the overall execution for checking feasibility and storing solutions
- Usage of impedance control for robust execution under the presence of uncertainties
- Tracking of object state changes during the execution in an online world model

Besides the system presented in this dissertation, significant contributions were made to the further development of main aspects of such a flexible robotic assembly system in further extensions and publications (not part of this dissertation):

- Extension of the original single-arm system to a dual-arm robotic assembly system [Rob20a; Roa+17] together with Andreas Stemmer, Tim Bodenmüller, Maximo Roa, Daniel Seidel, Arne Sachtler, Ismael Rodríguez, Michael Kaßecker, Timo Bachmann, Peter Lehner, Jan Cremer, Andrea Schwier and further colleagues at DLR RM; The goal was to make the system more flexible with respect to the possible product variants by removing constraints from fixtures and by adding more degrees of freedom into the overall system (see Fig. 5, right).
- Methods for making ASP more efficient were further investigated with Rodríguez et al. in [Rod+19] and [Rod+20]; While traditional methods often focus on the assembly itself without considering the robotic system, a major goal was to incorporate feasibility checks on different levels of complexity and to improve the reusability of found solutions through pattern matching.
- Methods for task specific analysis of the workspace for solving the geometric workpiece placement problem and optimization of workcell layouts were developed together with Bachmann et al. [Bac+20; BNR21].
- Knowledge representation for flexible robotic assembly systems using an ontology [Sch+21b] based on the IEEE Std 1872™-2015 [IEE15]
- Transfer of methods into the in-space assembly domain [Roa+17; Roa+19] and ground based demonstration [Rob21; Roa+22] within the EU project PULSAR; The goal is to apply similar methods for the autonomous assembly of (large) structures in space where teleoperation is not economical or technically feasible, for example due to communication delays.

1.4.2 Contact Sensing

In this work, a sequential Monte Carlo (SMC) approach is chosen to enable intrinsic tactile sensing using sensory input provided by a LWR [Alb+07]. The primary focus is on the estimation of part locations during assembly and, in particular, on insertion tasks, which are difficult to observe due to tight constraints in the configuration space, so-called *narrow passages* in the configuration space. Insertion tasks (also: *peg-in-hole* tasks) are very frequent in manufacturing in various forms and serve as a well established benchmark for contact-rich manipulation tasks in robotics.

Furthermore, although Bayesian filtering methods are in general suitable for tracking, existing works in contact sensing usually assume a static environment. Therefore, a further important aspect is to extend the methods so that they can deal with non-static objects. This could reduce the requirements on dedicated fixtures in flexible robotic workcells.

Note that the used impedance controller of the LWR is robust with respect to uncertainties in the contacts [AOH07, p.37] and stable in contact without exact knowledge about the constraints. The recognition of contact states for enabling compliant motions is therefore not investigated in this work,[15] but could be inferred from estimation outputs using knowledge of the parts geometries. However, knowledge about the locations of the parts obtained from contact sensing can be used for adapting the execution.

The general SMC algorithm recursively propagates samples of a state (i.e., *the particles*) with a propagation model and then updates their distribution according to their consistency using measurements and observation models [DJ11]. From the sample distribution, estimates of the uncertain (and typically partially hidden) state can be computed.

In **Publication 2** [Not+16b] heuristics from probabilistic roadmap planning (PRM) [Kav+96] are adopted in the propagation step of the filter to improve sampling performance in narrow passages of the configuration space (see Fig. 6). The contributions of this publication are:

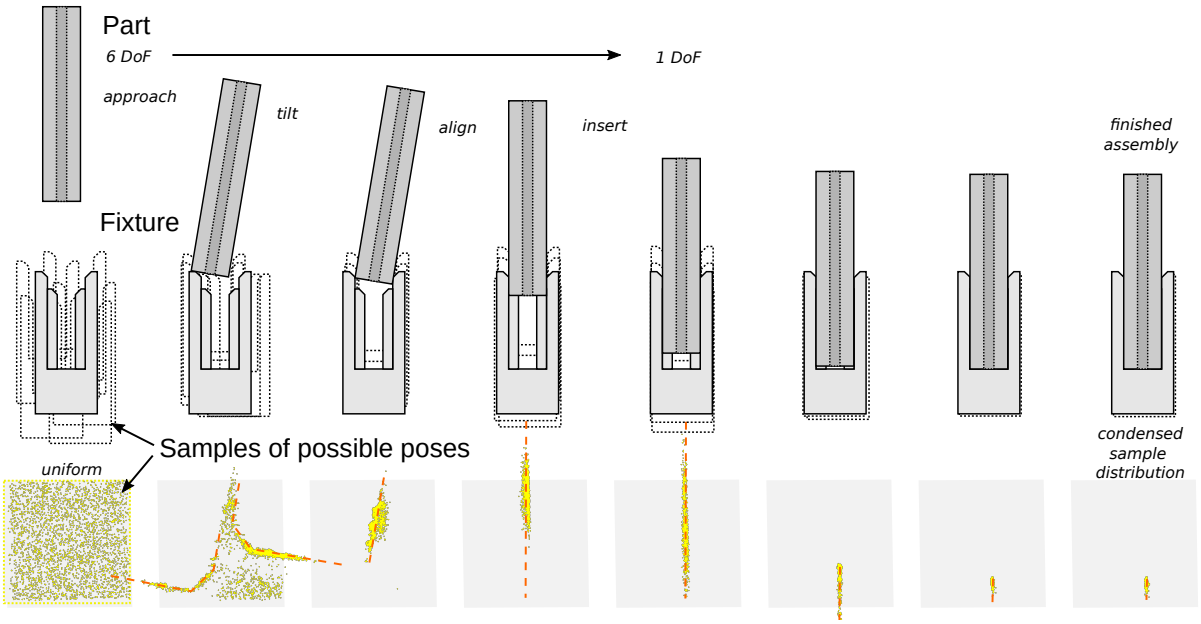


Figure 6: Evolution of the samples used for state estimation during the placement of a part into a fixture for which a tilt-and-align strategy is implemented. Each sample represents a possible pose of the fixture. The distribution of the samples condenses to a small region in the relative configuration space which correpsonds to an accurate localization. Source: Revised diagram from **Publication 2** (Fig. 6)

[15] Contact state recognition was investigated by the author in [Not12] using kinematic contact models.

- Formulation and evaluation of propagation steps using heuristics inspired by the *Gaussian sampler* of Boor et al. [BOS99] and the *bridge test* by [Sun+05]; Particles are collected at the regions of the configuration space in which contacts are expected.
- Integration of an efficient contact model using an accurate and fast implementation of the voxelmap-pointshell algorithm [MPT05], which supports complex non-convex geometries [Sag19]
- Formulation of the observation model using joint torque and position measurements from a LWR[16]

Publication 3 [NH17] and **Publication 4** [NSA21] consider cases in which parts are not fixed in the environment, and suggest two possibilities to efficiently incorporate motion of parts in the propagation model:

- Constraint-based propagation using information about the local shape of the contact surface; A formulation is derived using the reciprocity property [Bal98] in the contact and a method inspired by time-domain passivity control to compensate superfluous virtual energy.
- Integration of a constant velocity model (CV) for tracking moving objects [Cha+11, p. 57ff, p.64] in the propagation step.

Besides investigating intrinsic tactile sensing using joint torque sensors in this dissertation, it was researched how other sensor modalities could be used to retrieve information from sensed contacts. Contributions were made to the following publications in this context (not part of this dissertation):

- Fusion of visual and tactile sensing for probing-based localization of objects using circular features together with Sachtler et al. [Sac+19]; Parts of the visual likelihood model were reused in **Publication 4** [NSA21].
- Measuring structure-borne sound could be an alternative sensory modality to get information about the contact. Together with Neumann et al. [Neu+18] a material classification method using a variational auto-encoder (VAE) was developed and evaluated.

1.4.3 Adaptive Task Execution

In **Publication 4** [NSA21], the method of Stemmer et al. [SAH07] is adopted and combined with an online approach for intrinsic contact sensing and adaptive motion generation. The methods are implemented within an object-centric assembly skill. The combination of contact sensing and motion generation allows to execute assembly tasks with moving objects also in cases where the field of view of a vision system is occluded (see Fig. 7).

[16] A preliminary version of the observation model was presented as poster at the DGR Days 2013 [Not+13].

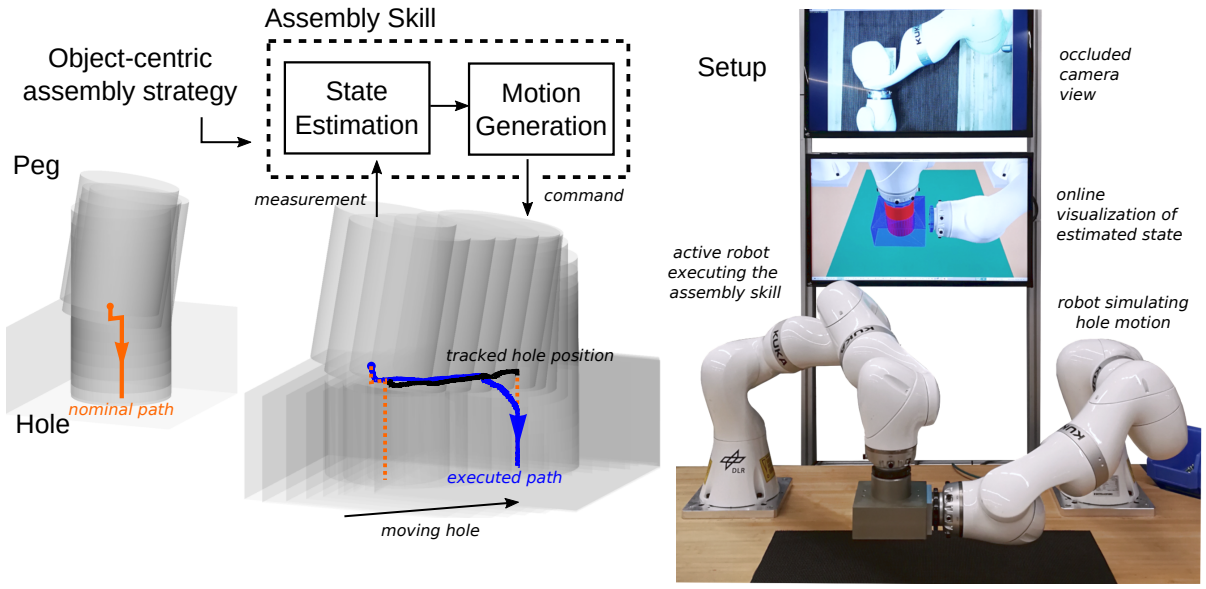


Figure 7: Adaptive execution of an assembly task with moving hole using state estimation based on intrinsic tactile sensing in combination with visual perception and the adaptive motion generator in the assembly skill. Source: Pictures taken and rearranged from **Publication 4** (Fig. 8 and Fig. 10).

The main contributions of this work are:

- Integration of intrinsic tactile sensing and visual perception in a recursive Bayesian estimation framework using SMC.
- Continuous online estimation of part poses also in cases where only a single sensing modality is available.
- Development of an adaptive path execution framework and demonstration with a predefined motion strategy.

In **Publication 5** [NSA20], the observed configuration space is explored to generate motion towards a defined goal pose instead of using a predefined motion strategy. During estimation, it can be observed that the particles condense within the feasible regions of the relative configuration space between both parts and describe thereby a space of possible local motions. In this space, a next motion for reaching the desired goal pose can be selected. The concept of reusing the particles from estimation for motion generation makes an additional sampling step superfluous. In short, the contributions of this work are:

- Reduction of programming effort by only requiring part geometry as mesh data and a relative assembly pose as goal specification.
- Integration of a sample-based motion generation step in the recursive estimation framework; The locally sampled configuration space can be exploited for motion generation.
- Introduction of a task-specific sample weight to guide relative part motion.

Furthermore, Bayesian estimation methods were used in an approach related to active uncertainty reduction through touch-based probing of circular features [Sac+19] (not part of the dissertation). Based on the distribution of the features in state space, the algorithm decides upon the next probing action to reduce uncertainty subsequently and localize the part with respect to the robotic system. Through this autonomous registration method, the manual calibration effort for new workcell components and parts in a flexible robotic assembly system can be significantly reduced.

1.5 Structure of Work

This work is a publication-based dissertation and is structured as follows (see also Fig. 8):

Chapter 3 introduces the methods and fundamentals of this work.

In Section 3.1, the background of task-level abstraction and robotic skills is briefly introduced before presenting the developed skill concept and implemented framework for flexible robotic assembly (relates to Research Question 1).

In Section 3.2, Section 3.3 and Section 3.4, fundamentals of robotic manipulation, probability theory and Bayesian estimation are described in short, as well as the basic models for representing manipulator kinematics and contacts.

Based on these basic methods, Section 3.5 explains the considered uncertainty model and introduces the developed methods for contact sensing in this work for static setups and scenarios with moving parts in the environment (relates to Research Question 2).

Similarly, Section 3.6 describes the methods for adaptive task execution using an adaptive path executor and a locally guided motion generation method (relates to Research Question 3).

Chapter 4 provides short summaries of the individual publications on which this dissertation is based. Full text versions are attached in the **Appendix**.

Chapter 5 discusses the developed approaches and further development steps from a general perspective and includes references to the latest developments in current and related works. Among others, it is discussed how the flexible system can be embedded in production networks, the framework for contact sensing can be further developed and more learning can be included in skills in the future.

Chapter 6 summarizes the results in short and concludes with possible future research directions.

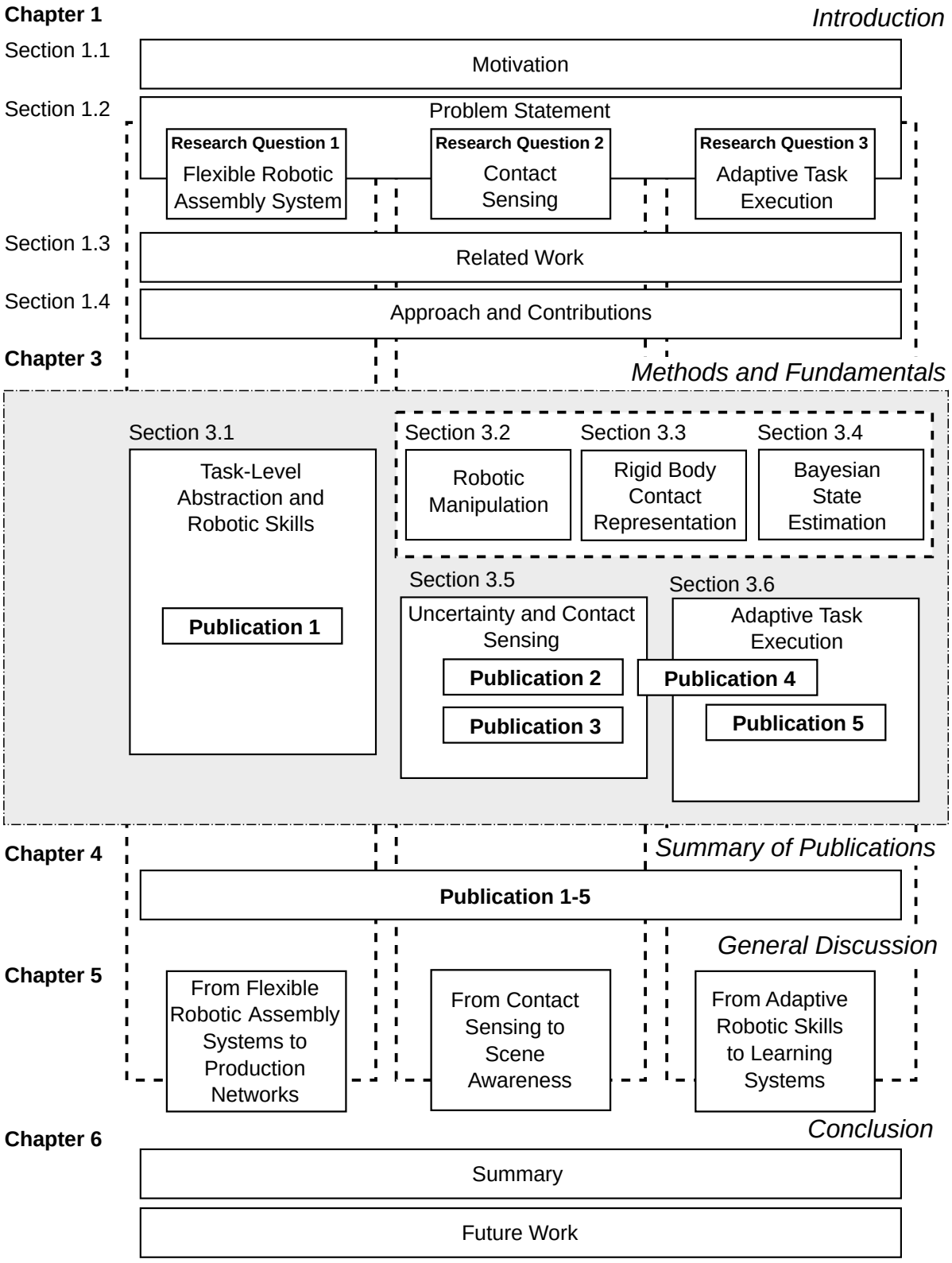


Figure 8: Structure of work.

2 Publication Note

This dissertation is directly based on the following 5 peer-reviewed papers (4 international conference publications, 1 journal article):

Publication 1 [Not+16a]
K. Nottensteiner, T. Bodenmüller, M. Kaßecker, M. A. Roa, A. Stemmer, T. Stouraitis, D. Seidel, and U. Thomas (2016): "*A Complete Automated Chain for Flexible Assembly using Recognition, Planning and Sensor-Based Execution*". 47st International Symposium on Robotics (ISR).

Publication 2 [Not+16b]
K. Nottensteiner., M. Sagardia, A. Stemmer, and C. Borst (2016): "*Narrow Passage Sampling in the Observation of Robotic Assembly Tasks* ". IEEE International Conference on Robotics and Automation (ICRA).

Publication 3 [NH17]
K. Nottensteiner and K. Hertkorn (2017): "*Constraint-based Sample Propagation for Improved State Estimation in Robotic Assembly*". IEEE International Conference on Robotics and Automation (ICRA).

Publication 4 [NSA21]
K. Nottensteiner, A. Sachtler, and A. Albu-Schäffer (2021): "*Towards Autonomous Robotic Assembly: Using Combined Visual and Tactile Sensing for Adaptive Task Execution*". Springer Journal of Intelligent & Robotic Systems (JINT).

Publication 5 [NSA20]
K. Nottensteiner, F. Stulp, and A. Albu-Schäffer (2020): "*Robust Locally Guided Peg-in-hole with Impedance Controlled Robots*". IEEE International Conference on Robotics and Automation (ICRA).

Besides these publications, which are directly part of this work, the following 7 related publications were co-authored, and significant contributions were made to the further development of the main aspects of flexible robotic assembly systems:

[Neu+18]
M. Neumann, K. Nottensteiner, I. Kossyk, and Z. Marton (2018): "*Material Classification through Knocking and Grasping by Learning of Structure-Borne Sound under Changing Acoustic Conditions*". IEEE 14th International Conference on Automation Science and Engineering (CASE).

[Rod+19]
I. V. Rodríguez Brena, K. Nottensteiner, D. Leidner, M. Kaßecker, F. Stulp, and A. Albu-Schäffer (2019): "*Iteratively Refined Feasibility Checks in Robotic Assembly Sequence Planning*". IEEE Robotics and Automation Letters (RA-L).

[Sac+19]
A. Sachtler, K. Nottensteiner, M. Kaßecker, and A. Albu-Schäffer (2019): "*Combined Visual and Touch-based Sensing for the Autonomous Registration of Objects with Circular Features*". IEEE 19th International Conference on Advanced Robotics (ICAR).

[Rod+20]
I. V. Rodríguez Brena, K. Nottensteiner, D. Leidner, M. Durner, F. Stulp, and A. Albu-Schäffer (2020): "*Pattern Recognition for Knowledge Transfer in Robotic Assembly Sequence Planning*". IEEE Robotics and Automation Letters (RA-L).

[Bac+20]
T. Bachmann, K. Nottensteiner, I. V. Rodriguez Brena, and A. Stemmer (2020): "*Using Task-Specific Workspace Maps to Plan and Execute Complex Robotic Tasks in a Flexible Multi-Robot Setup*". 52nd International Symposium on Robotics (ISR).

[BNR21]
T. Bachmann, K. Nottensteiner, and M. A. Roa (2021): "*Automated Planning of Workcell Layouts Considering Task Sequences*". IEEE International Conference on Robotics and Automation (ICRA).

[Sch+21b]
P. M. Schäfer, F. Steinmetz, S. Schneyer, T. Bachmann, T. Eiband, F. S. Lay, A. Padalkar, C. Sürig, F. Stulp, and K. Nottensteiner (2021): "*Flexible Robotic Assembly Based on Ontological Representation of Tasks, Skills, and Resources*". 18th International Conference on Principles of Knowledge Representation and Reasoning (KR).

Furthermore, the following 6 publications were co-authored with contributions related to transferring concepts of flexible robotic assembly into the space application domain:

[Roa+17]
M. A. Roa, K. Nottensteiner, A. Wedler, and G. Grunwald (2017): "*Robotic Technologies for In-Space Assembly Operations*". ESA Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA).

[Roa+19]
M. A. Roa, K. Nottensteiner, G. Grunwald, P. L. Negro, A. Cuffolo, S. Andiappane, M. Rognant, A. Verhaeghe, and V. Bissonnette (2019): "*In-Space Robotic Assembly of Large Telescops*". ESA Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA).

[Let+20]
P. Letier, T. Siedel, M. Deremetz, E. Pavlovskis, B. Lietaer, K. Nottensteiner, M. A. Roa, J. Casarrubios, J. Romero, and J. Gancet (2020): "*HOTDOCK: Design and Validation of a New Generation of Standard Robotic Interface for On-Orbit Servicing*". International Astronautical Congress (IAC).

[Rod+21]
I. V. Rodríguez Brena, A. S. Bauer, K. Nottensteiner, D. Leidner, G. Grunwald, and M. A. Roa (2021): "*Autonomous Robot Planning System for In-Space Assembly of Reconfigurable Structures*". IEEE Aerospace Conference (AeroConf).

[Mar+21]
J. Martínez-Moritz, I. V. Rodríguez Brena, K. Nottensteiner, J.-P. Lutze, P. Lehner, and M. A. Roa (2021): "*Hybrid Planning System for In-Space Robotic Assembly of Telescopes using Segmented Mirror Tiles*". IEEE Aerospace Conference (AeroConf).

[Roa+22]
M. A. Roa, C. E. S. Koch, M. Rognant, A. Ummel, P. Letier, A. Turetta, P. L. Negro, S. Trinh, I. V. Rodríguez Brena, K. Nottensteiner, J. Rouvinet, V. Bissonnette, G. Grunwald, and T. Germa (2022): "*PULSAR, Testing the Technologies for In-Orbit Assembly of a Large Telescope*". ESA Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA).

The research activities leading to the aforementioned publications were partially funded by the European Commission through the Seventh framework programme (project *TAPAS*, grant agreement ID: 260026; project *SMErobotics*, grant agreement ID: 287787) and the Horizon 2020 programme (project *MOSAR*, grant agreement ID: 821996; project *PULSAR*, grant agreement ID: 821858). Furthermore, internal funding has been received from DLR and the DLR project *Factory of the Future*.

3 Methods and Fundamentals

The robotic assembly of products is a complex process involving various steps and components for planning and execution. This chapter introduces basic methods and fundamentals used and developed in this thesis.

First, the overall topic is approached from a high abstraction on task-level in Section 3.1. The robotic skill concept is introduced in this section, as well as the developed framework for flexible robotic assembly in which it plays a central role.

Then, fundamentals of robotic manipulation (Section 3.2), modelling of contacts (Section 3.3), probability theory and Bayesian estimation (Section 3.4) are described in short in order to provide the theoretical background for the developed methods for adapting the process execution.

The connection to the physical assembly process is finally made by the presentation of the developed contact sensing approach and the adaptive execution of tasks under the presence of uncertainties in Section 3.5 and Section 3.6, respectively.

3.1 Task-Level Abstraction and Robotic Skills

In general, describing a goal to be reached or a problem to be solved should be the hardest part, the solution should then come by itself. Naturally, also in robotics research and assembly automation, a driving goal is to develop methods that allow (automatic) generation of robot programs from the specification of desired goals. With such methods, the process of implementation can be transformed to a process of specifying only tasks with goals to be achieved, thus reducing the effort significantly.

Accordingly, at least two levels can be distinguished in robotic applications and programming: Taylor [Tay76, p.2] describes the “task-level” and the “manipulator-level”, whereas the first contains the descriptions of tasks to be performed and the second specifies how to use the

functions provided by the robotic system to solve them.[1] In this context, so-called *task-level languages* were developed to support easy programming and the automatic planning of task sequences from a geometric model using an assembly planner [Wol89]. The search for a solution for a particular task on manipulator-level is traditionally associated with geometrical planning of grasps, gross and fine motions for the process execution [Xia90, p.2f; Wol89, p.62]. The combination of task-level description and planning methods on various levels enables the automatic assembly planning from assembly specifications.

Today, task-level programming approaches are provided through intuitively usable user interfaces, and desired task sequences can even be derived from human demonstration [SWW18; SNS19]. Furthermore, at manipulator-level, the concept of robotic skills is nowadays[2] used to describe and implement reusable robotic capabilities. The concept can be employed for solving tasks and extends the traditional planning approach with further modalities. Skills can be implemented or acquired following various paradigms, for example, through manual implementation by a robotic expert, geometric reasoning, or learning from human demonstration [SV00].

In this work, a framework for flexible robotic assembly is developed, which is able to plan assembly sequences for customized products at task-level and map tasks to reusable robotics skills. The following parts of this section provide a brief overview of assembly sequence planning (Section 3.1.1), the applied concept of robotics skills (Section 3.1.2) and the proposed framework for implementing a flexible robotic assembly system (Section 3.1.3).

3.1.1 From Assembly Specification to Task Sequence

The assembly specification describes the desired product and contains, in its minimal form, a list of parts and their relative poses with respect to a common reference frame. More complex specifications might contain the description of further aspects like tolerances on geometric features, material and surface properties, constraints and kinematic degrees of freedoms, type and strength of connections, or even detailed instructions on how to manufacture certain parts. The assembly specification is typically the output of the product design or construction process and documented with technical drawings or in a digital model generated by computer-aided design (CAD).

Although the product designers ensure that an assembly is in principle feasible, an exact sequence is not always documented, and it is typically not guaranteed that available robotic systems are able to directly realize the associated tasks. The subsequent process of detailing

[1] Similarly, Ejiri et al. [Eji+72, p.161] distinguish between “macroscopic” and “microscopic instructions”. Additional levels of abstraction are sometimes added, e.g., to represent multiple workstations in a production on which the tasks can be distributed [Bar77, p. 23ff]; or when a distinction is made between task-level programs with motions of the end-effector only and execution on joint level [Wol89, p.62]. With respect to robotic assembly planning, [Des89, p.2] specifies four levels: assembly level, task level, robot level and controller level.

[2] See, e.g., [HST92; Tho+03; Bjö+11; Bøg+12; Tho+13; WKW14; Wah+15; SW16].

the setup and implementing the programs is commonly referred to as assembly (and task) planning [Wol89; GRL94].[3]

The framework presented in **Publication 1** deals with the *automation* of assembly and task planning, which is especially critical for reducing the manual effort for customized and one-of-a-kind products. Therefore, the robotic systems should be able to solve assembly problems autonomously and provide early feedback to designers or customers if certain products can be manufactured using the available machinery and robotic systems.[4]

The understanding and planning of assembly tasks is part of robotics research since the 1960s and 70s and was strongly influenced by the rise of artificial intelligence methods at that time. One of the pioneering works is the one by Ejiri et al. [Eji+72], which presents a prototypical robot that is able to autonomously assemble constructions made out of polygonal prisms. It understands assembly goals provided in the form of 2D drawings, recognizes objects in scene images, automatically finds an assembly order and autonomously handles the parts till the goal is reached. Although the approach was restricted to simple geometries, it already showcased fundamental planning techniques like *assembly-by-disassembly*, a method that aims to find a feasible sequence order by starting from the assembled state and trying to remove parts one-by-one.

Furthermore, at that time, fundamental steps towards intelligent robotic systems were taken, for example, in the field of problem-solving using the PLANNER language of Hewitt [Hew70], or natural language understanding in the *blocks world* using SHRUDLU by Terry Winograd [Win71]. This was an inspiration to investigate similar methods to abstract geometric features and use those representations for decision-making in the domain of construction and assembly planning,[5] for example, as done in the BUILD system of Fahlman [Fah74], or in the symbolic description of geometric features and their relations by Ambler and Popplestone [AP75]. Further on, dedicated *task-level languages* were developed to describe and support the (automatic) assembly planning on higher abstraction levels [Wol89, p.62]: AL [Tay76], LAMA [Loz76], AUTOPASS [LW77]. More specific methods for sequencing of assembly operations were then developed later on, for example, by [Bou84; DW87; HS89; Wil92; LW93; Kau+96; HLW00].[6]

While the early approaches focused more on semantic and symbolic representations of the planning problem, the availability of improved CAD programs and collision detection algorithms helped to handle complex geometries and allowed an increase in the number of parts involved in an assembly. The Archimedes 2 system [Kau+96] used a geometric

[3] Note that since a longer time, efforts have been made to reduce the separation between product design and manufacturing engineering also known as the “Over the Wall” design, compare, e.g., [BA92].

[4] In this work, the robotic setup is considered to be fixed; a step towards full assembly planning including the placement of workcell components is given in [BNR21].

[5] Compare the statements by Fahlman [Fah74, p.42, p.140].

[6] See [GRL94, Section 2.3] for an overview of traditional methods in assembly sequence planning and [Jim11; Dan+22, Section II] for more recent surveys.

engine to find optimized sequences and is probably one of the earliest systems that was capable of generating robot control code for complex assemblies derived from CAD files as input. Halperin et al. [HLW00] describe a general framework based on analyzing constraints between subassemblies in the “motion space” and transferring it to a non-directional blocking graph, which can be used for sequence planning.

The sequence planners of [RMW96; MW01; TW01] combine the symbolic spatial relations of [AP75] with CAD data and generate an AND/OR graph [HS90], which contains all possible assembly sequences. Later, in the conceptional line of [HLW00], Thomas et al. [TBW03] integrate an efficient method for sequence planning using stereographic projections of configuration space obstacles. The assembly sequence planning algorithm, which was integrated into the flexible assembly system in **Publication 1**, is an improved version of [TBW03; TW10].[7] The approach combines geometrical checks for assembly motions on part level with a grasp planner to check feasibility of intermediate subassemblies, and allows furthermore to prioritize part types as described in [TSR15].

The exemplary use case investigated in this thesis is the assembly of unique products with parts from a modular aluminum profile construction set. The assembly specification is obtained through an intuitive interface by a user who does not require any knowledge about robotics. Just by demonstrating an arrangement of parts in front of a vision system, the desired product[8] is recognized with its parts and their relative poses. This specification is then passed to the planner, which decomposes the assembly into a sequence of tasks. An example of the output is shown in Fig. 9. Note that at this stage, the task description does not yet contain information about which robotic system to use and which motions to execute, but only the parts and subassemblies to be joined.[9]

The decision about which skill to select for a particular task is made here using a task classification approach as described in **Publication 1**. The approach involves identifying specific patterns in the assembly sequence, which may include properties and relationships of subassemblies from the current, previous, and future assembly steps. In the presented application, the pairs of subassemblies in each assembly step are classified into specific group types. The combinations of these group types then determines the type of assembly task. For the solution of each of the possible task types a skill sequence is known in advance,[10] and can be used to map each task in the sequence to the available robotics skills. These skills provide capabilities of the robotic system to solve tasks as described in the following.

[7] Sequence planning was further investigated in [Rod+19; Rod+20] using the dual-arm version of the system.

[8] Later in the dual-arm system, this was replaced by a graphical user interface on a tablet with similar functionalities, see [Rob20a].

[9] Further available information from the geometry, as the approach direction, is not used in this work.

[10] The initial assignment of skills to task types needs to be done only once and there are multiple possible ways to do it. Right now this is implemented manually by the robotic expert in the form of a classification and mapping function. However, this principal assignment can be solved automatically as well, e.g., by using a logic planner and modeling the preconditions and effects of each skill as done by Sürig [Sür21].

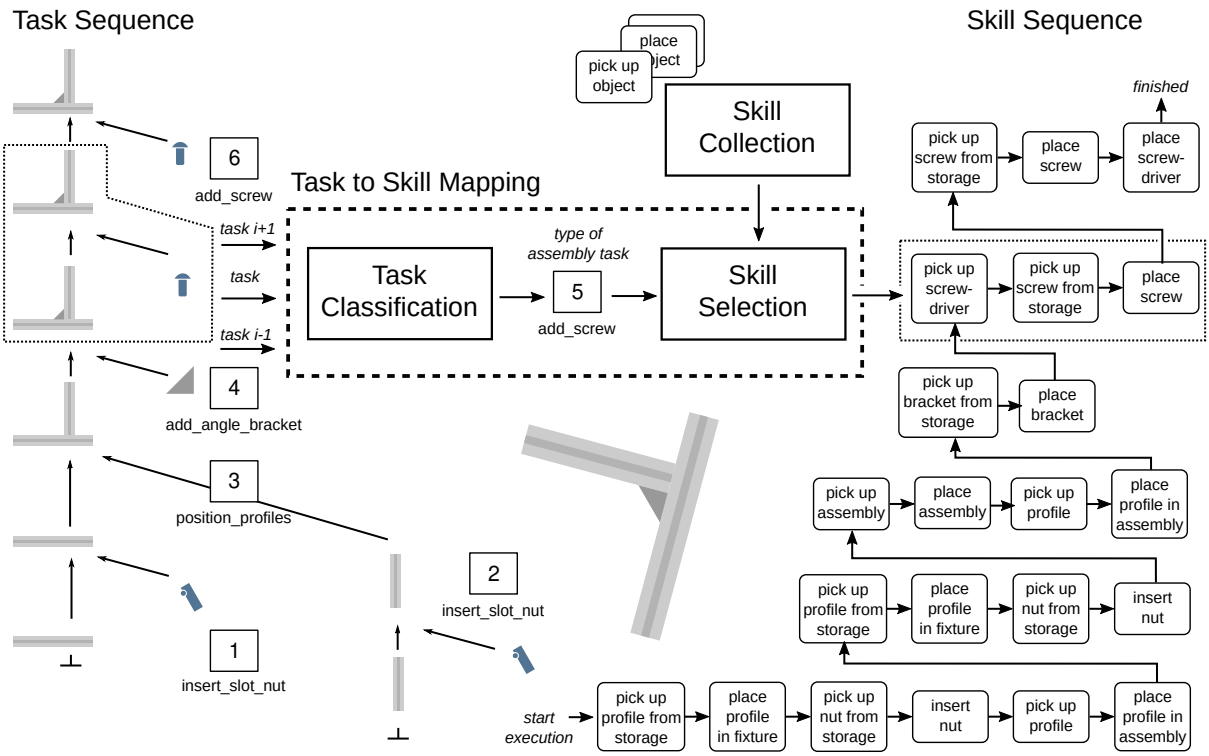


Figure 9: Mapping of assembly tasks to robotic skills for a one-of-a-kind product. The task sequence is generated by a dedicated assembly sequence planner using CAD information of the product and then analyzed stepwise to assign skills for the execution of the assembly process. Reuseable skills are selected and parameterized according to the type of assembly task.

3.1.2 Reusable Object-Centric Robotic Skills

The concept of robotic skills is introduced in multiple related works with slightly varying definitions, for example, by [HST92; Tho+03; Bjö+11; Bøg+12; Tho+13; WKW14; Wah+15; SW16].[11] In this work, a *robotic skill* is considered to be a "robotic behavior that reaches desired goal states in multiple situations and under varying conditions" (**Publication 4**, p. 3). An important aspect connected to this skill definition is the high *reusability*, which is also highlighted by the work of Björkelund et al. [Bjö+11]. Furthermore, skills are typically implemented *object-centric* or *object-centered* [Bøg+12], namely, using features of objects as motion references instead of absolute coordinates provided in a global coordinate frame.[12] This way it is for example possible to use perceived feature poses as an online input for motion generation instead of only relying on offline pre-determined static frames in absolute coordinates.[13] In summary, robotic skills can adapt to various situations without human intervention, in contrast to traditional industrial implementations of robot applications with fixed programmed solutions for specific tasks.

Skills are located in the abstraction level between task-level planning and low level functions of the robots and devices in a workcell, so-called *device primitives*.[14] It should be noted that the separation between skills and device primitives cannot always be clearly determined, but some guidelines can be formulated. A skill typically includes goal-directed interaction with an environment and uses feedback and sensory information to reason and adapt the behavior. On the other hand, a device primitive typically has a local scope, internal state changes, and no clear effect on the environment in varying situations. For example, a humanoid hand might have device primitives to open and close the fingers, but the outcome is unknown under varying conditions. Whereas a grasping-skill implements a behavior that has the clear effect of achieving a stable grasp for various object types at different locations. The larger the variety of possible conditions under which a skill can be successfully executed, the clearer it can be distinguished from a device primitive and the more *powerful* it is.

In the developed robotic assembly system, skills with different abstraction levels are collected in a modular skill library (**Publication 1**, Section 6.1). Skills at lower level can be used to compose hierarchical skills at a higher abstraction level (as suggested by, e.g., [Bøg+12; Tho+13]). An example at the lowest level is a skill that allows the robot to move collision-free within a known environment to a goal pose as parameter. The implemented version of this example uses a motion planner connected to a world state representation and standard

[11] A more general discussion about skills (or abilities) of agents is also carried out in philosophy, e.g., see [Jas20].

[12] Both terms, *object-centric* and *object-centered*, are often used with similary meaning. In this work, the term *object-centric* is used.

[13] An early example for the object-centric paradigm is explained by Taylor [Tay76, p.22ff]: motions are defined relative to a "fiducial mark", which could be obtained from a vision system.

[14] In **Publication 1** they are called *device primitives*. In literature, they are furthermore called *motion primitives* [Bøg+12], *elemental actions* [Tho+13] or similar. In the specification of the Factory of the Future Ontology [Sch+21a, Section 7.1] we refer to *atomic functions* in this context to generalize and include as well non-hardware and non-motion primitives.

device primitives of the robot such as point-to-point motions. The higher abstraction levels in the skill library provide the interface to the task-level description of the assembly planner, for instance, by referring directly to the objects to be manipulated as symbolic parameters. An example is a skill that enables the robot to pick up an object from a storage device, which is associated with a particular step in the overall assembly sequence. The only necessary parameter is the reference to the part in the assembly plan; where to find the object in the workspace and the motions to pick it up are determined within the skill.

The proposed skill model in **Publication 1** (Section 6.2) is inspired by the skill concept of Bøgh et al. [Bøg+12]. Similar to it, skills take a set of parameters as input and have an *execute-mode*, which is responsible for achieving the desired state changes in the real world. A difference is that less importance is given to (logic) pre-condition checks at the beginning of each skill execution. Of course, certain critical states are checked throughout the execution (e.g., if the gripper is free, an object is grasped, etc.), but overall, an optimistic approach is taken, which assumes that the known skill sequence is able to solve a particular task in the given situation. For the case that a skill sequence leads to an error, feedback can be provided to the deliberative task planning components in order to find a feasible solution.[15] For synchronizing local state changes of multiple objects, the skills are connected to a central world state representation [LBH12]. An alternative to execute it in the actual world is to switch to a virtual world by selecting the *simulate-mode*, which works on virtual representations of the objects and system components, but uses the same skill implementations. This can be used for pre-computing relevant data in the skills, and furthermore, to validate the execution of assembly plans on particular robot systems by checking their feasibility as done together with Rodríguez et al. [Rod+19]. Additionally, the skill model features an *observe-mode* to monitor the progress of the state changes, for example, by providing the estimated distance to the motion goal. Methods for state estimation during the execution of assembly skills are developed in this work and explained in more detail in Section 3.5. The implementation of adaptive motion generation in an assembly skill is described in Section 3.6.

3.1.3 Framework for Flexible Robotic Assembly

As described above, the framework in this work combines deliberative planning on task-level with reusable object-centric robotic skills for flexible execution of assembly tasks. The robotic skills are at the core of the software framework and are responsible for collecting and merging the information from the distributed components, and the coordination of the actions of all components of the robotic system during execution. The processes of the components are connected via the communication interfaces offered by the DLR RMC middleware links and nodes (LN) [SB14]. The main components of the overall framework are depicted in Fig. 10 and listed with their main functions in the following:[16]

[15] This is investigated for the dual-arm extension of the system in [Rod+19; Rod+20].

[16] Note that only the contributions of the author described in Section 1.4 and Chapter 4 are part of this dissertation.

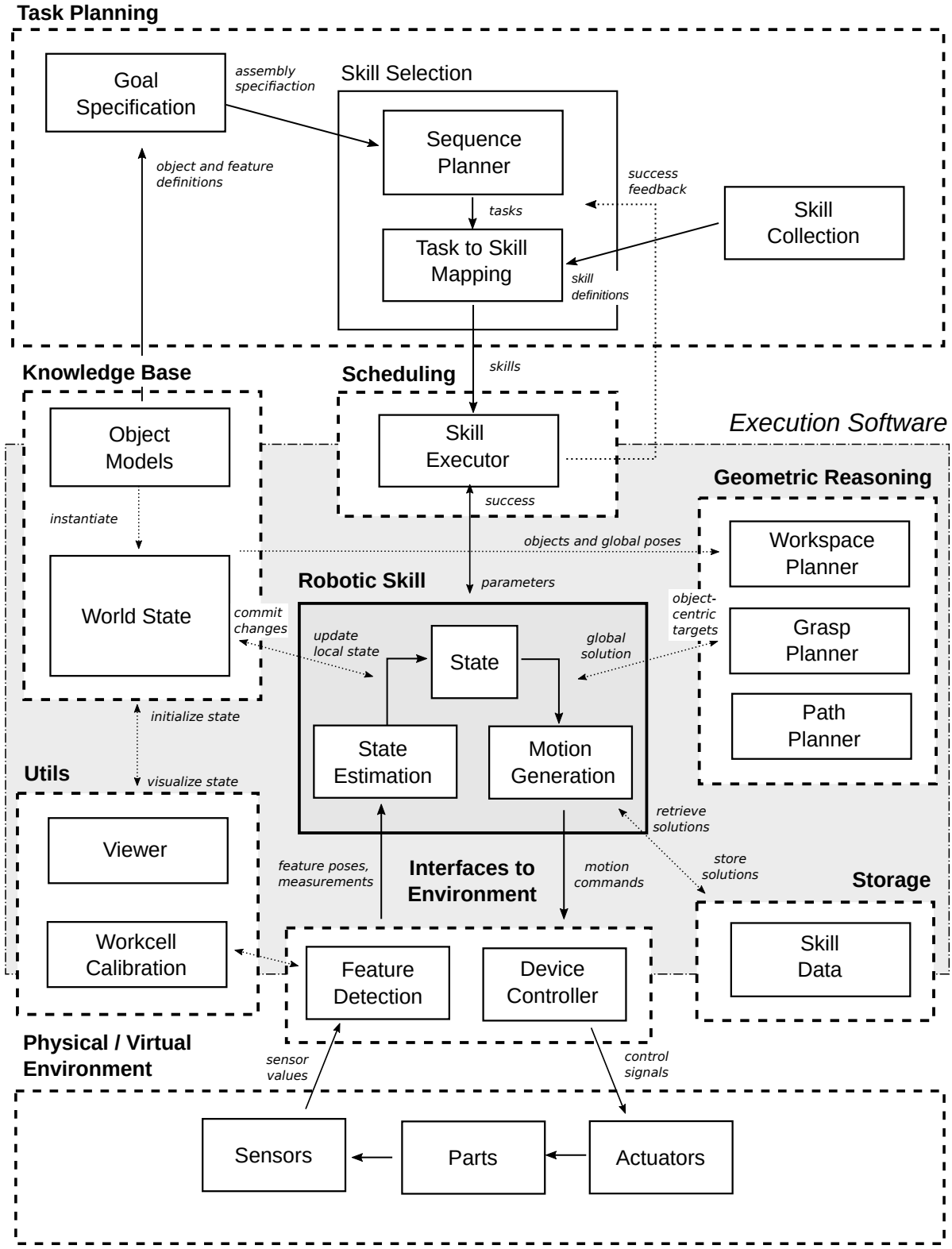


Figure 10: Components of the framework for flexible robotic assembly. Source: Expanded diagram of Fig. 2 of **Publication 4** with additional components (Utils, Storage, Geometric Reasoning, Environment) and additional aspects of Fig. 1 and Fig. 8 of **Publication 1**

- **Task Planning**: The assembly sequence planner by Thomas et al. [TSR15] is used in the presented work to decompose the specified assembly into tasks, which are then mapped to skills provided by the skill collection using the described task classification approach. In the later dual-arm version of the system, the assembly sequence planner got replaced by a new implementation as described in [Rod+19].
- **Knowledge Base**: Object information and knowledge about the world state are provided centrally by two components that are developed in their original form by Leidner et al. [LBH12, Sec. III].[17] The knowledge base contains physical properties and states, as well as pure symbolic states of objects. The dynamics of the physical objects are by default not modelled, that is, the objects are represented in a static state. However, the properties and states can be updated by the skills, which can observe a local state change during the execution in the actual environment, and commit it to the central world state. Known physical side effects and events that lead to state changes, for example, parts sliding down in a gravitation-based feeder after the removal of a part, can be applied locally through external simulation processes such that (potentially unobservable, but known) state changes are represented as well. The functionality of the world representation was furthermore enhanced such that assemblies can be represented as groups of parts and that available parts in the workcell are mapped to parts referenced in the assembly specification.[18] If the skills are executed in the real physical environment (execute-mode), then the component keeps track of all observed or known state changes in the real workcell. If the skills are executed in the virtual environment (simulate-mode), then the world state tracks the state of the virtual environment (but is itself not the simulation of the virtual environment).
- **Scheduling**: A skill execution engine is implemented in Java on top of the (proprietary) robot controller. It takes descriptions of skill sequences in XML-format as input, which includes skill type and parameters, and abstracts the implementations from the pure motion commands. The engine allows the execution either in the mentioned simulation- or execution-mode. For this purpose, all relevant processes of the execution software are duplicated in the LN setup such that they are available in either case and can be accessed in parallel.[19] In the later dual-arm extension, RAFCON [Bru+16] is used as a state machine to control the overall execution flow on top of the skill executor with an interface to the task planning unit.
- **Robotic Skill**: As mentioned above, the skill is at the core of the execution framework. The skill executor instantiates a skill of the required type and passes the parameters

[17] In this work, the “action templates” of the world representation [LBH12] are not used.

[18] Each object in the world representation has a unique identifier. At planning time it is usually not yet known which individuals are present later during the assembly process. Therefore, a mapping between identifiers of objects in the plan to the identifiers of the objects in the workcell needs to be done.

[19] The framework supports a multitude of virtual environments distributed over the network so that theoretically an arbitrary number of product plans can be generated and validated in simulation in parallel.

from the planning unit. The skill itself is responsible to orchestrate all necessary functions to achieve the desired effects. For this purpose, it can interact with all other software components via LN. Local state changes are synchronized with the central world state representation. If uncertainties are present, then a state estimator as described in Section 3.5 can be employed. Based on the desired effect the skill generates the desired motion commands; see Section 3.6 for the developed adaptive motion generation methods in this work.

- **Geometric Reasoning**: In order to provide the skill with geometric reasoning capabilities, various planning components can be used. These are needed because the motion targets of a skill are in general referenced with respect to the objects following the object-centric paradigm. However, the poses of the objects change during the execution of the application, and therefore the local targets need to be resolved in the global context of the world state. For this purpose, the skill can query geometric solutions from these reasoning components.

  For example, an approach pose for the tool is given with respect to the object coordinate frame and not with respect to the robot base frame. A path planner provides then the functions to generate collision free-paths from the current pose of the tool to the desired approach pose considering the motion obstacles registered in the current world state. A motion planner was at first provided by KUKA within the SMErobotics project. For the dual-arm version this component was replaced by RMPL [LA17].

  Additionally, a runtime interface to the grasp planner of [TSR15] provides the capability to determine grasps for individual assemblies. During the assembly planning at task-level, graspability is in general verified for all subassemblies. However once a plan is executed on a particular robotic system, the world state needs to be incorporated, and grasps need to be selected that are reachable for the actual poses of the objects.

  Furthermore, in the dual-arm system, the part placement is also decided online using a workspace analysis and planning component [Bac+20]. The reachability of object-centric motion paths can be verified and parts can be placed such that manipulability is optimized.

- **Interfaces to Environment**: The execution software interacts through dedicated interfaces with the environment. These provide functions (i.e., device primitives) to send motion commands through device controllers to the actuators of the robot hardware, and furthermore, functions to retrieve feature poses and sensor measurements. Notably, the interface components are able to communicate with the real hardware or alternatively with a virtual representation if the skill is executed in the simulate-mode. For this purpose, each relevant object in the environment is abstracted by an implementation that can use an interface to the virtual or to the real object.

In this work, the virtual representations are however simplified and do not simulate the dynamics of the real objects or hardware. In future work, the degree of realism could be increased by a full physics simulation of the virtual environment. However, in the presented form, it is sufficient to check feasibility on a kinematic level (compare [Rod+19]).

- **Utilities**: Multiple software utilities are available to improve the usability, efficiency, and the comfort during setup of the system. The viewer DLR MediView is used to visualize the world state and the robot movements. Furthermore, a software tool for workcell calibration enables the intuitive manual teach-in of workcell components. Methods for automatic self-calibration of components are investigated together with Sachtler et al. [Sac16; Sac+19]. By default, it is assumed that the poses of part storages and fixtures are initially known from the manuel teach-in and calibration process. However, in this dissertation, it is investigated how remaining or newly arising uncertainties can be estimated using tactile sensing. With the help of such methods, the initial calibration effort can then be reduced.
- **Storage**: Each new individual product to be manufactured obtains a dedicated storage place in the file system to keep and document data from planning and execution persistently. Data generated by a skill can be stored inside a tree structure that represents the task sequence and skill hierarchy. For example, path data generated in a simulated run in the virtual environment can be kept for later reuse in the actual run on the real system. Furthermore, the world state at the beginning and at the end of the skill execution is stored by default for error recovery.

3.2 Robotic Manipulation

This section introduces basic methods and fundamentals in the field of robotic manipulation, which are required to describe and implement robotic skills in this work. In general, various types of manipulation can be distinguished. Mason and Lynch [ML93, p. 152] suggest a *taxonomy of manipulation* that includes: kinematic manipulation, static manipulation, quasi-static manipulation, and dynamic manipulation. In this work, although dynamic effects occur during the real execution and the applied low-level impedance controller considers flexible joint models, a description of the assembly problems will be used that is based on static manipulation, meaning, “operation[s] that can be analyzed using only kinematics and static forces.” [ML93, p. 152][20]

[20] In the same line, the term *kinestatics* was introduced by Staffetti [Sta09, p.80]: ”[...] kinestatics [...], the statics and instantaneous kinematics of a rigid body constrained by one to six contacts with a rigid static environment.”

Therefore, the description of rigid body motions using screw theory will be introduced at first from a general view point in Section 3.2.1. This is followed by the modeling of kinematic chains with details on robotic manipulators with serial kinematics in Section 3.2.2. Finally, a brief introduction to compliance and force control methods is provided in Section 3.2.3.

3.2.1 Rigid Body Motions

The theorem by Chasles [Cha30; MLS94, p. 49] and further works on rigid body displacements[21] conclude that a rotation about a certain axis in combination with a translation in the direction of that axis is sufficient to describe every possible motion of a rigid body.

The *theory of screws* by Sir Robert Ball [Bal98] is based on this insight and provides the methods to describe rigid body motions and forces in a unified framework using the concept of a *screw*. The works of Hunt [Hun78], Lipkin [Lip85], Murray et al. [MLS94], and Husty et al. [Hus+97], and others, introduced screw theory in the field of robotics for modeling the kinematics and statics of manipulators. The description of rigid body motions in this section uses methods of screw theory and is mainly based on the book by Murray et al. [MLS94] with minor differences in the notation.

First of all, for the purpose of providing a reference, a right-handed, orthonormal coordinate frame is attached to each rigid body or to dedicated features rigidly connected to a body. The position and the orientation of a coordinate frame A of a body with respect to an inertial frame B, represented by a position vector ${}_B\boldsymbol{r}_{BA} \in \mathbb{R}^3$ and a rotation matrix $\boldsymbol{R}_{BA} \in SO(3)$, form together the *configuration* $\boldsymbol{g}_{BA} = ({}_B\boldsymbol{r}_{BA}, \boldsymbol{R}_{BA}) \in SE(3)$ of the frame A relative to the frame B. The special Euclidean group $SE(3) = \mathbb{R}^3 \times SO(3)$ is also called *configuration space* of a rigid body [MLS94, p.35]. The homogeneous representation of the configuration is given by the 4×4-*transformation matrix*:

$$\boldsymbol{H}_{BA} = \begin{bmatrix} \boldsymbol{R}_{BA} & {}_B\boldsymbol{r}_{BA} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}. \tag{3.1}$$

If the configuration of frame D relative to frame A is known, then the configuration of D relative to frame B can be composed using matrix multiplication: $\boldsymbol{H}_{BD} = \boldsymbol{H}_{BA}\boldsymbol{H}_{AD}$. Furthermore, it applies $\boldsymbol{H}_{BA}{\boldsymbol{H}_{BA}}^{-1} = \boldsymbol{I}_4$, where $\boldsymbol{I}_4$ denotes the 4×4 identity matrix, and the inverse matrix of the transformation is given by:

$${\boldsymbol{H}_{BA}}^{-1} = \begin{bmatrix} \boldsymbol{R}_{BA}^T & -\boldsymbol{R}_{BA}^T{}_B\boldsymbol{r}_{BA} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{AB} & {}_A\boldsymbol{r}_{AB} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} = \boldsymbol{H}_{AB}, \tag{3.2}$$

using the general property of rotation matrices: $\boldsymbol{R}_{BA}^{-1} = \boldsymbol{R}_{BA}^T = \boldsymbol{R}_{AB}$.

[21] See the historical review by Dai [Dai06].

The *twist* of a rigid body describes its instantaneous motion with respect to an inertial frame D. It consists of the angular velocity ${}_D\boldsymbol{\omega} \in \mathbb{R}^3$ and the linear velocity component ${}_D\boldsymbol{v} \in \mathbb{R}^3$. In general, the instantaneous motion can be interpreted as a screw motion. The angular velocity vector defines the direction of the screw axis. The linear velocity component is the sum of the linear velocity ${}_D\boldsymbol{v}_{\parallel}$ in direction of the axis and the linear velocity ${}_D\boldsymbol{v}_{\perp}$ induced by the rotation about the axis measured in the origin of the inertial reference frame D: ${}_D\boldsymbol{v} = {}_D\boldsymbol{v}_{\parallel} + {}_D\boldsymbol{v}_{\perp}$, with ${}_D\boldsymbol{v}_{\parallel} = h \cdot {}_D\boldsymbol{\omega}$ and ${}_D\boldsymbol{v}_{\perp} = [{}_D\boldsymbol{r}]_{\times D}\boldsymbol{\omega}$. Here, $h \in \mathbb{R}$ is the pitch parameter that describes the coupling of linear and angular velocity of the screw motion, $h = {}_D\boldsymbol{\omega}^T {}_D\boldsymbol{v}/|{}_D\boldsymbol{\omega}|^2$. The position vector ${}_D\boldsymbol{r} \in \mathbb{R}^3$ points from the origin of D to an arbitrary chosen point on the screw axis. The vector norm is denoted $|\bullet|$, and $[\bullet]_{\times}$ denotes the conversion of a vector to a skew-symmetric matrix according to

$$[\boldsymbol{r}]_{\times} = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}. \tag{3.3}$$

The angular velocity and the linear velocity vector can be combined to a single vector ${}_D\boldsymbol{t} = [{}_D\boldsymbol{v}^T, {}_D\boldsymbol{\omega}^T]^T \in \mathbb{R}^6$ called the *twist coordinates*.[22] The twist coordinates can be converted to another (inertial) reference frame by using the adjoint operation:

$${}_C\boldsymbol{t} = \mathrm{Ad}(\boldsymbol{H}_{CD}) {}_D\boldsymbol{t}, \tag{3.4}$$

where the adjoint matrix is given with

$$\mathrm{Ad}(\boldsymbol{H}_{CD}) = \begin{bmatrix} \boldsymbol{R}_{CD} & [{}_C\boldsymbol{r}_{CD}]_{\times} \boldsymbol{R}_{CD} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{R}_{CD} \end{bmatrix}. \tag{3.5}$$

The twist describes a velocity field from which displacements can be computed, which represent changes of the configuration of a rigid body. Formally, a twist is an element of the Lie algebra $se(3)$ of the Lie group $SE(3)$, which are connected via the *exponential map*.[23] Accordingly, a rigid body transformation $\boldsymbol{H}_{EF}$ can be obtained from the exponential map of a twist:[24]

$$\boldsymbol{H}_{EF} = \exp([{}_E\boldsymbol{t}]\Delta T), \tag{3.6}$$

[22] More specifically, the components are denoted here in *axis coordinates* order of a screw; in contrast to the *ray coordinates* order in which the components are interchanged, compare [Hun03, p. 326; LD85, p. 378].

[23] See, e.g., the appendix of [MLS94] for details; furthermore, [SDA18] provides a compact summary to important concepts of Lie theory for robotics.

[24] Here the frame F can be seen as a frame that coincides with frame E at the beginning of the motion, i.e., at time $t = t_{k-1}$: $E = F_{k-1}$ and $\boldsymbol{H}_{EF_{k-1}} = \boldsymbol{I}_4$; then at $t = t_k$: $\boldsymbol{H}_{EF_k} = \exp([{}_E\boldsymbol{t}]\Delta T)$ with $\Delta T = t_k - t_{k-1}$. If the frames do not coincide, then the change of the frame F due to the screw motion can be computed with: $\boldsymbol{H}_{EF_k} = \exp([{}_E\boldsymbol{t}]\Delta T)\boldsymbol{H}_{EF_{k-1}}$ according to [MLS94, (2.37) on p.42].

where ΔT is the duration of the screw motion with constant velocity, and $[{}_E\boldsymbol{t}] \in se(3)$ is the (unnormalized) twist, and written in the form of its matrix representation:

$$[{}_E\boldsymbol{t}] = \begin{bmatrix} [{}_E\boldsymbol{\omega}]_\times & {}_E\boldsymbol{v} \\ \mathbf{0}_{3\times 3} & 0 \end{bmatrix}. \tag{3.7}$$

Analogous to velocities, forces and moments can be represented in screw theory using *wrenches*. Similar to a twist, a wrench uses the concept of a screw. Here, the direction of the screw axis is defined by the direction of the force ${}_D\boldsymbol{F} \in \mathbb{R}^3$. The location of the screw axis can be specified by an arbitrary point on the axis. The moment ${}_D\boldsymbol{M} \in \mathbb{R}^3$ is composed of the moment ${}_D\boldsymbol{M}_\perp$ induced by the force and an additional moment ${}_D\boldsymbol{M}_\parallel$ parallel to the screw axis. The *wrench coordinates* are given by the vector ${}_D\boldsymbol{w} = [{}_D\boldsymbol{M}^T, {}_D\boldsymbol{F}^T]^T \in \mathbb{R}^6$. The reference frame can be changed as well using the adjoint matrix ${}_A\boldsymbol{w} = \mathrm{Ad}(\boldsymbol{H}_{AD}){}_D\boldsymbol{w}$.

The *instantaneous work* or power generated by a wrench $\boldsymbol{w}$ acting on a body moving with twist $\boldsymbol{t}$ can be computed in any inertial reference frame using the *reciprocal product*:

$$P = \boldsymbol{w}^T \boldsymbol{\Delta} \boldsymbol{t} = \boldsymbol{M}^T \boldsymbol{\omega} + \boldsymbol{F}^T \boldsymbol{v}, \tag{3.8}$$

where $\boldsymbol{\Delta}$ denotes the *interchange operator* introduced by Lipkin [Hun03, p. 328f; LD82, p. 365; Lip85, p. 33f]:

$$\boldsymbol{\Delta} = \begin{bmatrix} \mathbf{0} & \boldsymbol{I}_3 \\ \boldsymbol{I}_3 & \mathbf{0} \end{bmatrix}. \tag{3.9}$$

Note that the instantaneous work is invariant with respect to the chosen reference coordinate frame. In the case that it vanishes ($P = 0$), the body is in equilibrium and the screw pair of twist and wrench are *reciprocal* [Bal98, p.26].[25]

3.2.2 Kinematic Chain of a Serial Robotic Manipulator

Using the concepts of screw theory, the kinematics and statics of a robotic manipulator can be described. A robotic manipulator consists of links connected by joints that provide degrees of freedom such that the links can move relative to each other [Cra09, p. 5]. In this work, serial manipulators are considered in which the links are connected one after the other with revolute joints forming an open kinematic chain. In particular, robotic arms of the LWR-type [Alb+07] with $n = 7$ joints are investigated.

Throughout this work, the frame B is located at the center of the robot base by convention, which is rigidly attached to the environment. The frame A is located at the last link of the robotic arm. The frame D represents the reference frame of a manipulated or grasped

[25] The term *reciprocity* goes back to the work of [Bal98]; Ohwovoriole and Roth [OR81] introduces furthermore the notion of *contrary* and *repelling* screws (using the *virtual coefficient* of [Bal98, p. 17]).

object, and is typically located at one of the geometric features of the object. An object in the environment has the reference frame C (compare Fig. 13).

The configuration of the robotic arm is given by its joint positions $\boldsymbol{q} = [q_1, ..., q_n]^T \in \mathbb{R}^n$. The *forward kinematics* of the robotic arm can be computed by the *product of exponentials* formula by Brockett [Bro84; MLS94, p. 85ff]:

$$\boldsymbol{H}_{BA}(\boldsymbol{q}) = \exp([{}_B\boldsymbol{t}_1]\ q_1) \cdot ... \cdot \exp([{}_B\boldsymbol{t}_n]\ q_n)\boldsymbol{H}_{BA}(\boldsymbol{0}). \tag{3.10}$$

The homogeneous transformation between frame B and A at time $t = t_k$ will be denoted as $\boldsymbol{H}_{BA,k} := \boldsymbol{H}_{BA}(\boldsymbol{q}_k)$ with $\boldsymbol{q}_k = \boldsymbol{q}(t_k)$.[26]

In (3.10), the twists ${}_B\boldsymbol{t}_i$ of the joints $i \in \{1, 2, ..., n\}$ are normalized and represent *unit twists*. For the case of having only revolute joints, the twists are zero pitch screws ($h = 0$) with normalized angular velocities $|{}_B\boldsymbol{\omega}_i| = 1$ pointing into the direction of the joint axes:

$$[{}_B\boldsymbol{t}_i] = \begin{bmatrix} [{}_B\boldsymbol{\omega}_i]_\times & -[{}_B\boldsymbol{\omega}_i]_\times {}_B\boldsymbol{r}_{BJ_i} \\ \boldsymbol{0}_{1\times 3} & 0 \end{bmatrix}, \tag{3.11}$$

where ${}_B\boldsymbol{r}_{BJ_i}$ connects B with a point J_i on the joint axis i (at the joint configuration $\boldsymbol{q} = \boldsymbol{0}$).

The *Jacobian matrix* of a manipulator contains the information about the possible motions given the current joint configuration. Each column of the Jacobian ${}_B\boldsymbol{j}_i(\boldsymbol{q})$ is a unit twist associated with a single joint, which is transformed from the initial reference configuration to the current configuration of the robot using the adjoint operation in combination with the product of exponentials [MLS94, p. 116]:

$${}_B\boldsymbol{j}_i(\boldsymbol{q}) = \mathrm{Ad}(\exp([{}_B\boldsymbol{t}_1]\ q_1) \cdot ... \cdot \exp([{}_B\boldsymbol{t}_{i-1}]\ q_{i-1}))_B\boldsymbol{t}_i, \tag{3.12}$$

with ${}_B\boldsymbol{t}_1, ..., {}_B\boldsymbol{t}_{i-1}$ written in matrix representation (3.11).

The Jacobian ${}_B\boldsymbol{J}(\boldsymbol{q}) = [{}_B\boldsymbol{j}_1(\boldsymbol{q}), ..., {}_B\boldsymbol{j}_n(\boldsymbol{q})] \in \mathbb{R}^{6\times n}$ with reference frame B is called *spatial* manipulator Jacobian. Whereas the *body* manipulator Jacobian refers to the frame A and can be obtained with the help of the adjoint operation [MLS94, p. 117]:[27]

$${}_A\boldsymbol{J}(\boldsymbol{q}) = \mathrm{Ad}(\boldsymbol{H}_{BA}(\boldsymbol{q}))^{-1}{}_B\boldsymbol{J}(\boldsymbol{q}). \tag{3.13}$$

[26] For convenience in the notation: functions can only be identified by the presence of an argument, i.e., variables in brackets. For example, constant transformations between two frames R and S will be denoted with $\boldsymbol{H}_{RS} = \text{const.} \in SE(3)$; functions that map a variable $\boldsymbol{q} \in Q$ to a transformation are written as $\boldsymbol{H}_{RS}(\boldsymbol{q}) \in SE(3)$. So a transformation without argument is always considered a constant value and no function. The outcome of a function associated with a value $\boldsymbol{q}_k$ of the argument can be marked with a dedicated subscript, e.g., $\boldsymbol{H}_{RS,k}$.

[27] Note that A is viewed here as an inertial frame that coincides with the last link at the current instance of time, but has no own motion.

The resulting velocity of a manipulator motion is then the combination of joint motions, specifically, the linear combination of the columns of the Jacobian with the joint velocities $\dot{\boldsymbol{q}} = [\dot{q}_1, ..., \dot{q}_n]^T \in \mathbb{R}^n$ as factors.[28] The summation written as matrix multiplication gives:

$$ {}_A\boldsymbol{t} = \sum_{i=1}^{n} {}_A\boldsymbol{j}_i(\boldsymbol{q}) \cdot \dot{q}_i = [{}_A\boldsymbol{j}_1(\boldsymbol{q}), ..., {}_A\boldsymbol{j}_n(\boldsymbol{q})]\ \dot{\boldsymbol{q}} = {}_A\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3.14} $$

In the implementation, the joint velocity at time t_k is approximated using $\dot{\boldsymbol{q}}_k \approx (\boldsymbol{q}(t_k) - \boldsymbol{q}(t_k - \Delta T_{rob}))/\Delta T_{rob}$ with sample time $\Delta T_{rob} \in \mathbb{R}$ of the joint position measurement.

An external wrench acting on the end effector at frame A can be mapped to an equivalent manipulator joint torque $\boldsymbol{\tau} = [\tau_1, ..., \tau_n] \in \mathbb{R}^n$ using the Jacobian:[29]

$$ \boldsymbol{\tau} = {}_A\boldsymbol{J}^T(\boldsymbol{q}_k)\boldsymbol{\Delta}_A\boldsymbol{w}. \tag{3.15} $$

Such external wrenches are typically induced by the contact of the manipulator with the environment, for example, during execution of an assembly task. The modeling of contacts and the used implementation of a contact model will be described in detail in Section 3.3.

3.2.3 Compliance and Force Control

Compliance and force control methods deal with the behavior of the manipulator being in contact with the environment. *Compliance* in robotics is defined as the “flexible behavior of a robot or any associated tool in response to external forces exerted on it” [ISO12, p. 20]. For realizing such a compliant behavior, dedicated hardware adapters were developed, which can be attached to the end of the manipulator and provide a passive compliant mechanical structure to compensate geometric uncertainties during assembly (e.g., by Drake [Dra78] and Watson [Wat78]). Furthermore, compliance can be achieved by sensory feedback in a control loop. *Force control* provides methods for achieving stable and robust compliant behavior and consequently enables successful handling of contacts between the robot and the environment [VD16, p.195]. An overview of force control methods for robotic assembly is provided by Marvel and Falco [MF12].

In this work, the implementation of robotic assembly skills is based on *impedance control*, which was proposed by Hogan [Hog84] as a unifying framework for robotic manipulation in free space and especially in contact with the environment. Hogan distinguishes two fundamental types of physical systems: *admittances*, which in the context of mechanical systems take a force as input and can respond with a motion; and *impedances* which have a motion input and respond with a force. Furthermore, he suggests to control deviations of a robot from a desired motion with the help of impedances in order to “ensure physical compatibility” during

[28] Compare also the derivation of the *screw Jacobian* by Lipkin and Duffy [LD82, p.364].
[29] This can be derived using the principal of virtual work, compare [LD82, p. 365f; MLS94, p. 121].

the interaction with static rigid objects in the environment, which cannot move and can only be represented as admittances [Hog84, p. 305].[30]

Following this concept, such impedance behavior can be implemented in a very basic form through a spring-like model that relates position errors as input to a force response as output, for example:

$$F_x = K(x_d - x). \tag{3.16}$$

Here, $F_x \in \mathbb{R}$ is a force acting in the x-direction of a one-dimensional system with current coordinate x, the motion reference is provided as a desired equilibrium position x_d, and $K \in \mathbb{R}$ is a (virtual) linear stiffness constant. A deviating motion consequently produces a force pointing back towards the equilibrium point. In practice, the value of the force F_x can be used as input for the low-level control interface of a system actuator to generate this force effect.

The basic principle in (3.16) can be expanded to systems with multiple degrees of freedom, such as robotic manipulators. Impedance controllers for robotic manipulators can be implemented on joint level and in Cartesian coordinates.[31] Furthermore, also nonlinear stiffness can be modeled and additional components can be added, for example, a damping term relating velocity errors to forces, or a term to compensate the gravity effect on the dynamics.

An impedance control framework was established by Albu-Schäffer and Ott [AOH07; Ott+08] for robots with flexible joints and redundant kinematics. In particular, the DLR LWR [Alb+07] is a torque-controlled robot, which uses this impedance control framework to enable sensitive interaction with the environment. The feedback provided by the integrated joint torque sensors (besides measurements of the motor positions) allows the implementation of a torque control loop on joint level, which reduces friction effects and can be interpreted as “shaping of the motor inertia” [AOH07, p.25]. A (Cartesian) impedance controller can then be used on top of the low-level torque controller and allows a task dependent parameterization of the spatial stiffness. The performance of the control framework has been demonstrated in multiple scenarios including human-robot collaboration and sensitive execution of assembly tasks.

Due to its relevance in the industrial domain, the technology of the DLR LWR was transferred to the company KUKA [Bis+10] and became part of the product *LBR iiwa* (iiwa), which had its premiere on the 2013 Hannover Fair [KUK13]. The implementation of this work is based on the controllers and interfaces provided by the *KUKA RoboticsAPI* of the iiwa robot.

[30] The interaction between two systems along a certain degree of freedom is only possible if the systems physically complement each other: “Along any degree of freedom, if one is an impedance, the other must be an admittance and vice versa” [Hog84, p. 305].

[31] However, an appropriate choice needs to be done with respect to the representation of orientations, e.g., compare [Cac+99].

3.3 Rigid Body Contact Representation

The assembly of parts involves the sequential closure of contacts until the final pose of a part according to the assembly specification is reached. During this transition from a non-contact configuration to the final pose, parts get successively constrained in multiple spatial directions. At each stage of this assembly process, a particular *fine motion* of the manipulator must be executed to continue. The necessary knowledge about geometric constraints, feasible motions, and physical effects can be provided by contact models.

In this section, a subset of general contact modeling is presented. Although friction and local deformations are effects that influence the execution in practice, the focus lies mainly on modeling frictionless contacts between rigid bodies.[32]

First, the relative configuration space between rigid bodies in contact is introduced in Section 3.3.1. Then, a brief review on representing constrained motions using twists and wrenches is provided in Section 3.3.2. Finally, the contact model based on haptic rendering used in the implementation of this work is described in Section 3.3.3. Fig. 11 shows an overview of these contact representations.

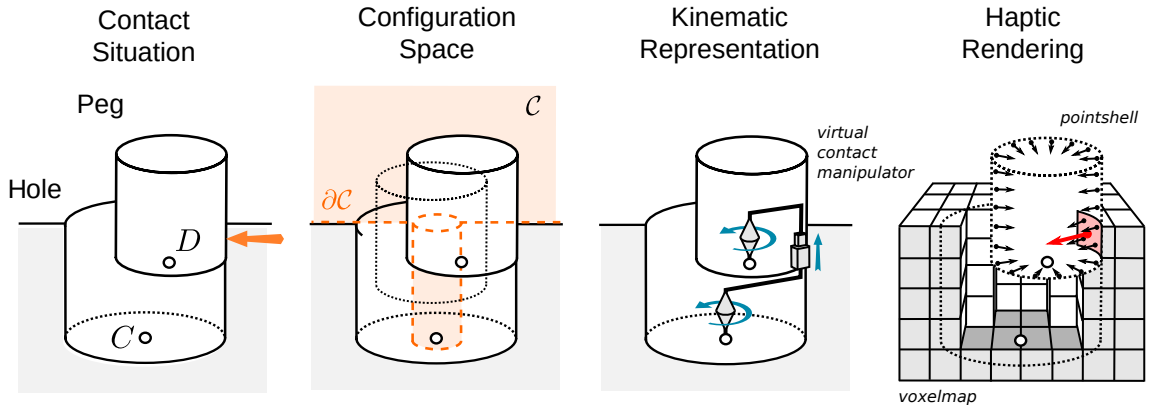


Figure 11: Overview of considered rigid body contact representations. The real contact situation corresponds with the relative pose of the parts on the configuration space border. Geometric constraints can be described using a kinematic representation. Specialized data structures, for instance, voxelmaps and poinshells, are used in haptic rendering to efficiently compute contact forces.

3.3.1 Configuration Space of Rigid Bodies in Contact

In general, the *configuration* is a set of (generalized) coordinates that uniquely describes the locations of every point of a physical system. The possible values of the coordinates span the configuration space. For example, the locations of every point of a robotic manipulator can be

[32] A general overview of contact modeling considering contact dynamics and other effects is provided, e.g., by [GS02] and [FL16].

described uniquely using the joint positions. Thus, the configuration space of a robot can be associated with a space that contains combinations of all possible joint positions.[33]

Furthermore, as stated above in Section 3.2.1, the configuration of a rigid body is given by a tuple consisting of a position and an orientation, and accordingly, the configuration space is the $SE(3)$ with 6 degrees of freedom. For example, a possible set of coordinates for a rigid body can contain 3 coordinates for the position and 3 Euler angles [MLS94, p.31ff; Cra09, p.43ff] for the orientation [Loz81, p.682; Mas01, p.12].

However, the motion of a single object is constrained by the presence of other objects in the environment. As suggested in the approach for motion planning by Lozano-Pérez [Loz83], the configuration space can be divided into a space in which objects are free to move, and spaces in which other objects block the motion (i.e., obstacles). The associated *configuration space obstacles* can be derived from the geometry of the objects [Loz83].

In this work, the relative pose of an object, which might be attached to the end-effector of a robotic manipulator, with respect to another object in the environment is called the *relative configuration* between the objects. The relative configuration between two objects at time $t = t_k$ with reference frame C and D can be represented by a homogeneous transformation matrix $\boldsymbol{H}_{CD,k} =: \boldsymbol{H}_k$.

The relative configuration space $\mathcal{C}$ is then defined implicitly using the contact distance $d_k = d(\boldsymbol{H}_k) \in \mathbb{R}$ between the surfaces of the objects:

$$\begin{cases} \boldsymbol{H}_k \in \mathcal{C} & : d_k \leq 0 \\ \boldsymbol{H}_k \notin \mathcal{C} & : d_k > 0. \end{cases} \tag{3.17}$$

The contact distance is measured along the contact normal defined by the surface normals of the objects. If the objects do not intersect, then the contact distance represents the shortest possible distance between the surfaces, in the other case the maximal distance inside of the intersecting volume. Note that the convention is used that the distance is defined to be positive for intersecting objects. The boundary of the relative configuration space ($d_k = 0$) is denoted by $\partial\mathcal{C}$ and represents closed contacts between the objects.[34]

The boundary of the relative configuration space can be subdivided into regions according to the number and types of occurring contacts between the objects surfaces. Such a region is called *contact formation (CF)* and can be defined using a set of *elemental contacts* [Des89;

[33] Note that the configuration space of the end-effector for all possible joint configurations is also called *workspace* of a manipulator [MLS94, p. 95]. In this work it is assumed that the considered object motions are within the workspace of the manipulator, which can be verified using methods for workspace analysis as done together with Bachmann et al. [Bac+20].

[34] Contacts are sometimes also modeled as unilateral constraints using a complementary condition, the so-called *Signorini-(Fichera) condition* $g \geq 0, \lambda \geq 0, g \cdot \lambda = 0$, where $g = -d$ is the gap, i.e., contact distance, and $\lambda \in \mathbb{R}$ is the intensity of a normal reaction force; see, e.g., [Gei+06, p. 796; KLB16, p. 1208].

DV89, p.802; Tan07, p.43].[35] The set of possible CFs can be automatically computed from the geometry of the objects and stored together with the information of adjacent regions in contact state graphs [XJ01; Tan07]. Each CF can be associated with a *kinestatic* description as will be described in the following Section 3.3.2.

3.3.2 Twist and Wrench Spaces of Contacts

The instantaneous motions of rigid bodies in contact are directly related to the local contact geometry. Waldron concludes, based on the reciprocity condition of Ball [Bal98] (see Section 3.2.1), that the possible motions of a rigid body in contact can be derived from the contact geometry [Wal72, p.347]. In general, the *twist space* and the *wrench space* of a constrained body are dual screw systems. This *principle of duality* is summarized by Lipkin and Duffy [LD82, p.364] as follows: "a rigid body in space that can twist about n independent screws ($1 \leq$ n ≤ 6) has the same instantaneous mobility when constrained by (6 - n) independent reciprocal wrenches". Accordingly, the twist space is a vector space of n-dimensions and can be represented as matrix in which the columns are the basis vectors given as independent unit twists: $\boldsymbol{T} = [\boldsymbol{t}_1, ..., \boldsymbol{t}_n] \in \mathbb{R}^{6\times n}$. A twist of a constrained object that maintains the contact can therefore be represented as a linear combination of the basis vectors. Similarly, the wrench space is represented as matrix consisting of unit wrenches as basis vectors: $\boldsymbol{W} = [\boldsymbol{w}_1, ..., \boldsymbol{w}_{6-n}] \in \mathbb{R}^{6\times(6-n)}$, and all reaction forces can be obtained from it as linear combination. Following the principle of duality, each combination of basis vectors of the wrench and twist space needs to fulfill the reciprocity condition (3.8) [BDD93, p.351].

The matrix $\boldsymbol{T}$ can be interpreted intuitively as Jacobian matrix of a so-called virtual contact manipulator (VCM) as suggested by Bruyninckx et al. [Bru+95; Bru95]. This allows the use of the same modeling methods as known from robotic manipulation and, furthermore, representing the motion space up to the second order, which enables accurate estimation of geometric uncertainties. In the general case of a single point contact between curved surfaces, the VCM consists of 5 revolute joints, one in the contact point aligned with the contact normal and for each body two joints that directly relate to the principal curvatures of the surfaces in contact. The wrench space is one-dimensional and defined by a unit force along the contact normal. Other contact types as vertex, line and plane contacts can be handled as limit cases. If multiple parallel contacts are present then an intersection of the twist systems is required [Bru+95, p.471]. An approach to representing multiple contacts using Grassman-Cayley algebra, which allows obtaining closed-form expressions for such intersections, is presented by Staffetti [Sta09] and was used by the author for contact state recognition using analytical contact models in [Not12].

[35] Elemental contacts are also called *principal contacts* in the context of polyhedral objects [Xia93, p. 66] as well as for curved objects [Tan07, p.42]; and should not be confused with *elementary contacts*, which denote the three types of contacts between polyhedrons (face-vertex, vertex-face, edge-edge)[Mee+08, p.438].

The exact kinematic information about contacts helps to identify uncertainties and setup controller models. Meeussen et al. [Mee+07, p.225] present a method using singular value decomposition (SVD) to obtain twist and wrench spaces from elementary contacts between polyhedrons. The knowledge about the twist space is also used in this work to implement a constrained-based propagation step in the observation algorithm (see Section 3.5.3).

However, in practice, the accurate computation of the twist space is expensive if mesh models are the only input for the algorithm and the geometries are complex (fine features, convex, large size etc.). Common collision detection libraries only provide information about the contact point and possibly the contact normal. The next Section 3.3.3 describes the default contact model used in this work for handling contacts between mesh models.

3.3.3 Contact Model Implementation Using Haptic Rendering

According to Gilardi and Sharf [GS02], contact models can be divided into *discrete* and *continuous* models, where the first are mainly used to model short impacts between rigid bodies on impulse level, and the latter have advantages in representing flexible multi-body systems with non-impact dynamic problems, including the possibility to use various types of contact dynamics models. In the continuous case, the contact force can be written as an explicit function of the local deformation and its derivatives [GS02, p.1227, p.1231]. For this purpose, implementations using explicit contact models need to be able to detect if contacts between objects occur and compute the minimal distance or penetration depth from the geometries.

In this work, the haptic rendering algorithm for collision and distance computation of Mikel Sagardia [Sag19] is used as basis of the applied contact model. The implementation is based on the penalty-based voxelmap-pointshell algorithm (VPS) by McNeely et al. [MPT99] and optimizes it in several aspects to obtain results faster and more accurate. It can answer collision queries with complex and non-convex geometries in less than 1 ms. The algorithm uses two specialized types of data structures to represent the objects and to efficiently support the computation of collisions. Both can be generated offline from triangle mesh data. The *pointshell* consist of fine sampled surface points, where each point is associated with a surface normal vector. The *voxelmap* discretizes the volume of an object into cubic elements and corresponds to a signed distance field. Typically, objects in the environment are represented by voxelmaps and the manipulated objects by pointshells. Once two objects intersect, the basic algorithm computes a penalty force by averaging the normals of the pointshell weighted by the individual penetration depths of the points obtained from the distance field in the voxelmap [Sag19, p.76]. Especially for planar contact cases, the resultant force can be interpreted as buoyancy force as it is directly proportional to the intersecting volume [Sag19, p.77f].

Given a relative transformation $\boldsymbol{H}_{CD}$ between two rigid bodies, VPS computes multiple properties of the contact. The relevant quantities in this work are the penalty force ${}_D\boldsymbol{F}_P(\boldsymbol{H}_{CD}) \in \mathbb{R}^3$ and, furthermore, the contact distance $\tilde{d}(\boldsymbol{H}_{CD}) \in \mathbb{R}$ associated with the deepest point of the pointshell in contact Q (or the closest point in the case of no penetration). The applied contact model in this work uses only the directional information of ${}_D\boldsymbol{F}_P$ to compute a virtual contact wrench ${}_D\tilde{\boldsymbol{w}} = {}_D\tilde{\boldsymbol{w}}(\boldsymbol{H}_{CD})$. The resultant force is shifted such that it is acting on the deepest point Q instead of on the (approximated) center of the intersecting volume.[36] Together with an explicit linear elastic stiffness model $F_C(\boldsymbol{H}_{CD}) = k_C \cdot \tilde{d}(\boldsymbol{H}_{CD}) \in \mathbb{R}; k_C = \text{const.} \in \mathbb{R}^+$ the wrench is then computed according to:

$$ {}_D\tilde{\boldsymbol{w}}(\boldsymbol{H}_{CD}) = \begin{bmatrix} {}_D\tilde{\boldsymbol{M}} \\ {}_D\tilde{\boldsymbol{F}} \end{bmatrix} = F_C \cdot \begin{bmatrix} [{}_D\boldsymbol{r}_{DQ}]_\times \frac{{}_D\boldsymbol{F}_P}{|{}_D\boldsymbol{F}_P|} \\ \frac{{}_D\boldsymbol{F}_P}{|{}_D\boldsymbol{F}_P|} \end{bmatrix}, \tag{3.18} $$

where ${}_D\boldsymbol{r}_{DQ} \in \mathbb{R}^3$ is the position vector from the object reference frame D to the deepest intersecting point Q of the pointshell.

Furthermore, the computed contact distance can be used to implicitly define the configuration space between the objects in the virtual contact model. As the algorithm requires the intersection of the geometries to generate a contact force, a minor penetration must be allowed, and thus, the theoretical configuration space (3.17) needs to be expanded. Consequently, in this work, a configuration space $\tilde{\mathcal{C}}$ is defined for the virtual contact model:

$$ \begin{cases} \textit{no contact}, & \boldsymbol{H}_{CD} \in \tilde{\mathcal{C}} & : \tilde{d} < 0 \\ \textit{contact}, & \boldsymbol{H}_{CD} \in \partial\tilde{\mathcal{C}} & : 0 \leq \tilde{d} < d_t \\ \textit{invalid}, & \boldsymbol{H}_{CD} \notin \tilde{\mathcal{C}} & : d_t < \tilde{d}. \end{cases} \tag{3.19} $$

The offset d_t expands the boundary of the configuration space to a contact zone $\partial\tilde{\mathcal{C}}$ where feasible contacts can be established. It is clear that the range of force intensities is bounded by d_t, that is, the width of the zone, and the assumed virtual contact stiffness k_C. In practice, the values of these parameters are chosen such that the contact model can represent the force intensities that are expected during assembly execution.

Overall, the used contact model is a simplification of the physical effects of a real contact. However, it provides fast and reasonable information for contact sensing, which also takes model and measurement errors in the observation algorithm into account (see Section 3.5).

[36] This improves the reproduction quality of the moment induced by the normal contact force in edge contacts. This can be illustrated in a planar case in which a slightly rotated rectangle is in contact with a plane. The force naturally acts on the corner of the rectangle. To obtain a higher intensity of the force, the penetration depth in the model must be increased. In the default model, an increase of the depth would shift the point on which the force acts to the center of the intersecting area (buoyancy). In the here used model, the force still acts on the corner as it is the deepest point (and uses the directional information of the buoyancy) and is therefore closer to the real effect of the contact. A drawback might be that the model is more sensitive to small variations in the orientation, which produce larger changes in the moment as in the default case.

3.4 Bayesian State Estimation

The general idea of *Bayesian reasoning* or *Bayesian inference* is to update prior knowledge, taking into account newly available information from observations, in such a way that the uncertainty contained in the prior knowledge is ideally reduced. Probability theory permits describing uncertainties with probabilities and probability density functions, and provides a mathematical framework for Bayesian reasoning. In particular, the *Bayes's theorem* relates the prior with a posterior distribution using the likelihood of observing a certain value given a state hypothesis.

The Bayesian approach is nowadays a central method in *state estimation* and (nonlinear) filtering to reduce uncertainties and estimate hidden states of various systems. A general introduction for engineers into filtering using the Bayesian approach for stochastic processes is given by the well known work of Jazwinski [Jaz70]. The history of Bayesian filtering is covered in large parts by the comprehensive survey of [Che03]. Bayesian methods for the application in object tracking are collected and described by [Cha+11]. The contact sensing approach in this work builds upon these methods as will be described later in Section 3.5.

In this section, the fundamentals of probability theory and Bayesian state estimation are presented in short. First, a brief review of basic terms in probability theory is provided, covering probability density functions and fundamental relations as the Bayes' theorem (Section 3.4.1). Then, an introduction into recursive Bayesian estimation is given (Section 3.4.2). The last part describes the basics of particle filtering (Section 3.4.3), which is at the core of the state estimator implemented in this work. Fig. 12 shows the main step of the recursive algorithm for state estimation.

3.4.1 Probability Theory and Density Functions

In this section, basic terms and relations of probability theory are briefly introduced, which are used for modeling stochastic processes and uncertainties in this work. The text in this section is partially based on the introduction by [Jaz70; Chi09, p.64ff; Tab17].

In general, the value of a *random variable* X depends on the outcome of a probabilistic experiment, is not fixed and a priori unknown. Multiple random variables can be grouped into a *random vector* $\boldsymbol{X} = [X_1, ..., X_n]^T$, also called multivariate random variable.

In the case of a *continuous random vector* $\boldsymbol{X} \in \mathbb{R}^n$ with an uncountable number of possible values, there exists a *probability density function* $p_{\boldsymbol{X}}(\boldsymbol{x}) \geq 0 \; \forall \; \boldsymbol{x} \in \mathbb{R}^n$ according to which the vector values are distributed: $\boldsymbol{X} \sim p_{\boldsymbol{X}}(\boldsymbol{x})$.[37] In contrast to $\boldsymbol{X}$, the vector $\boldsymbol{x} = [x_1, ..., x_n]^T \in \mathbb{R}^n$ denotes a concrete and clear value, also called *realization* of the random vector [Jaz70,

[37] It can be also seen as a *joint* probability density function of the (scalar) jointly distributed random variables $p_{\boldsymbol{X}}(\boldsymbol{x}) = p_{X_1,...,X_n}(x_1, ..., x_n)$; see, e.g., [Jaz70, p.31f].

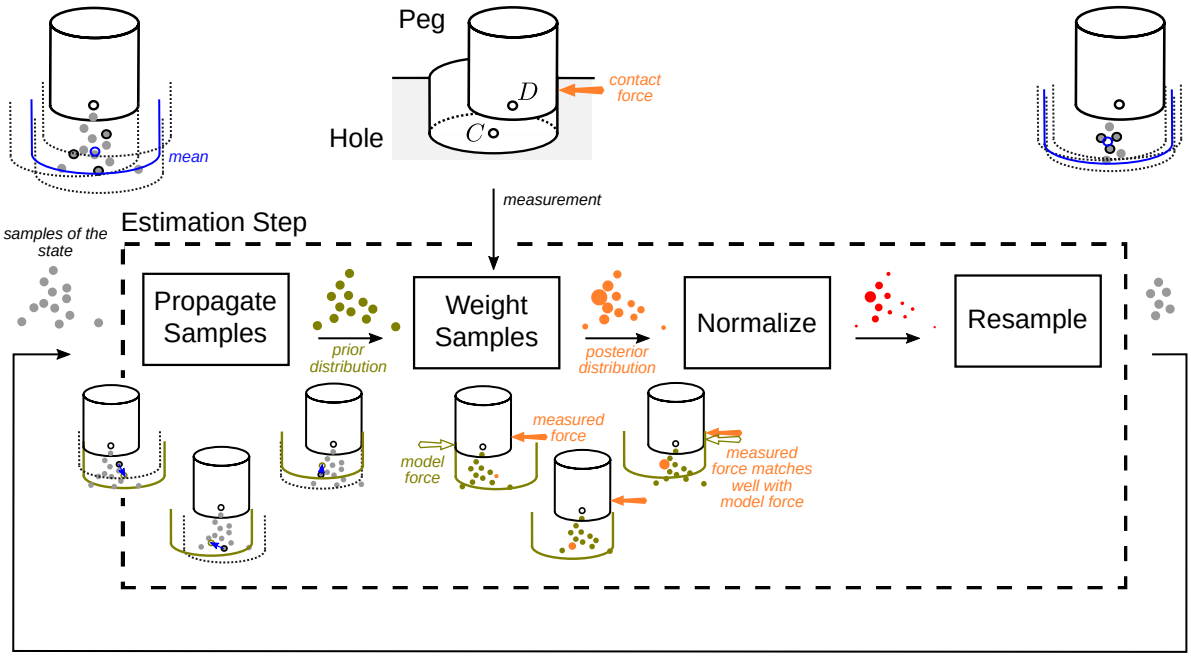


Figure 12: Recursive Bayesian state estimation using a particle filter. Samples of the state are propagated and weighted according to the observations made from the real system. An estimate of the hole pose can be computed as the mean of the sample values. A resampling step selects the best fitting samples for the next iteration of the algorithm.

p.13]. By definition, such probability density functions integrate to unity $\int_{\mathbb{R}^n} p_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} = 1$ with $d\boldsymbol{x} = dx_1 dx_2 ... dx_n$.

The *probability* that the outcome of the random experiment lies within in a certain region $\boldsymbol{X} \in D_n \subset \mathbb{R}^n$ is given by the integral:

$$Pr[\boldsymbol{X} \in D_n] = \int_{D_n} p(\boldsymbol{x})\, d\boldsymbol{x}. \tag{3.20}$$

Furthermore, the expectation or *expected value* of a function $h(\boldsymbol{X})$ is given by:[38]

$$E[h(\boldsymbol{X})] = \int_{\mathbb{R}^n} h(\boldsymbol{x})\, p(\boldsymbol{x})\, d\boldsymbol{x}. \tag{3.21}$$

In particular, the *mean* value of a random vector $\boldsymbol{X}$ is obtained from the expected value of the random vector itself:

$$\boldsymbol{\mu} := E[\boldsymbol{X}] = \int_{\mathbb{R}^n} \boldsymbol{x}\, p(\boldsymbol{x})\, d\boldsymbol{x}, \tag{3.22}$$

and the *covariance* by $\boldsymbol{\Sigma} := E[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})^T]$.

[38] The expected value $E[h(\boldsymbol{X})]$ of a function $h(X)$ is sometimes also denoted as $\hat{h}$ or $\langle h \rangle$.

The *normal distribution*, or also Gaussian distribution, is relevant in many areas of probability theory and the description of stochastic processes. In the multivariate case, the density of the normal distribution is given by the function:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) := p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{3.23}$$

with the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ and the mean $\boldsymbol{\mu} \in \mathbb{R}^n$.

In the special case of a diagonal covariance matrix $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_n^2)$, in which the random variables are mutually independent, see (3.26), the density function can be written as product of univariate normal density functions $\mathcal{N}(x_i|\mu_i, \sigma_i^2)$:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \operatorname{diag}(\sigma_1^2, \ldots, \sigma_n^2)) = \prod_{i=1}^{n} \mathcal{N}(x_i|\mu_i, \sigma_i^2), \tag{3.24}$$

where $\boldsymbol{\mu} = [\mu_1, ..., \mu_n]^T$, and $\sigma_i := \sqrt{E[(X-\mu_i)^2]} \in \mathbb{R}$ denotes the *standard deviation*.

The probability density function of a univariate normal distribution is given as

$$\mathcal{N}(x|\mu, \sigma^2) := p_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\Big(\frac{x-\mu}{\sigma}\Big)^2\right). \tag{3.25}$$

Note that the first argument x is often omitted when referencing the density itself and not a particular value of it. For example, when expressing that a random variable is distributed according to a normal distribution: $X \sim \mathcal{N}(\mu, \sigma^2)$.

Random variables $X_1, ...X_n$ are *mutually independent* if their joint probability density function satisfies [Jaz70, p.28, (2.67)]:

$$p_{X_1,...,X_n}(x_1, ..., x_n) = p_{X_1}(x_1) \cdot ... \cdot p_{X_n}(x_n). \tag{3.26}$$

Furthermore, $\boldsymbol{X} = [X_1, ...X_n]^T$ and $\boldsymbol{Y} = [Y_1, ...Y_m]^T$ are (jointly) independent, if their joint probability $p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y})$ is given by [Jaz70, p.28]:

$$\begin{aligned} p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y}) &= p_{X_1,...,X_n,Y_1,...,Y_m}(x_1, ..., x_n, y_1, ..., y_m) \\ &= p_{X_1}(x_1) \cdot ... \cdot p_{X_n}(x_n) \cdot p_{Y_1}(y_1) \cdot ... \cdot p_{Y_m}(y_m) \\ &= p_{\boldsymbol{X}}(\boldsymbol{x}) \cdot p_{\boldsymbol{Y}}(\boldsymbol{y}). \end{aligned} \tag{3.27}$$

The *marginal density function* of a joint density is given by:

$$p_{\boldsymbol{Y}}(\boldsymbol{y}) = \int_{\mathbb{R}^n} p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x}. \tag{3.28}$$

The *conditional density function* $p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y})$ is the "the fundamental entity" in filtering for stochastic processes in which $\boldsymbol{x}$ represents the state of the system and $\boldsymbol{y}$ is the available measurement [Jaz70, p. viii]. In general, the conditional density function of $\boldsymbol{X}$ *given* $\boldsymbol{Y}$ can be obtained from the joint and the marginal densities [Jaz70, p.36ff]:[39]

$$p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) = \frac{p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y})}{p_{\boldsymbol{Y}}(\boldsymbol{y})} = \frac{p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y})}{\int_{\mathbb{R}^n} p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x}}. \tag{3.29}$$

Rearranging the first part of (3.29) provides a generalized version of (3.27), which is sometimes also referred to as the *product rule*:[40]

$$\begin{aligned} p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y}) &= p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) \cdot p_{\boldsymbol{Y}}(\boldsymbol{y}) \\ &= p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}) \cdot p_{\boldsymbol{X}}(\boldsymbol{x}). \end{aligned} \tag{3.30}$$

Finally, this leads to the equation of the well-known *Bayes' theorem* for continuous random vectors [Jaz70, p.39, (2.102)]:

$$p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}, \tag{3.31}$$

which plays the key role in (recursive) Bayesian estimation as will be presented in the next section.

Note that (3.29),(3.30) and (3.31) are in some standard references only presented for continuous random variables, but in general, they apply to random vectors as well, as pointed out by Jazwinski [Jaz70, p.39].

For convenience in the notation, the subscripts naming the random vectors in the probability density functions will be omitted in the following, and it will be only referred to the realizations of the random vectors in the arguments, for example: $p(\boldsymbol{x},\boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{y}) \cdot p(\boldsymbol{y})$. Furthermore, the notation will not strictly distinguish between random vectors and their realization.[41]

3.4.2 Recursive Bayesian Estimation Methods

Recursive Bayesian estimation methods can be used to estimate the state of systems whose dynamics can be represented as *stochastic processes*.[42] The underlying models use random variables to model uncertainties and random effects in the state evolution. In this

[39] Jazwinski shows there how the conditional density function for continuous random vectors can be derived starting from the *conditional probability* $Pr[A|B] = Pr[A \cap B]/Pr[B]$ with the random events A and B.

[40] For independent random vectors it applies $p_{\boldsymbol{X}}(\boldsymbol{x}) = p_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y})$ and $p_{\boldsymbol{Y}}(\boldsymbol{y}) = p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x})$; (3.27) can then be obtained from (3.30).

[41] See [Jaz70, p.47; DFG01, p.5; Chi09, p.65] for comparable conventions in notation.

[42] See Table 3.1 of [Jaz70] for a classification of stochastic processes.

work, continuous state spaces with a discrete time parameter are considered, which can be described with nonlinear state space models of the form [CGM07, p.899]:

$$\boldsymbol{x}_k = a(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k), \tag{3.32}$$

$$\boldsymbol{y}_k = b(\boldsymbol{x}_k, \boldsymbol{n}_k), \tag{3.33}$$

where (3.32) is the *system model* with the random vector $\boldsymbol{x}_k$ that represents the state at time $t = t_k, k \in \mathbb{N}_0$, and which is modeled as a function of the state $\boldsymbol{x}_{k-1}$ of the previous time step and the state disturbance $\boldsymbol{u}_k \sim p(\boldsymbol{u}_k)$. The *measurement model* is given by (3.33) with the available measurement $\boldsymbol{y}_k$ as a function of the current state and the noise $\boldsymbol{n}_k \sim p(\boldsymbol{n}_k)$.

Associated with the system model is a transition probability density function or *transition density* $f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \; \forall \; k > 0$, which also describes the state changes and can for example be obtained through rearranging (3.32) or from experiment data and model identification; similarly, the observation probability density function or *observation density* $g(\boldsymbol{y}_k|\boldsymbol{x}_k)$ is associated with the measurement model [CGM07, p.899]. The initial state is assumed to be distributed according to the initial state density $p(\boldsymbol{x}_0)$. Examples for system models of this category are provided by [Kit87; DFG01; DJ11; CGM07].

Such a model of a stochastic process is also called a Markovian model or *hidden Markov model (HMM)* as it assumes that the state evolution only depends on the values from the previous time step and not on values further in the past and, furthermore, that the state itself is hidden, meaning that it is not directly observable. Consequently, the HMM with the Markovian assumption can be represented as well using the transition and the observation density [CGM07, p.900]:

$$\boldsymbol{x}_k \sim f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}), \tag{3.34}$$

$$\boldsymbol{y}_k \sim g(\boldsymbol{y}_k|\boldsymbol{x}_k). \tag{3.35}$$

In general, a goal in Bayesian estimation is to estimate the posterior distribution $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$ of a state trajectory, and in particular, its marginal distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$, which is the state at the current time based on the knowledge of the values of all past measurements.[43] The latter problem is typically called *filtering*, and the first (fixed interval) *smoothing* [CGM07, p. 900]. The set of (hidden) states until time $t = t_k$ is here denoted as $\boldsymbol{x}_{0:k} := \{\boldsymbol{x}_0, ..., \boldsymbol{x}_k\}$, and the set of all past measurements as $\boldsymbol{y}_{1:k} := \{\boldsymbol{y}_0, ..., \boldsymbol{y}_k\}$.

The theoretical solution for estimating the posterior distribution $p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$ follows from the Bayes' theorem, and is described, for example, by Doucet et al. [DFG01, p.6]. Using the Bayes' theorem and the Markov assumption, a recursive method can be derived for estimating the marginal distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ as well, as will be described briefly in the following.

[43] The posterior is in some works also called *belief*, denoted as $bel(\boldsymbol{x}_k)$; especially in robot localization and motion planning.

An iteration of a recursive method in Bayesian estimation consist conventionally of a *prediction* and an *update step* (compare, e.g., Kitagawa [Kit87, p.1033]). The prediction step provides the *prior* density $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$. It can be derived by applying (3.28), (3.30), and using the Markovian system model (3.34) together with the estimate $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$ from the previous time step:

$$\begin{aligned} p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) &= \int_{\mathbb{R}^n} p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}) d\boldsymbol{x}_{k-1} \\ &= \int_{\mathbb{R}^n} f(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})\, p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{1:k-1}) d\boldsymbol{x}_{k-1}. \end{aligned} \tag{3.36}$$

This prediction can then be corrected in the update step, taking into account the newly available measurement $\boldsymbol{y}_k$ to obtain an estimate of the marginal distribution, namely, the *posterior* density $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k})$. The posterior is computed using Bayes' theorem (3.31) and the Markovian assumption:

$$\begin{aligned} p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k}) &= p(\boldsymbol{x}_k \mid \boldsymbol{y}_k, \boldsymbol{y}_{1:k-1}) \\ &= \frac{p(\boldsymbol{x}_k, \boldsymbol{y}_k \mid \boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k \mid \boldsymbol{y}_{1:k-1})} \\ &= \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k, \boldsymbol{y}_{1:k-1})\, p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1})}{p(\boldsymbol{y}_k \mid \boldsymbol{y}_{1:k-1})} \\ &= \frac{g(\boldsymbol{y}_k \mid \boldsymbol{x}_k)\, p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1})}{\int g(\boldsymbol{y}_k \mid \boldsymbol{x}_k)\, p(\boldsymbol{x}_k \mid \boldsymbol{y}_{1:k-1}) d\boldsymbol{x}_k}. \end{aligned} \tag{3.37}$$

Note that (3.37) is a function of $\boldsymbol{x}_k$ as the observations $\boldsymbol{y}_{1:k}$ are assumed to be known from all past measurements carried out. Furthermore, the denominator does not depend on $\boldsymbol{x}_k$ and represents a normalizing constant.[44] For a known value $\boldsymbol{y}_k$, the observation density $g(\boldsymbol{y}_k \mid \boldsymbol{x}_k)$ as function of $\boldsymbol{x}_k$ is also called *likelihood function* (compare, e.g., [Cha+11, p.9]).

The prediction (3.36) and update (3.37) step can be called recursively starting with an initial prior density $p(\boldsymbol{x}_0)$. However, closed-form solutions of the equations only exist in special cases, for example, where the equations are linear and densities are represented with Gaussians [DJ11, p.4]. The Kalman filter [Kal60] for stochastic processes with zero-mean Gaussian noise falls into this category and an optimal estimate of the (hidden) state can be obtained.[45]

Because of these limitations in finding a closed-form solution, several approaches have been developed to handle systems with nonlinear stochastic processes, for example: extended Kalman filter (EKF) [Jaz70; AM79], Gaussian quadrature Kalman filter (QKF) [IX00],

[44] The equation is sometimes simplified to a proportional relation written as $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \propto p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$, compare, e.g., [Cha+11, p.9; IB96, p.344].

[45] Ho and Lee [HL64] derive solutions for this type of stochastic processes using a Bayesian approach.

unscented Kalman filter (UKF) [JU04], probability hypothesis density (PHD) filter [Mah03], Gaussian sum filter [SA70; AS72], point mass and grid-based filters [BS71], numerical approximation of density functions [Kit87], (sequential) Monte Carlo and particle methods [HM69; AK75; ZSŠ75; Del94; GSS93; KKR95; IB96; Kit96]. An overview of the various (nonlinear) filtering methods is given by [CGM07, p.900f; DFG01, p.6; Cha+11, p.22ff].

In particular, the sequential Monte Carlo (SMC) methods have advantages due to their support of multi-modal and non-Gaussian distributions, and nonlinear system dynamics, as well as their straight-forward implementation without the need for closed-form solutions. These methods became especially popular in robotics for localization and contact state estimation (e.g., [TBF05; GLB05]), and are also used in this work for contact sensing. The next section presents the basics of the SMC framework in more detail.

3.4.3 Particle Filtering

The basic idea of SMC and particle filtering methods goes back to the insight that the outcome of complex processes for which no closed-form solutions exist, or for which solutions are too expensive to compute, can be investigated statistically by simulating a set of samples in individual runs; each sample represents alternative (start) conditions as suggested by the Monte Carlo method of Stan Ulam [MU49; Eck87; DPV13, p.2]. The outcome of all simulations can then be used to find an approximation for the actual system behavior.

The idea was later adopted in the field of nonlinear filtering [HM69; AK75; ZSŠ75; DGA00, p.197] and had its breakthrough in the nineties with the works of [GSS93; Del94; KKR95; IB96; Kit96]. While the early approaches had only little impact in practice, the so-called *bootstrap filter* of Gordon et al. [GSS93] is commonly considered to be the "first successful application of sequential Monte Carlo techniques to nonlinear filtering" [CGM07, p.900]. Del Moral [Del96] provided the first theoretical convergence results on this class of approaches.[46] Today, multiple variants and extensions of the method are available, for example, the Rao-Blackwellized particle filter [DGA00; AD02; SGN05] and the auxiliary particle filter [PS99]. Furthermore, efficient libraries and techniques for the (parallel) implementation exist, for example, by [Joh09; HKG10; Chi+13; Zho14; Dem+14; Lin+17].

In the following, this section introduces the basics of particle filtering. It is mainly based on the introductions and tutorials by [DFG01; CGM07; DJ11]. Furthermore, a recent hands-on tutorial on particle filtering is provided by [ETM21].

The basic idea of particle filtering is to approximate probability density functions by a set $\mathcal{X}_k$ of weighted samples of the state (i.e., particles): $(\boldsymbol{x}_k{}^{(i)}, W_k{}^{(i)}) \in \mathcal{X}_k$, where $i \in \{1, ..., N\}$ denotes the index of the ith particle, N is the total number of particles and the (normalized)

[46] See furthermore [CD02] for a survey on convergence results.

weight is defined as ${W_k}^{(i)} \geq 0$ with $\sum_{i=1}^{N} {W_k}^{(i)} = 1$. The posterior density $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ can then be represented using the discrete distribution of the particles:

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \approx \sum_{i=1}^{N} {W_k}^{(i)} \delta(\boldsymbol{x}_k - {\boldsymbol{x}_k}^{(i)}), \tag{3.38}$$

where $\delta(\,\bullet\,)$ denotes the Dirac delta function.[47]

Based on the distribution of the particles, the expectation of a (nonlinear) function $h(\boldsymbol{x})$ (3.21) can be approximated by a weighted average[48] of the function values evaluated for each sample [CGM07, p.903]:

$$E[h(\boldsymbol{X})] \approx \sum_{i=1}^{N} {W_k}^{(i)} h({\boldsymbol{x}_k}^{(i)}). \tag{3.39}$$

If perfect Monte Carlo sampling would be used, that is, random samples are independent and identically distributed (i.i.d.) and directly drawn from $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$, each of the weights would be equal (${W_k}^{(i)} = 1/N$) and by increasing the number of samples, the result of the approximation (3.39) would converge (almost sure) to the exact expectation (3.21), which follows from the *strong law of large numbers* [DFG01, p.7].

However, it is usually not possible to sample directly from complicated probability density functions, which is typically the case for the posterior density [DJ11, p.8; CGM07, p.902]. This leads to the methods based on *importance sampling* that are used to compute the weights ${W_k}^{(i)}$. Instead of sampling directly from the *target density*, an *importance density*[49] is used, which can be selected such that samples can easily be drawn from it. A prerequisite is that the support of the importance density function contains the support of the target density function, meaning that if $p(\boldsymbol{x}) > 0$ then also $q(\boldsymbol{x}) > 0$.

The expectation of a function $h(\boldsymbol{x}_{0:k})$ can then be computed using the importance density

[47] Compare the notation in, e.g., [DFG01, p.8f, (1.5); DJ11, p.8; Cha+11, p.43, 46f; HSG06, p.1; ETM21, p.11].

[48] In general, the computation of the expected value is interpretable as the minimization of a cost function using the squared 2-norm in $\mathbb{R}^n$ [Chi09, p.66]. However, expected values of elements of non-Euclidean spaces (like $SO(3)$ or SE) need to be handled in a special way. The *Riemannian center of mass* [GKR74; Kar77] (or *Karcher mean* as it was called by [Ken90, p.395]) minimizes the quadratic distance defined on any Riemannian manifold. The Riemannian center of mass for averaging of rotations is investigated by Moakher [Moa02] (Section 5 for weighted averages). An algorithm is implemented by [Man04] for (compact) Lie groups. Furthermore, the method by [Mar+07] based on quaternions is also suitable for particle filtering. A general review of methods for averaging rotations is provided by [Har+13].

[49] The importance density is called differently in various combinations of the terms {importance, instrumental, proposal, propagation} and {density, distribution}, e.g., [DJ11, p.9; CGM07, p.902].

$q(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$. From (3.21) follows [CGM07, p.902f, (3), (8)]:

$$\begin{aligned} E[h(\boldsymbol{X})] &= \int h(\boldsymbol{x}_{0:k})\; p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})\; \frac{q(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})}{q(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})} d\boldsymbol{x}_{0:k} \\ &= \int h(\boldsymbol{x}_{0:k})\; r(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) q(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k}) d\boldsymbol{x}_{0:k} \\ &\approx \sum_{i=1}^{N} \frac{\tilde{W}_k{}^{(i)}}{\sum_{j=1}^{N} \tilde{W}_k{}^{(j)}} h(\boldsymbol{x}_{0:k}{}^{(i)}) = \sum_{i=1}^{N} W_k{}^{(i)} h(\boldsymbol{x}_{0:k}{}^{(i)}), \end{aligned} \tag{3.40}$$

where samples $\boldsymbol{x}_{0:k}{}^{(i)}$ were now drawn from the importance density $q(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$, and the *importance function* $r(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k})$ corrects the bias that was introduced by not directly sampling from the posterior.[50]

The value of the importance function evaluated for a sample is the so-called *unnormalized* importance weight defined as [CGM07, p.902ff]:

$$\tilde{W}_k{}^{(i)} := r(\boldsymbol{x}_{0:k}{}^{(i)}|\boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{x}_{0:k}{}^{(i)}|\boldsymbol{y}_{1:k})}{q(\boldsymbol{x}_{0:k}{}^{(i)}|\boldsymbol{y}_{1:k})}. \tag{3.41}$$

Normalized importance weights can always be obtained using

$$W_k{}^{(i)} = \tilde{W}_k{}^{(i)} / \sum_{j=1}^{N} \tilde{W}_k{}^{(j)}. \tag{3.42}$$

By decomposing the importance and the posterior density in (3.41) into factors, a recursive computation of the importance weights can be derived:[51]

$$\tilde{W}_k{}^{(i)} \propto W_{k-1}{}^{(i)}\; \frac{g(\boldsymbol{y}_k|\boldsymbol{x}_k{}^{(i)})\; f(\boldsymbol{x}_k{}^{(i)}|\boldsymbol{x}_{k-1}{}^{(i)})}{q(\boldsymbol{x}_k{}^{(i)}|\boldsymbol{x}_{0:k-1}{}^{(i)}, \boldsymbol{y}_{1:k})}, \tag{3.43}$$

where $W_{k-1}{}^{(i)}$ is the importance weight from the previous timestep, and the so-called *incremental weight* on the right consists of the likelihood, the transition density and the importance density.[52]

If the so-called *prior kernel* is used for the importance density (as done by [GSS93]), where the importance density equals the transition density of the system model, then (3.43) simplifies to [CGM07, p.905]:

$$\tilde{W}_k{}^{(i)} \propto W_{k-1}{}^{(i)}\; g(\boldsymbol{y}_k|\boldsymbol{x}_k{}^{(i)}). \tag{3.44}$$

The filtering method that draws samples from $q(\boldsymbol{x}_k{}^{(i)}|\boldsymbol{x}_{0:k-1}{}^{(i)})$ and weights them according to the above equations is called sequential importance sampling (SIS).

[50] Note that in (3.40) a renormalization of the importance weights is carried out [CGM07, Remark 1, p.902].
[51] Compare, e.g., [ETM21, p.8, (13); DFG01, p.9, (1.6); CGM07, p.903f, (10)], and Ho and Lee [HL64, p.335f].
[52] An additional constant term in the denominator was omitted as the weights are in any case unnormalized.

The downside of SIS is that the weights degenerate quickly after a few time steps due to the multiplication of the previous weight with the incremental weight (compare, e.g., [DFG01, p.10]). *Resampling* can help to remedy this effect.

The sampling and importance resampling (SIR) algorithm of [Rub88] draws samples from an importance density (sampling), weights them (importance weighting), and then resamples from this set the ones with high weights (resampling) by setting the resampling probability equal to the importance weight. The classical SIR algorithm aims to concentrate the samples in regions with relevant data, but is not meant to be applied iteratively.

A resampling step was then also suggested by Gordon et al. [GSS93] in the (sequential) bootstrap filter, which reduced the degeneracy effect SIS suffered from. This finally led to successful applications of the sequential Monte Carlo methods in practice. At each time step, N particles are selected from the original population for the next iteration. The probability of a particle being selected is given by its importance weigth: $Pr[\boldsymbol{x}_k{}^{(i)} = \boldsymbol{x}_k^{\star(j)}] = W_k{}^{(i)}$ [GSS93, p.108], where $\boldsymbol{x}_k^{\star(j)}$ is a copy of $\boldsymbol{x}_k{}^{(i)}$ and is passed to the next filter iteration. After the resampling step, all selected particles obtain the same weight $W_k{}^{(i)} = 1/N$ and the posterior is represented by

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \approx \frac{1}{N}\sum_{i=1}^{N} N_k{}^{(i)}\delta(\boldsymbol{x}_k - \boldsymbol{x}_k{}^{(i)}) = \frac{1}{N}\sum_{j=1}^{N}\delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{\star(j)}). \tag{3.45}$$

Here, $N_k{}^{(i)} \in \mathbb{N}_0$ indicates how often a particle $\boldsymbol{x}_k{}^{(i)}$ was selected and copied for the next iteration, compare [DFG01, p.10; HSG06, p.1].

The bootstrap filter by Gordon et al. [GSS93] is the basis of the common particle filtering methods and also of the implementation in this work. Alg. 1 shows the main function calls of a general estimation step (DoEstimationStep), which is executed sequentially after initialization of the particles in the implementation of the state observer used for contact sensing.

```
1: function DoEstimationStep(X_{k-1}, y_k )
2:     for all (W_{k-1}^(i), x_{k-1}^(i)) ∈ X_{k-1} do
3:         x_k^(i) ← PropagateSample(x_{k-1}^(i), y_k)
4:         W̃_k^(i) ← WeightSample(x_k^(i), y_k)
5:     X̃_k ← {(W̃_k^(i), x_k^(i))}
6:     X_k ← Normalize(X̃_k)
7:     X_k^⋆ ← Resample(X_k)
8:     return X_k^⋆
```

Algorithm 1: The general estimation step of the state observer.

First, each sample is propagated with PROPAGATESAMPLE using the importance density function. Then, the samples are weighted according to the newly available measurement $\boldsymbol{y}_k$ in WEIGHTSAMPLE and normalized in NORMALIZE. Finally, the resampling step selects the samples for the next call of the estimation step (RESAMPLE).

In this work, *systematic resampling* [HSG06] is used for selecting the particles for the next iteration. Note that resampling is carried out here at every timestep, but it could be also omitted in cases where the effective sample size (ESS) is above a certain threshold [DJ11, p. 14]. The overall algorithm is implement in a custom C extension for Python with parallel execution of the propagation and weighting step.

Functions for the propagation (PROPAGATESAMPLE) and weighting (WEIGHTSAMPLE) of the samples in $\mathcal{X}_k$ are developed and investigated in this work; see Section 3.5 for more details. The integration of this estimation step in algorithms for adaptive motion generation is presented in Section 3.6.

3.5 Uncertainty and Contact Sensing

Contact sensing is considered a prerequisite for adaptive execution of assembly tasks, as it provides necessary awareness for decisions about the strategy and, in particular, about the next motion commands. In this work, it is mainly investigated how geometric uncertainties can be reduced and observed with the help of contact sensing during the execution of assembly tasks. This section provides the description of the uncertainty model and the developed system models within the estimation framework.

First, the model of the uncertainty is described in Section 3.5.1. Then, the propagation models for static setups (Section 3.5.2) and for setups with moving parts in the environment (Section 3.5.3) are presented. Finally, Section 3.5.4 describes how the sensory input is taken into account in the likelihood function of the estimator.

3.5.1 Uncertainty Model

Geometric uncertainty can arise from various sources. Considered in robotic assembly are typically: manufacturing tolerances, the imprecision in the pose of static objects in the environment, imprecision in the pose of the gripper, and in the pose of the manipulated object in the gripper [Dut98, p.43; RBS05, p. 568f]. Furthermore, Rosell et al. [RBS05, p.568] distinguish between *global* and *local* sources of uncertainty: ”A source of uncertainty is considered global if it determines the deviation of either all the topological elements of the manipulated object or of all the topological elements of all the static objects, and it is considered local otherwise.” In this work, mainly a global source of uncertainty is considered.

Besides the sources of uncertainty listed above, the pose of an object in the environment can have additional uncertainty as soon as it is not fixed in a static pose, but is free to move. The motion might come from an external actuation (e.g., conveyor belt), a human co-worker, or might be induced by the robotic arm itself (pushing manipulation). In this work, mainly the uncertainty in the pose of an object in the environment is treated. Cases with static and with moving parts are considered.

For robotic assembly, an important question is how the uncertainty affects the success and performance of an assembly process. Taylor [Tay76, Ch. 7; TR88] and Brooks [Bro82, Sec. 2] investigated early the propagation of uncertainty through kinematic structures using (in-equality) constraints. Their methods provide worst-case bounds on the size of the uncertainty to find a feasible system setup such that a given task can be executed. Later, Dutré [Dut98, p.55ff] models (dependent) kinematic uncertainties of various types considering a manipulator, which is in contact with the environment, and using VCMs by Bruyninckx et al. [Bru+95]. Finally, De Schutter et al. [De +07] propose a general formalism to specify uncertainties in kinematic structures consisting of robots and features of objects.

Although the actual uncertainty may have different sources in the kinematic structure, the errors eventually add up to an overall uncertainty in the relative pose between the parts to be mated in the assembly process. Therefore, and under the assumption that uncertainties along the kinematic chain are considerably small, a model is used in this work in which all uncertainty is located in the pose of the object in the environment.

Accordingly, it is assumed that the grasp transformation between end-effector with frame A and manipulated object with frame D is known and constant: $\boldsymbol{H}_{AD} = \text{const.} \in SE(3)$, and furthermore, that the pose of the frame A with respect to the robot base B can be accurately computed from the measured joint positions $\boldsymbol{q}_k$ at time $t = t_k$ using forward kinematics: $\boldsymbol{H}_{BA,k} = \boldsymbol{H}_{BA}(\boldsymbol{q}_k)$, see (3.10). The actual pose of the object in the environment with frame C relative to the robot base B is subject to uncertainty and given by:

$$\boldsymbol{H}_{BC,k} = \boldsymbol{H}_{B\bar{C}}\boldsymbol{H}_{\bar{C}C}(\boldsymbol{x}_k), \tag{3.46}$$

where the uncertainty is located in the transformation $\boldsymbol{H}_{\bar{C}C}(\boldsymbol{x}_k)$ represented by the state $\boldsymbol{x}_k$ (see Fig. 13). The frame $\bar{C}$ is assigned to the nominal or initially assumed pose of the object.[53]

The initial pose is considered available from prior knowledge about the workcell setup or from a preliminary localization process. The considered uncertainty at the beginning of the assembly process is in a small range of a few centimeters and with small errors in the orientation. In this case, no global search phase is required beforehand. However, the

[53] This corresponds to the formulation of Taylor in which the actual transformation is composed of a nominal transformation and a perturbation [Tay76, p.142; TR88, p.231].

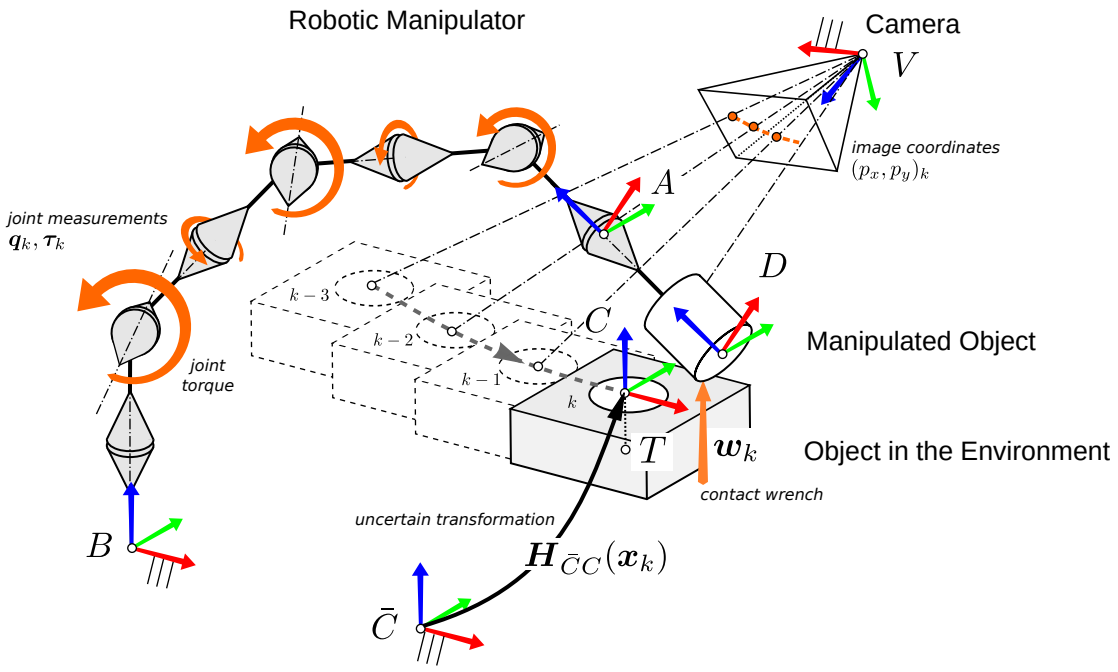


Figure 13: Frames of the robotic manipulator, the manipulated object and the object in the environment, and the camera, as well as available measurements and the considered uncertain transformation. Source: Figure taken and modified from **Publication 4** (Fig. 4)

combination of contact sensing with visual sensory inputs as carried out in [Sac+19] and **Publication 4** allows for an increase in the size of the uncertainty.

The state variables (or error variables) $\boldsymbol{x}_k = (x_1, ..., x_U)$ encode the uncertainty in the position and orientation. As long as not stated differently, it will be assumed that the state encodes full 3D-uncertainty in the position and orientation with:

$$\boldsymbol{x} = (x, y, z, \alpha, \beta, \gamma) \in \mathbb{R}^6, \tag{3.47}$$

where α, β and γ are Z-Y-X Euler angles [Cra09, p.43].

By closing the kinematic chain, the actual relative pose between the parts can be expressed in dependency of the state and the joint position measurement with the transformation:

$$\boldsymbol{H}_k := \boldsymbol{H}_{CD}(\boldsymbol{q}_k, \boldsymbol{x}_k) = \boldsymbol{H}_{C\bar{C}}(\boldsymbol{x}_k)\boldsymbol{H}_{\bar{C}B}\boldsymbol{H}_{BA}(\boldsymbol{q}_k)\boldsymbol{H}_{AD}, \tag{3.48}$$

which can be used to compute the contact distance $\tilde{d}_k$ and the wrench $\tilde{\boldsymbol{w}}_k$ from the virtual contact model as described in Section 3.3.3.

Note that the investigated methods would in principle allow to handle uncertainties at different locations in the kinematic structure, for example, in the grasp transformation or in the dimensions of the elements of the robot. On the side of the robotic arm, it would then additionally require to model the influence of the uncertainty on the manipulator dynamics.

However, higher dimensional spaces will also be more challenging to treat in the estimation framework.[54] It is therefore recommended to identify kinematic and dynamic properties related to the robotic arm itself beforehand through calibration routines, and furthermore, ensure stable grasps before the actual assembly process starts.

3.5.2 Propagation Models for Static Setups

Although the considered uncertainty is small, the distribution can be strongly nonlinear and multi-modal due to the possible contact situations, which can vary strongly under small perturbations of the poses of the objects. Therefore, the SMC framework is used in this work to develop observation methods for the uncertainty in the kinematic chain. Accordingly, the uncertainty is represented by a set of N samples ${\boldsymbol{x}_k}^{(i)}$, which will be propagated and updated in the sequential estimation algorithm (see Section 3.4.3).

As a starting point, a static setup is considered in which the components and parts are at fixed locations in the robotic workcell. The main uncertainty arises from inaccurately known poses of part feeders and fixtures, which might not be calibrated in an exact manner to save efforts during setup in a flexible production, for example. In this case, as no further information is available, the initial uncertainty is assumed to be uniform around the nominal location and the initial set of particles $\boldsymbol{\mathcal{X}}_0$ is initialized by drawing samples from an uniform distribution.

Furthermore, as the parts in the setup are static, estimating the uncertainty may be seen as a parameter estimation process. For this purpose, a common practice is to add *artificial noise* in the propagation step to circumvent the degeneracy of the sample population [CGM07, p.918], also known as *sample impoverishment* during the identification of supposedly constant parameters.[55] Accordingly, the system dynamics are described with a random motion model of the form:

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \boldsymbol{u}_{\boldsymbol{x}}, \tag{3.49}$$

where $\boldsymbol{u}_{\boldsymbol{x}}$ is the artificial noise with covariance matrix $\Sigma_{\boldsymbol{x},p}$.

The propagation model can be implemented by a function PUREDIFFUSION that draws a sample from a multivariate Gaussian (Alg. 2).

1: **function** PUREDIFFUSION(${\boldsymbol{x}_{k-1}}^{(i)}$)
2: draw ${\boldsymbol{x}_k}^{(i)} \sim \mathcal{N}({\boldsymbol{x}_{k-1}}^{(i)}, \Sigma_{\boldsymbol{x},p})$.
3: **return** ${\boldsymbol{x}_k}^{(i)}$

Algorithm 2: Propagation function representing a pure diffusion of the samples.

[54] The treatment of higher dimensional spaces is discussed briefly in Section 5.2.

[55] Compare the *roughening* procedure by Gordon et al. [GSS93, p.112], and the *self-organizing state-space model* by Kitagawa [Kit98] using state augmentation for parameter identification; further examples of artificial noise in [CB03] and [TMO10] with similar application scenarios as in this work.

Without update and resampling, the samples would randomly diffuse in all directions governed by the specified covariance and ignoring the constraints from the geometries of the objects. The necessary consistency with the configuration space will be established after the propagation in the update step using a likelihood function on the poses of the parts (see Section 3.5.4).

Note that the small variations induced by the artificial noise will furthermore be necessary to establish the consistency between the measured contact wrench and the virtual contact model through small changes of the relative pose within the contact zone between the objects.

The function PUREDIFFUSION is used as the default propagation model in the estimation framework if no additional prior knowledge about the task is available. However, especially in scenarios with tight constraints, it requires a high number of samples to increase the density in the relevant regions. It is difficult to place and keep samples in so-called *narrow passages* of the configuration space that enclose a small volume compared to the free space.

In the field of probabilistic roadmap planning (PRM) [Kav+96] techniques where developed to increase the sample density in such regions. As described in **Publication 2**, two heuristic sampling methods were integrated in the propagation step to improve the sampling performance for assembly tasks (see Alg. 3 and Fig. 14).

```
1: function GAUSSIANSAMPLER(x_{k-1}^(i), q_k)
2:     x_I := x_{k-1}^(i)
3:     C_I ← EVALCONTACT(x_I, q_k)
4:     if C_I ≠ contact then
5:         for j = 1 to L_max do
6:             draw x_II ~ N(x_I, Σ_{x,g})
7:             C_II ← EVALCONTACT(x_II, q_k)
8:             if C_II = contact then
9:                 return x_II
10:    return PUREDIFFUSION(x_I)
```

```
1: function BRIDGETEST(x_{k-1}^(i), q_k)
2:     x_I := x_{k-1}^(i)
3:     for j = 1 to L_max do
4:         draw x_II ~ N(x_I, Σ_{x,b})
5:         C_II ← EVALCONTACT(x_II, q_k)
6:         if C_II = invalid then
7:             x_III ← AVERAGE(x_I, x_II)
8:             C_III ← EVALCONTACT(x_III, q_k)
9:             if C_III ≠ invalid then
10:                return x_III
11:    return PUREDIFFUSION(x_I)
```

Algorithm 3: The propagation functions of the Gaussian and the Bridge Test sampler.

The propagation function GAUSSIANSAMPLER is inspired by the PRM method of [BOS99]. It collects samples that represent configurations at the border of the configuration space inside of the contact zone $\partial\tilde{\mathcal{C}}$, and thus increases the density in the regions where potential contacts can occur and the first feedback from tactile sensing will be received. The BRIDGETEST function adopted from [Sun+05] pulls the samples into the narrow passage and increases the performance especially for insertion tasks. Both functions draw auxiliary samples in a loop and evaluate their contact situation according to (3.19) with EVALCONTACT. The number of iterations L_{max} controls the search effort and finally how dense the regions will be filled with samples.

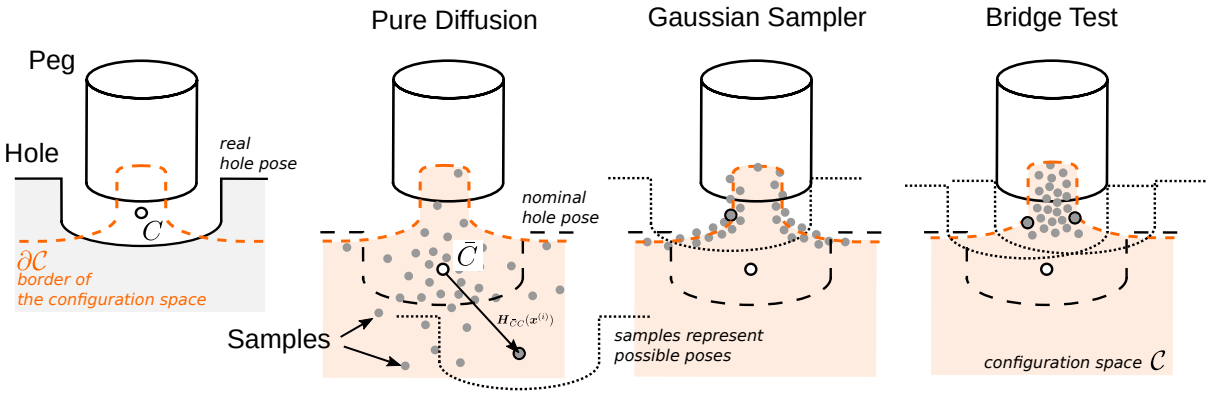


Figure 14: Conceptual visualization of the propagation models for static setups. The Pure Diffusion method spreads samples around the nominal pose of the hole comparable to a diffusion process, the Gaussian sampler collects samples at the border of the configuration space, the Bridge Test method favors samples in so-called narrow passages of the configuration space.

As shown in **Publication 2**, the application of these propagation functions produces reliable estimates in assembly tasks with fewer particles, and the risk of sample impoverishment can be reduced.

3.5.3 Propagation Models for Moving Parts in the Environment

The setup in which assembly tasks are carried out is not always static. Typically, products are carried by logistic systems (e.g., on mobile transport systems and conveyor belts) for an efficient production flow, especially in large facilities. The conventional approach is to directly measure the position and velocity of the logistic system and communicate it to the robot.[56] However, although these velocities are ideally constant and continuous there is uncertainty due to fluctuations and unplanned stops, or the measurements cannot be forwarded due to the lack of an appropriate communication infrastructure.

Another aspect is relevant in scenarios with small lot sizes. In order to stay flexible with respect to product variants, there might be fewer specialized fixtures in the workcell, which adds additional uncertainty to the poses. With fewer constraints, parts might be pushed and moved during the assembly process. However, if the robot is capable of executing assembly tasks in such an unstable environment this would clearly increase the flexibility of the overall system.

In either case, it is advantageous if the robot can adapt autonomously to moving objects without depending on further external measurement devices to reduce the overall setup complexity. Therefore, it is desirable that additional intrinsic sensory input supports the localization and tracking of parts.

[56] See related products of robot suppliers for motion tracking, e.g., [Yas17; KUK20].

Existing approaches dealing with moving objects using sensor guided motions in assembly are, for example, the insertion of a piston using visual servoing in combination with a force controller [Jör+00] or the assembly of car wheels on conveyor systems [Che+09; LSH10]. In this work, methods are investigated that support the observation of moving parts using the Bayesian estimation framework as will be described in the following.

The first considered case is a moving object actuated by an external motion source (e.g., a conveyor belt). A common approach for a propagation model from the field of general object tracking is a constant velocity model (CV) model [Cha+11, p. 57ff, p.64]. In this work, it is assumed that the object in the environment is moving with an unknown translational motion in an arbitrary direction. Accordingly, the state is augmented with velocity variables for the position components of the object pose: $\boldsymbol{x} = (x, \dot{x}, y, \dot{y}, z, \dot{z})$. Then, the model assumes that the object moves with constant velocity and the artificial noise $\boldsymbol{u}_k$ corresponds to changes of the motion:[57]

$$\boldsymbol{x}_k = \left(\boldsymbol{I}_3 \otimes \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} \right) \boldsymbol{x}_{k-1} + \boldsymbol{u}_k, \tag{3.50}$$

with the 3×3 identity matrix $\boldsymbol{I}_3$ and $\otimes$ denotes the Kronecker product.

This propagation model is investigated in **Publication 4** for peg-in-hole assembly, in which the (unknown) motion of the hole is simulated with a second robotic arm and intrinsic tactile sensing is used for tracking the object pose during the insertion phase when the vision system is occluded (see Fig. 7). It is shown that using object tracking methods in a Bayesian framework allows implementing adaptive assembly skills that have motion commands specified relative to feature frames of moving objects (see also Section 3.6.1).

The implementation of the CV model is summarized in Alg. 4 and visualized in Fig. 15.

1: **function** CONSTANTVELOCITY($\boldsymbol{x}_{k-1}{}^{(i)}$)
2: $\quad \bar{\boldsymbol{x}}_k \leftarrow$ INTEGRATEVELOCITY($\boldsymbol{x}_{k-1}{}^{(i)}, \Delta T$)
3: $\quad$ draw $\boldsymbol{x}_k{}^{(i)} \sim \mathcal{N}(\bar{\boldsymbol{x}}_k, \Sigma_{\boldsymbol{x}})$.
4: $\quad$ **return** $\boldsymbol{x}_k{}^{(i)}$

Algorithm 4: Propagation of the samples using a constant velocity motion model.

The CV model as presented does not incorporate the current velocity of the robotic arm in the propagation of the samples and does not directly model the effect of the contact on the motion of the object. In general, the reaction of the object in the environment after contact with a robotic arm could be described physically with the equations of motion. However, those are typically nonlinear differential equations, which require specific solution strategies. In the context of particle filtering, it was investigated to use linear models for known contact

[57] Various motion models exist in the field of tracking [LJ03]; commonly used is a random acceleration process noise [Sah18, p.237]. In this work, a random velocity change is assumed, which is applied at the last instance of each time period, compare [Li14, p.4] and see **Publication 4** [NSA21, p.12].

situations in a hybrid filtering approach [LLT15; LLT18], or to integrate a physics engine for simulating the state dynamics [Duf+11; ZT12; Wir+19]. However, for complex geometries and contacts with tight clearances it is still computationally very expensive to reach an update rate in simulation that is sufficient for the online integration of the simulation into the estimation framework and guaranteeing real time execution. Note that this simulation would need to be called not only once, but multiple times at each estimation step.

A cheap, naive approach without simulation or knowledge of the contact situation would be to propagate the samples with the same velocity as the robot by assuming a fixed attachment of the object to the end effector. Unfortunately, ignoring the possible relative motions in contact leads to estimation errors in the direction of unobservable relative motions.

As a compromise between full physics simulation and estimation accuracy, a method is developed in **Publication 3** that pushes the samples into the direction of motion of the robotic arm, but only if this motion would violate the contact constraints. A violation is present when negative virtual energy $\Delta\tilde{E}^{(i)} \approx {}_D\boldsymbol{t}_k{}^T {}_D\tilde{\boldsymbol{w}}_{k-1}{}^{(i)}\Delta T < 0$ is generated in the virtual contact model. This can be addressed in the propagation step by applying a compensation motion ${}_D\boldsymbol{t}^{(i)}$ for each particle computed from the the currently measured twist ${}_D\boldsymbol{t}_k$ of the robot motion and the virtual contact wrench ${}_D\tilde{\boldsymbol{w}}_{k-1}{}^{(i)}$ with the relation:

$$ {}_D\boldsymbol{t}^{(i)} = \frac{\Delta\tilde{E}^{(i)}}{(\tilde{\boldsymbol{C}}^{(i)}{}_D\boldsymbol{t}_k)^T {}_D\tilde{\boldsymbol{w}}_{k-1}{}^{(i)}\Delta T}\tilde{\boldsymbol{C}}^{(i)}{}_D\boldsymbol{t}_k. \tag{3.51} $$

The matrix $\tilde{\boldsymbol{C}}^{(i)} \in \mathbb{R}^{6\times6}$ is a twist *filter* [DMB93, p.13f] that selects the motion directions that should be used to neutralize the superfluous virtual energy. By setting $\tilde{\boldsymbol{C}}^{(i)} = \boldsymbol{I} - \boldsymbol{T}^{(i)}\boldsymbol{T}^{(i),\#}$ with the twist space of the contact $\boldsymbol{T}^{(i)} = [{}_D\boldsymbol{t}_1, \ldots, {}_D\boldsymbol{t}_{N_C}], N_C \leq 5$ (see Section 3.3.2) the compensation motion of the object corresponds to the motion of the robot that violates the contact.[58] A proposal for the state transition can then be computed for each particle from the twist ${}_D\boldsymbol{t}^{(i)}$ with the exponential map (see Section 3.2.1).

By using the knowledge from the local geometry to construct the twist space and the measurement of the current robot motion, this constraint-based propagation model improves the state estimation for assembly cases in which the robot pushes the parts in the environment. Therefore, there is now a second propagation function (CONSTRAINTBASED) available besides the CV model to handle moving objects (Alg. 5).

In summary, multiple methods for propagating samples were investigated in this work. Table 3.1 provides an overview of the presented functions that can be used as implementation for the propagation step (PROPAGATESAMPLE) in the estimator (Section 3.4.3).

[58] The weighted generalized inverse is computed with $\boldsymbol{T}^{(i),\#} = (\boldsymbol{T}^T\boldsymbol{M}_v\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{M}_v$ using the *kinetic energy metric* [DMB93, (38) and (83)] and selecting the mass matrix of the moving object for M_v.

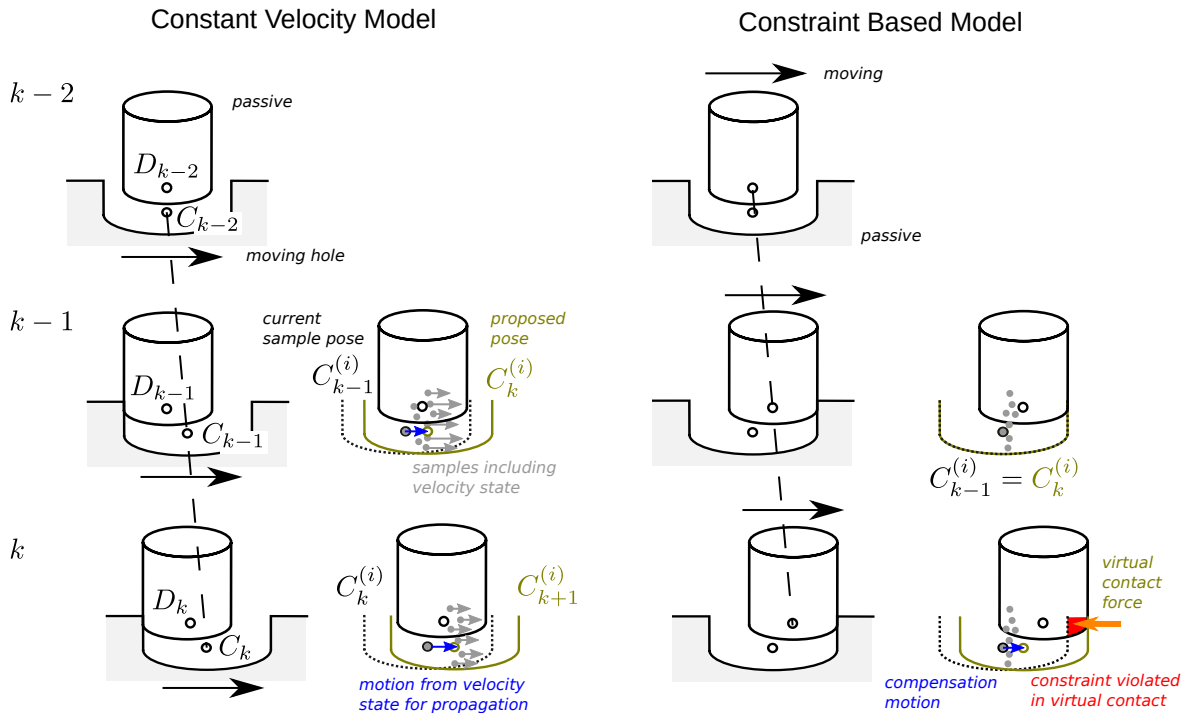


Figure 15: The constant velocity model (left) contains the velocity of the moving hole in the state and can use it for the propagation function. Each sample is propagated according to its assumed velocity. The constrained-based model (right) propagates samples via a compensation motion as soon as a constraint is violated in the virtual contact model.

Table 3.1: Propagation models investigated for contact sensing in various situations

PUREDIFFUSION	default model	⊕ fast implementation ⊖ high number of particles required
GAUSSIANSAMPLER	contact-rich tasks	⊕ more particles in contact relevant regions ⊕ reduced risk of particle impoverishment ⊖ additional contact model queries
BRIDGETEST	insertion tasks	⊕ more particles in narrow passages ⊕ reduced risk of particle impoverishment ⊖ additional contact model queries
CONSTANTVELOCITY	moving objects	⊕ fast implementation
CONSTRAINTBASED	push motions	⊕ efficient incorporation of current robot motion ⊖ requires model of twist space in contact

```
1: function ConstraintBased(x_{k-1}^{(i)}, q_k, q̇_k, _Dw̃_{k-1})
2:     _Dt_k ← ComputeCurrentTwist(q_k, q̇_k)
3:     ΔẼ^{(i)} ← ComputeVirtualEnergy(_Dt_k, _Dw̃_{k-1})
4:     if ΔẼ^{(i)} < 0 then
5:         T^{(i)} ← GetCurrentTwistspace(x_{k-1}^{(i)}, q_k)
6:         x_k^{(i)} ← ComputeCompensationMotion(ΔẼ^{(i)}, T^{(i)}, _Dt_k, _Dw̃_{k-1}, x_{k-1}^{(i)}, q_k)
7:     else
8:         x_k^{(i)} ← x_{k-1}^{(i)}
9:     return x_k^{(i)}
```

Algorithm 5: Constraint-based propagation function.

3.5.4 Intrinsic Tactile Likelihood

After the propagation of the samples, the next step in the estimation framework is the update of the (hidden) state distribution by evaluating how well the current measurement $\boldsymbol{y}_k$ can be explained by the state samples. For this purpose, the likelihood $g(\boldsymbol{y}_k|\boldsymbol{x}_k{}^{(i)})$ is computed for each sample (compare general algorithm in Section 3.4.3).

In tactile sensing, general properties of contacts can be used for setting up a likelihood model. For rigid bodies with frictionless contacts, it is a well-known condition that no power is generated through the contact forces during the constraint motion, which can be derived from the general *reciprocity* condition [Bal98, p.26] for bodies in equilibrium (see 3.2.1). A second property is the *consistency* condition, which states that the actual velocity and the acting forces in contact must match the constraints defined by the local geometry, which means that they are composed of a linear combination of the basis vectors of the twist and wrench space, respectively.

Reciprocity and consistency conditions were used for setting up equations for estimating geometric uncertainties and formulating likelihood models [Bru+95; Mee+07]. However, applying the consistency conditions in the update step has advantages if various different contact formations (CFs, see Section 3.3.1) may occur as Meeussen et al. [Mee+07, p.229] conclude: “Therefore, while the reciprocity model is only appropriate to distinguish between two models in the same CF, the consistency model can also be used to distinguish between different CFs.” In this work, the consistency of the measured external wrench with the contact wrench from the virtual contact model will be considered for the likelihood computation as described in the following.

The considered robot types provide joint position and joint torque measurements as input for the computation of the likelihoods.[59] The measured joint torque contains the effects of the gravitation, the robot dynamics and the influence of external contacts. In this work, only the

[59]The fusion with measurements of visual features is described in [Sac+19] and used in **Publication 4** [NSA21], but is omitted here to keep a brief description.

external torque induced by the contact with the environment is of interest. The external joint torque can be estimated from the joint torque sensor measurements taking into account the joint positions and their derivatives [Had+08, p. 3357]. In the implementation of this work, an estimate of the external joint torque $\boldsymbol{\tau}_k$ is provided by the controller of the robot.[60]

The external joint torque is the first ingredient for the likelihood needed for the update step of the filter, see (3.44). It is compared with the hypothetical joint torque that is computed with the virtual contact model (see Section 3.3.3) for a particular sample value ${\boldsymbol{x}_k}^{(i)}$ and the current joint position $\boldsymbol{q}_k$. For this purpose, the virtual contact wrench ${}_D\tilde{\boldsymbol{w}}_k{}^{(i)} = {}_D\tilde{\boldsymbol{w}}({\boldsymbol{H}_{CD}({\boldsymbol{x}_k}^{(i)}, \boldsymbol{q}_k)})$ is mapped into the joint space using the Jacobian $\boldsymbol{J}_k = \boldsymbol{J}_A(\boldsymbol{q}_k)$ according to:

$$\tilde{\boldsymbol{\tau}}_k{}^{(i)} = \boldsymbol{J}_{k\,A}^T \tilde{\boldsymbol{w}}_k{}^{(i)} = \boldsymbol{J}_k^T \boldsymbol{\Delta}\, \mathrm{Ad}(\boldsymbol{H}_{AD}) \boldsymbol{\Delta}_D \tilde{\boldsymbol{w}}_k{}^{(i)}, \tag{3.52}$$

where $\boldsymbol{\Delta}$ is the interchange matrix and $\mathrm{Ad}(\boldsymbol{H}_{AD})$ is the adjoint of the transformation $\boldsymbol{H}_{AD} = \mathrm{const.}$ between the object and the last frame of the manipulator (see Section 3.2.2).

The error in the joint torque measurements is assumed to be normally distributed, and thus the likelihood can be calculated by evaluating the Gaussian probability density function with mean $\tilde{\boldsymbol{\tau}}_k{}^{(i)}$ and covariance $\Sigma_{\boldsymbol{\tau}}$ (compare **Publication 2**):

$$s_F(\boldsymbol{\tau}_k | {\boldsymbol{x}_k}^{(i)}) = \mathcal{N}(\boldsymbol{\tau}_k | \tilde{\boldsymbol{\tau}}_k{}^{(i)}, \Sigma_{\boldsymbol{\tau}}). \tag{3.53}$$

Note that the likelihood depends on the kinematics and the current configuration of the robotic arm. However, the computation of the likelihood in the joint space has the advantage that there is a uniform metric space and no artificial metric in the wrench space needs to be defined to compare the effect of the contacts.

As a second ingredient for the likelihood, the pose measurements are incorporated by evaluating the contact distance. This ensures consistency with the relative configuration space of the parts. Note that the contact distance cannot be measured directly with a sensor. Instead, only the position measurement from the robot joints is available. Together with a hypothetical pose of a sample, a virtual contact distance $\tilde{d}_k$ can be computed from the geometrical model (see Section 3.3.3). Accordingly, the purpose of this likelihood is to keep the sample distribution consistent with the measured pose of the robotic arm (and not to incorporate a direct distance measurement of an actual real sensor).

The likelihood is computed using the virtual contact distance and the following observation

[60] The controller of the KUKA LBR iiwa provides such an estimate through the RoboticsAPI. Additionally, a *driving torque* filter [DMB93, p.14] is applied to process only the torque induced by the external wrench acting at the end-effector.

density as introduced in **Publication 2**:

$$s_d(\boldsymbol{q}_k|\boldsymbol{x}_k{}^{(i)}) = (\sigma_d\sqrt{2\pi})^{-1} \cdot \begin{cases} 1, \ \boldsymbol{H}_k{}^{(i)} \in \tilde{\mathcal{C}} \cup \partial\tilde{\mathcal{C}} \\ \exp(-\frac{(\tilde{d}_k - d_t)^2}{2\sigma_d^2}), \ \boldsymbol{H}_k{}^{(i)} \notin \tilde{\mathcal{C}}. \end{cases} \tag{3.54}$$

Samples representing invalid contacts (with penetrations exceeding the boundaries of the configuration space) are punished immediately by a reduction of their likelihood. The parameter σ_d is a parameter to smooth the transition between valid and invalid contacts. In this way, particles that are very close to the configuration space boundary do not immediately vanish, and sample impoverishment can be reduced.[61] Consequently, the sample distribution also approximates locally the configuration space between the parts.

Note that the joint position measurement errors are not modeled here and are furthermore considered to be neglectable. The likelihood therefore can be interpreted as a constraint in the state space model integrated in the update step of the estimator.[62]

Furthermore, due to the feasible contact zone defined by d_t in the virtual model, compare (3.19), the likelihood on the distance does not interfere with the likelihood on the contact force in the relevant space (i.e., in $\partial\tilde{\mathcal{C}}$ and $\tilde{\mathcal{C}}$). Under the assumption of independence, the total likelihood for contact sensing is then obtained by the multiplication of the likelihood on the external contact and the likelihood on the configuration space consistency:

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k{}^{(i)}) = s_d(\boldsymbol{q}_k|\boldsymbol{x}_k{}^{(i)}) \ \cdot \ s_F(\boldsymbol{\tau}_k|\boldsymbol{x}_k{}^{(i)}). \tag{3.55}$$

This is the basic likelihood function suggested and used for instrinsic tactile sensing throughout this work. In **Publication 4** it is furthermore demonstrated how to include visual detections in the likelihood based on the approach developed together with Sachtler et al. [Sac+19].

3.6 Adaptive Task Execution

The robotic skill is the core element of the flexible robotic assembly system. It is an object-centric and reusable implementation of a robotic behavior to reach the goals from the task specification (see Section 3.1). In order to be able to deal with varying conditions and uncertainties an adaptive task execution needs to be achieved.

The awareness about the present situation obtained from contact sensing (Section 3.5) allows to implement such an adaptive execution inside of the skills. The combination of a (Bayesian)

[61] A sharp transition with a binary function was used in [Sac+19].

[62] Compare also the statement by Lefebvre et al. [LBD03, p. 29] and the mentioned modeling of the closure condition as a state constraint in the measurement equations of a Kalman Filter in [De +99, p. 1171].

state estimator with decision-making about future actions applied in this work is also known in a related form in the field of motion planning under uncertainties (e.g., [KL13]). Furthermore, *visual servoing* describes methods in which the motions of a manipulator are guided by the visual sensor input.[63] The methods developed in this work mainly focus on using the input from intrinsic tactile sensing and use it to adapt the assembly motions. The approach may be interpreted as a form of *tactile servoing*.

This section presents and summarizes the methods developed in this work, which use state estimation as a basis for the implementation of adaptive assembly skills. First, the adaptive path executor is described in Section 3.6.1, which enables the robotic system to follow paths specified relative to (non-static) objects. Then in Section 3.6.2, the developed motion generation algorithm is presented that uses the sample distribution of the estimator to locally guide the parts during assembly tasks.

3.6.1 Adaptive Path Executor

Many existing assembly strategies assume static environments. Poses of the objects are fixed, and need to be identified before the actual execution of the task starts with costly calibration routines. In contrast, the adaptive path executor presented in **Publication 4** can be applied inside of a robotic skill to execute assembly motions under the presence of pose uncertainties, which may arise due to inaccurately known poses of workcell components or moving parts. As a result, the effort required to design and implement the application can be reduced, and the overall system can become more flexible.

Following the object-centric paradigm of robotic skills (see Section 3.1.2), the path to be executed is given with respect to object frames as a reference, as opposed to globally fixed reference frames. In particular, the path is specified as a sequence of transformations $\boldsymbol{H}_{CT,l} \in \mathcal{T}, l \in \{1, ..., L\}$ (see Fig. 16). The frame T with index l denotes a path point for the manipulated object with frame D. The final configuration is given by the transformation $\boldsymbol{H}_{CD} = \boldsymbol{H}_{CT,L}$ relative to the object in the environment with frame C. The feasibility of executing such an object-centric path with a robotic manipulator, for example, with respect to reachability, can be verified using task specific workspace maps as developed together with Bachmann et al. [Bac+20].

The path already includes the knowledge about an appropriate strategy to solve the task. Such a strategy might be generated offline by fine motion planners [LMT84; Don86; Erd86; RBS05; SAH07; Wir+18], or specified by a user through demonstration or CAD interfaces. In this work, the method by Stemmer et al. [SAH07] is applied as an example to define a nominal assembly motion. The method allows to analyze the shape of the parts and identifies a suitable geometric region to execute a compliant assembly motion such that the parts can be mated successfully.

[63] See, e.g., the survey of [KC02].

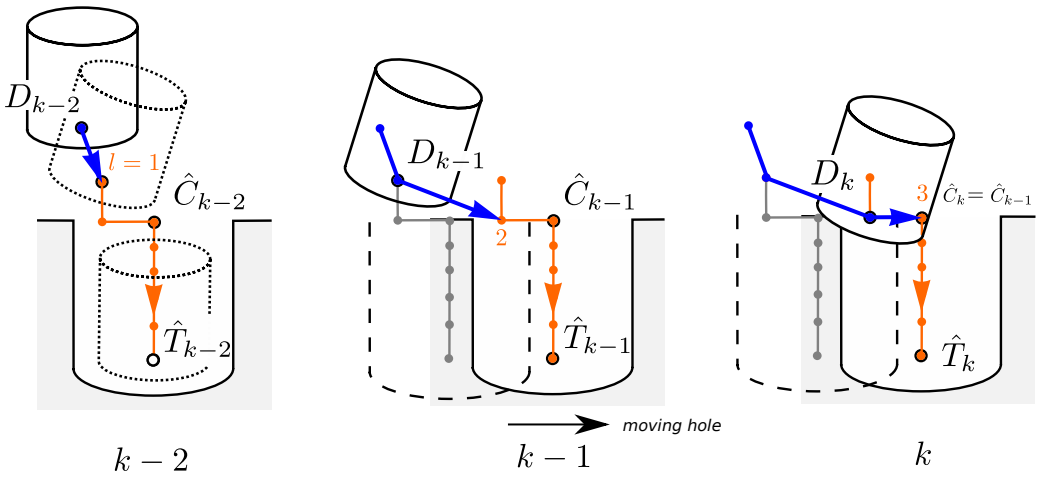


Figure 16: Adaptive execution of an object-centric motion strategy considering the currently estimated pose of the hole. Source: Taken and adapted from **Publication 4** (Fig. 7)

The implemented adaptive motion generation algorithm uses the estimate of the relative pose $\hat{\boldsymbol{H}}_{CD}$ between the parts obtained from contact sensing as a progress measure during execution. As soon as a desired intermediate path point $\boldsymbol{H}_{CT,l}$ is reached, the next motion goal is scheduled. Thereby, the skill adapts the motion to the currently estimated part pose. Here, the Euclidean d_T and geodesic distance d_R for the rotation component of the path deviations are used.[64] Alg. 6 summarizes the general implementation of the adaptive path executor.

```
1:  function GenerateMotion(𝒯)
2:      for l := 1 to L do
3:          reached ← false
4:          while not reached do
5:              y_k = (q_k, τ_k) ← GetMeasurements( )
6:              𝒳_k ← DoEstimationStep(𝒳_{k−1}, y_k)
7:              Ĥ_BC ← EstimatePartPose(𝒳_k)
8:              Ĥ_CD ← GetRelativePose(Ĥ_BC, q_k)
9:              d_T, d_R ← GetDistances(Ĥ_CD, H_CT,l)
10:             if d_T ≥ d_T,max ∨ d_R ≥ d_R,max then
11:                 H_BD,d ← ComputeNextMotion(Ĥ_BC, H_CT, l)
12:             else
13:                 reached ← true
14:                 H_BD,d ← ComputeNextMotion(Ĥ_BC, H_CT, l+1)
15:             ExecuteMotion(H_BD,d)
16:             k ← k + 1
```

Algorithm 6: Motion generation algorithm of the adaptive task executor.

[64] See, e.g., [Huy09] for a discussion of distance functions and metrics for rotations.

3.6.2 Locally Guided Motion Generation

The general adaptive path executor does not explicitly use further information from the estimator besides the expected values of the particle distribution to estimate the pose of the part in the environment. However, the distribution of the particles provides hints on the local shape of the relative configuration space, which can be exploited to generate a motion towards a motion goal. The motion goal might be given as a single target frame T specified relative to the part in the environment with a relative transformation $\boldsymbol{H}_{CT} = \text{const.} \in SE(3)$.

In **Publication 5**, a method is proposed and evaluated that guides an object to the target frame using the information about the relative configuration space between the objects in contact. This information is captured by the spatial distribution of the particles. During the estimation, the particles distribute in the feasible regions of the relative configuration space.

The particles in the neighborhood $B_m(\hat{C}) = \{{\boldsymbol{x}_k}^{(i)} \mid \text{dist}({\boldsymbol{H}_{BC}}^{(i)}, \hat{\boldsymbol{H}}_{BC}) < r_m\}$ of the currently estimated pose of the object frame $\hat{C}$ can be analyzed to determine possible and desired relative motions (see Fig. 17). For this purpose a *task-specific weight* ${L_k}^{(i)}$ is assigned to each particle in the neighborhood:

$$ {L_k}^{(i)} = L({\boldsymbol{x}_k}^{(i)}) = \text{dist}({\boldsymbol{H}_{BT}}^{(i)}, \boldsymbol{H}_{BD,k}), \tag{3.56} $$

which measures a distance between the manipulated object frame D and the frame $T^{(i)}$. The frame $T^{(i)}$ is the target frame in the hypothetical case that the object in the environment is located at ${\boldsymbol{x}_k}^{(i)}$, that is, ${\boldsymbol{H}_{BT}}^{(i)} = \boldsymbol{H}_{BC}({\boldsymbol{x}_k}^{(i)})\boldsymbol{H}_{CT}$.

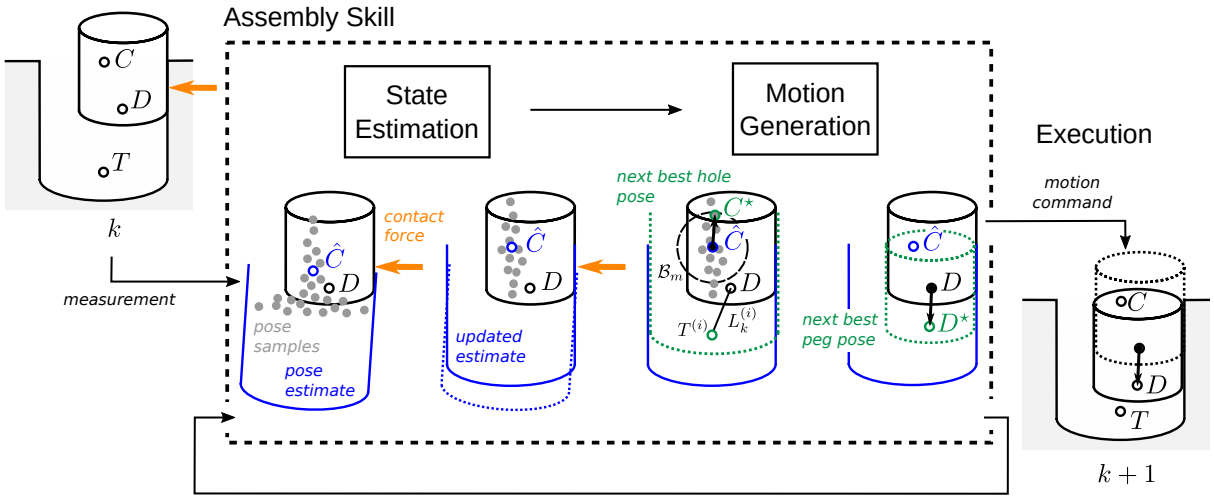


Figure 17: Locally guided motion generation as part of an assembly skill following the state estimation step and reusing the information about the local configuration space contained in the distribution of the samples. Source: Based on Fig. 1 in **Publication 5**

A local search for the smallest task-specific weight provides the hypothetical pose of the object $\boldsymbol{H}_{BC}{}^{\star}$ that reduces the distance to T:

$$\boldsymbol{H}_{BC}{}^{\star} = \underset{\boldsymbol{H}_{BC} \in B_m}{\arg\min}(L_k{}^{(i)}). \tag{3.57}$$

This hypothetical pose can then be converted to a executable motion by switching the reference frame to the frame of the grasped object D, using $\boldsymbol{H}_{DD^{\star}} = \boldsymbol{H}_{D\hat{C}}\boldsymbol{H}_{C^{\star}\hat{C}}\boldsymbol{H}_{\hat{C}D}$, which delivers $\boldsymbol{H}_{BD,k+1} = \boldsymbol{H}_{BD,k}\boldsymbol{H}_{DD^{\star}}$.

The procedure for computing a local motion, which guides the object to the target frame, is summarized in Alg. 7. The function COMPUTELOCALMOTION is called within the adaptive skill at each cycle after the DOESTIMATIONSTEP.

function COMPUTELOCALMOTION($\hat{\boldsymbol{H}}_{BC}, \boldsymbol{H}_{CT}, \mathcal{X}_k$)
  $\mathcal{B}_m(\hat{C}) \leftarrow$ GETNEIGHBORHOOD($\hat{\boldsymbol{H}}_{BC}, \mathcal{X}_k$)
  for all $\boldsymbol{x}_k{}^{(j)} \in \mathcal{B}_m(\hat{C})$ **do**
    $L_k{}^{(j)} \leftarrow$ WEIGHTSAMPLETASKSPECIFICALLY($\boldsymbol{H}_{CT}, \boldsymbol{x}_k{}^{(j)}, \boldsymbol{y}_k$)
  $\mathcal{P}_L \leftarrow \{(L_k{}^{(j)}, \boldsymbol{x}_k{}^{(j)})\}$
  $\boldsymbol{H}_{BC}{}^{\star} \leftarrow$ SELECTBESTPARTPOSE($\mathcal{P}_L$)
  $\boldsymbol{H}_{BD,d} \leftarrow$ CONVERTTOROBOTMOTION($\hat{\boldsymbol{H}}_{BC}, \boldsymbol{H}_{BC}{}^{\star}$)
return $\boldsymbol{H}_{BD,d}$

Algorithm 7: Locally guided motion generation using task specific sample weights.

Note that instead of reusing the samples from the estimation step, new samples could be generated in a neighborhood around $\hat{C}$. The exploration of new motions could then be made independent of the estimation, but would also require additional efforts to sample from the configuration space $\mathcal{C}$. In contrast to PRM, the motion is generated online and samples are present with a density around the current pose such that fine motions can be planned when needed.

The method provides the capability to adapt the motion locally, however for complex geometries it can get stuck in local minima due to its gradient descent character. This can be avoided by combining the approach with an appropriate strategy using the adaptive path executor in which then the function COMPUTELOCALMOTION is called instead of COMPUTENEXTMOTION. Alternatively, further reasoning could be integrated in future work to detect local minima and actively decide to continue exploration of the configuration space.

4 Summary of Publications

This chapter provides one-page summaries of the publications on which this dissertation is based.

- **Publication 1** introduces a flexible robotic assembly system combining planning methods and reusable assembly skills (Research Question 1).
- **Publication 2**, **Publication 3** and **Publication 4** provide methods for the observation of assembly tasks using intrinsic tactile sensing (Research Question 2).
- **Publication 4** and **Publication 5** describe methods usable in the implementation of robotic skills for adaptive execution of assembly tasks (Research Question 3).

For each publication the reference and abstract is provided. The individual contributions of the author of the dissertation are summarized textually for each publication.

Additionally, the contributions of all authors are provided using the roles defined by *CRediT*.[1] Further contributions of other persons mentioned in the acknowledgments are listed in gray text color. Note that the roles *funding acquisition*, *project administration* and *resources* are not documented.

The full text versions of the publications are enclosed in the **Appendix** together with copyright information.

Furthermore, all publications can be found using the ORCID iD[2] of the author of the dissertation ORCID: 0000-0002-6016-6235.

[1] *Contributor Roles Taxonomy*, http://credit.niso.org/
[2] *Open Researcher and Contributor ID*, https://orcid.org

Publication 1: Flexible Robotic Assembly System

Reference and Abstract

Nottensteiner, K., T. Bodenmüller, M. Kaßecker, M. A. Roa, A. Stemmer, T. Stouraitis, D. Seidel, and U. Thomas (2016): "*A Complete Automated Chain for Flexible Assembly using Recognition, Planning and Sensor-Based Execution*". 47st International Symposium on Robotics (ISR).

Full text of the publication enclosed in the Appendix, reference in bibliograpy [Not+16a].

Abstract – This paper presents a fully automated system for automatic assembly of aluminum profile constructions. This high grade of automation of the entire process chain requires novel strategies in recognition, planning and execution. The system includes an assembly sequence planner integrated with a grasp planning tool, a knowledge-based reasoning method, a skill-based code generation, and an error tolerant execution engine. The modular structure of the system allows its adaptation to new products, which can prove especially useful for SMEs producing small lot sizes. The system is robust and stable, as demonstrated with the repeated execution of different geometric assemblies.

Author's Contributions

The author of the dissertation contributed substantially to the publication. He refined the skill concept and integrated existing components for assembly grasp and sequence planning [TW10; TSR15], as well as motion planning and the component for visual assembly recognition. With the support of the co-authors, he implemented the method for mapping tasks to robotic skills and the skill library for the robot control using the execution engine and integrated it in the entire system architecture. He furthermore carried out the experimental validation of the overall process chain, contributed substantially to multiple parts of the original draft and created the visualizations (except Fig. 2).

CRediT: **K. Nottensteiner**: Conceptualization, investigation, methodology, software, visualization, writing – original draft **T. Bodenmüller**: Conceptualization, methodology, software, supervision, writing – review & editing **M. Kaßecker**: Conceptualization, investigation, methodology, software, writing – review & editing **M. Roa**: Conceptualization, methodology, supervision, writing – original draft **A. Stemmer**: Conceptualization, investigation, methodology, software, supervision, writing – review & editing **T. Stouraitis**: Conceptualization, investigation, methodology, software, writing – review & editing **D. Seidel**: Conceptualization, investigation, methodology, software, writing – review & editing **U. Thomas**: Conceptualization, investigation, methodology, software, supervision, writing – original draft M. Danzer, C. Scheurer, U. Zimmermann: Software.

Publication 2: Observation of Robotic Assembly Tasks

Reference and Abstract

Nottensteiner, K., M. Sagardia, A. Stemmer, and C. Borst (2016): "*Narrow Passage Sampling in the Observation of Robotic Assembly Tasks*". IEEE International Conference on Robotics and Automation (ICRA).

Full text of the publication enclosed in the Appendix, reference in bibliograpy [Not+16b].

Abstract – The observation of robotic assembly tasks is required as feedback for decisions and adaption of the task execution on the current situation. A sequential Monte Carlo observation algorithm is proposed, which uses a fast and accurate collision detection algorithm as a reference model for the contacts between complex shaped parts. The main contribution of the paper is the extension of the classic random motion model in the propagation step with sampling methods known from the domain of probabilistic roadmap planning in order to increase the sample density in narrow passages of the configuration space. As a result, the observation performance can be improved and a risk of sample impoverishment reduced. Experimental validation is provided for a peg-in-hole task executed by a lightweight-robot arm equipped with joint torque sensors.

Author's Contributions

In order to describe the assembly tasks, the author of the dissertation developed the model of the geometric uncertainties in the kinematic chain and combined it with a contact model based on the voxelmap-pointshell algorithm, whose implementation was provided by M. Sagardia. The sampling methods in the propagation model based on approaches from probabilistic roadmap planning were adapted and integrated by the first author into a state estimator for the observation of assembly tasks under the supervision and with the support of the co-authors. The evaluation of the method including the data curation and analysis based on simulations and experiments with the robotic system were carried out solely by the first author, as well as the writing of the original draft of the publication.

CRediT: **K. Nottensteiner**: Conceptualization, data curation, formal analysis, investigation, methodology, software, visualization, writing – original draft **M. Sagardia**: Methodology, software, visualization, writing – review & editing **A. Stemmer**: Conceptualization, methodology, supervision, writing – review & editing **C. Borst**: Supervision, writing – review & editing.

Publication 3: Constraint-based Sample Propagation

Reference and Abstract

Nottensteiner, K. and K. Hertkorn (2017): "*Constraint-based Sample Propagation for Improved State Estimation in Robotic Assembly*". IEEE International Conference on Robotics and Automation (ICRA).

Full text of the publication enclosed in the Appendix, reference in bibliograpy [NH17].

Abstract – In fast changing assembly scenarios, it is required to adapt the task execution to the current state of the setup without extensive calibration routines. Therefore, it is important to estimate the geometric uncertainties and contact states during the assembly execution. We use a sequential Monte Carlo (SMC) method to track the relative poses between workpieces during a robotic assembly based on joint torque and position measurements only. In contrast to existing approaches, we focus on assembly tasks where the workpiece is not fixed in the workcell, but can, for example, slide on a table surface. We propose a new constraint-based propagation model for the SMC approach: a compensation motion for the samples dependent on the violation of contact constraints is derived. This allows us to track the motion of the workpieces in cases where a common random diffusion model fails. The method is evaluated with experiments using an assembly scenario with two KUKA LBR iiwa robot arms and shows accurate tracking performance.

Author's Contributions

The author of the dissertation derived theoretically the model for constraint-based propagation of part motions in the observation framework and implemented it for the robotic testbed with two manipulators. The model was discussed and analyzed with the support of the co-author K. Hertkorn. The experimental evaluation of the method and the comparison with an alternative velocity-based method was carried out by the first author, as well as the visualizations and documentation in the publication.

CRediT: **K. Nottensteiner**: Conceptualization, data curation, formal analysis, investigation, methodology, software, visualization, writing – original draft **K. Hertkorn**: Conceptualization, formal analysis, methodology, writing – review & editing F. Stulp, A. Stemmer, M. Kaßecker: Writing – review & editing M. Sagardia: Software.

Publication 4: Adaptive Assembly Task Execution

Reference and Abstract

Nottensteiner, K., A. Sachtler, and A. Albu-Schäffer (2021): "*Towards Autonomous Robotic Assembly: Using Combined Visual and Tactile Sensing for Adaptive Task Execution*". Springer Journal of Intelligent & Robotic Systems (JINT).

Full text of the publication enclosed in the Appendix, reference in bibliograpy [NSA21].

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.

Author's Contributions

The concept for the adaptive task execution was designed and modelled by the author of the dissertation with the support of the co-authors. Ideas discussed with A. Stemmer on robust assembly strategies helped to develop the approach. In particular, the tilt-and-align strategy from [SAH07] was adopted in this work. The visual likelihood model and the fusion with a tactile likelihood for multiple features was published together with the co-authors in [Sac+19]. The implementation of the framework was done by the author of the dissertation and used software components from A. Sachtler and M. Kaßecker for the visual detection and the haptic rendering algorithm of M. Sagardia. The evaluation was carried out and analyzed with the support of A. Sachtler and under supervision of A. Albu-Schäffer. The author of the dissertation made all figures and wrote the original draft of the publication.

CRediT: **K. Nottensteiner**: Conceptualization, data curation, formal analysis, investigation, methodology, software, visualization, writing – original draft **A. Sachtler**: Conceptualization, data curation, investigation, methodology, software, writing – review & editing **A. Albu-Schäffer**: Conceptualization, supervision, writing – review & editing A. Stemmer: Conceptualization M. Kaßecker, M. Sagardia: Software M. Roa: Writing – review & editing.

Publication 5: Locally Guided Peg-In-Hole

Reference and Abstract

Nottensteiner, K., F. Stulp, and A. Albu-Schäffer (2020): "*Robust Locally Guided Peg-in-hole with Impedance Controlled Robots*". IEEE International Conference on Robotics and Automation (ICRA).

Full text of the publication enclosed in the Appendix, reference in bibliograpy [NSA20].

Abstract – We present an approach for the autonomous, robust execution of peg-in-hole assembly tasks. We build on a sampling-based state estimation framework, in which samples are weighted according to their consistency with the position and joint torque measurements. The key idea is to reuse these samples in a motion generation step, where they are assigned a second task-specific weight. The algorithm thereby guides the peg towards the goal along the configuration space. An advantage of the approach is that the user only needs to provide: the geometry of the objects as mesh data, as well as a rough estimate of the object poses in the workspace, and a desired goal state. Another advantage is that the local, online nature of our algorithm leads to robust behavior under uncertainty. The approach is validated in the case of our robotic setup and under varying uncertainties for the classical peg-in-hole problem subject to two different geometries.

Author's Contributions

The concept for the locally guided task execution and the method for reusing the samples from an estimator in a motion generation step was developed by the author of the dissertation and discussed with the co-authors, who supervised the work. The experimental evaluation with the robotic system, analysis and documentation of the results in the publication were carried out by the first author. F. Stulp supported the writing of the original draft and helped to increase the overall quality, especially by streamlining the introduction and improving the general style of presentation.

CRediT: **K. Nottensteiner**: Conceptualization, data curation, formal analysis, investigation, methodology, software, visualization, writing – original draft **F. Stulp**: Conceptualization, supervision, visualization, writing – original draft, writing – review & editing **A. Albu-Schäffer**: Conceptualization, supervision, writing – review & editing.

5 General Discussion

This work investigated how to realize a flexible robotic assembly system that reduces the manual effort in the overall pipeline from product design to assembly execution. Such systems allow to get closer to the ultimate vision of mass customization, in which manufacturing processes of custom products are planned and executed with similar efforts as high volume products. This chapter will discuss what has been accomplished so far and what further development steps are possible.

5.1 From Flexible Robotic Assembly Systems to Production Networks

The framework for flexible robotics assembly developed in this work provides the ability to solve assembly tasks within known classes of problems using a dedicated robotic system. It was shown that product variants can be assembled without knowing the exact configuration of possible products during the setup and implementation of the system. Variations in the spatial configuration of the desired structure and the number and type of elements are possible, which covers theoretically an infinite number of distinguishable products. Compared to related work (Section 1.3), a high number of possible product variants can be treated, and furthermore, delicate assembly tasks that require fine manipulation skills can be handled.

At the core of the framework are reusable and adaptable robotic skills. These skills can be configured by using only a few parameters such that they can be applied to various tasks in the manufacturing processes of individual products. Contact sensing, compliant control and strategies, as well as integrated planners for geometric reasoning are used to adapt to the given situation. Thus, the considered robotic system implements multiple forms of flexibility to reduce the manual effort required to produce individual products. This includes the flexibility to find appropriate grasps and assembly sequences, which was extended and investigated in more details in [Rod+19; Rod+20], and furthermore, the flexibility to find appropriate placements of parts and components in the workspace, which was further developed based on the results of this work in [Bac+20; BNR21]. Using the planners and the robotics skills, solutions for assembly tasks can be solved autonomously on multiple stages and executed reliably under varying conditions.

However, further developments are necessary in order to scale up the concept from a single robotics assembly system to fully reconfigurable and flexible production systems and factories. As stated in the introduction, concepts for such systems were proposed by [Kor+99; ElM06; Wie+07; Zäh+09; SVB15] and others. For example the *cognitive factory* by Zäh et al. is a factory, "where machines and processes are equipped with cognitive capabilities in order to allow them to assess and increase their scope of operation autonomously" [Zäh+09, p.355].

An important step towards the realization of such a vision is that the knowledge about the capabilities and skills of production entities, in particular robots, is made explicit and accessible, as suggested for example by [Bjö+11; Bjö+12]. *Ontologies* [GOS09] are commonly used to represent and share knowledge, and were also applied in the field of autonomous robots to describe tasks and systems [Oli+19]. Examples of ontological descriptions in the assembly domain are given by [Per+15; Bal+17; Pan+20]. Using such knowledge representations in a production network will increase the overall flexibility as concepts and solutions can be shared between various entities, and can then be used during planning and execution to reduce the manual effort. Robotic systems, human co-workers, and agents [FG96] in general can offer their capabilities in such a network, realizing a fully reconfigurable and flexible production system.

Within the project *Factory of the Future* at DLR concepts for realizing such a flexible production network are investigated. An ontological description was developed [Sch+21b] based on the IEEE Std 1872™-2015 [IEE15] standard, which further formalizes the concepts of the robotic assembly system of this work and merges it with concepts and description of other developments. The midterm goal is to demonstrate a flexible production network in which multiple robotic stations can be reconfigured autonomously with respect to flexibly assigned tasks and to execute assembly tasks autonomously, as well as in cooperation with human co-workers.

5.2 From Contact Sensing to Scene Awareness

The methods for contact sensing proposed and evaluated in this work provide the capability to reduce geometric uncertainties during skill execution and monitor the progress towards reaching a goal configuration. This allows for adaption to the situation and reduces the need for accurate calibration steps beforehand. As a result, contact sensing provides greater flexibility as requirements on accuracy are minimized and efforts to achieve accuracy in the production facilities become smaller. In **Publication 2**, **Publication 3** and **Publication 4** multiple methods are proposed to support assembly scenarios in static setups and in setups with moving parts (see Section 3.5). The chosen contact sensing approach is embedded in a Bayesian framework using particle filters for state estimation. In the following, advantages

and limitations are discussed, as well as how the approach could be extended to provide more scene awareness to robotic systems.

Although this framework is open to being used with data-driven models, an approach was chosen that relies on explicit contact and kinematic models. For example, the methods presented in this work strongly depend on accurate geometric models in the form of 3D-meshes. The used implementation of the contact model supports non-convex geometries in a high spatial resolution. This allows handling a broad class of part geometries and obtaining accurate estimates, which was at this level not yet demonstrated in related work for complex geometries.

However, there are physical effects that are difficult to model efficiently in the current approach. These include realistic friction effects in multi-contact situations, large deformations, and flexible bodies, as well as mechanisms and subassemblies with many degrees of freedom. Overall, there are two distinct development directions to solve these issues.

On the one hand, more complex environments and physical effects can be modeled inside dedicated physical simulations and embedded in this estimation framework, e.g, as successfully demonstrated by Wirnshofer [Wir21]. Nevertheless, the dependency on accurate geometric and physical models is not vanishing, and the required fidelity of the simulator might conflict with real-time constraints in an online application. Consider, for example, the realistic modeling of a snap-fit connection that requires specialized and computationally expensive simulations using finite-element analysis (FEA) or comparable approaches to handle the non-linear dynamics.[1] A common pragmatic approach in robotics is to use simplified mechanical models, but these might not be able to generalize well and require manual modeling efforts.

On the other hand, an alternative to modeling the details is to use learned representations of contacts. However, learning such representations typically requires a large number of training examples, and it is unclear how well they generalize over complicated geometries. A promising future trade-off might be the combination of physical simulations and learned models, as, for example, the *data-augmented physics engine* suggested by [Aja+19]. In conclusion, an appropriate balance between data-driven and model-based methods need to be found to represent complex scenes within the estimation framework.

A further aspect of providing more scene awareness relates to the dimensions of the state that represents the space to be observed. In this work, only a few state variables are considered. However, in order to handle more complex scenes, it is desirable to support states with larger dimensions, which shall be discussed in the following.

In general, the rate of convergence of a particle filter does not directly depend on the state dimension and therefore does not suffer from the *curse of dimensionality* in the classical

[1] See for example [JS09] for handling snap-fits in simulation. [Wah+15] uses a force-over-position profile of a simplified mechanical model for a robotic assembly skill; see also [Sto15] for robotic snap fit assembly.

sense as other estimation methods do [CD02, p.744]. However, the computational complexity increases for higher dimensions [DH03]. As a rule of thumb, state dimensions > 10 are considered to be challenging (e.g. [Che03, p.33]), and might require dedicated strategies and methods. The consequences of high-dimensionality on particle filters are specifically discussed by [DH03; Che03, p.53f; BLB08; Sny+08]. Critical issues are an increased risk of the distribution collapsing to the value of a single particle and the rising computational complexity with an increase of the number of particles to keep a high effective sample size. The number of effectively needed particles typically increases exponentially with the number of dimensions.[2]

In order to deal with these issues, various methods are proposed, for example, the block particle filter [RV15], the coordinate particle filter (CPF) [Wüt+15], the space-time particle filter (STPF) [Bes+17], or nested SMC [NLS19]. These approaches need to be considered in more detail as soon as the number of variables in the observed state space is increased more.

Further methods and techniques to handle the computational complexity are the online adaption of the number of particles, for example, via KLD-sampling [Fox03], and Rao-Blackwellized particle filters [CGM07, p.910ff], compare [ETM21, p.22ff]. Rao-Blackwellized particle filters use state marginalization to implement a staged estimation scheme in which only certain (nonlinear) parts of the state are represented with samples, and estimates of other parts are obtained from other sources such as a Kalman filter, for example. This approach was used by Taguchi et al. [TMO10] in the localization of objects by dividing the uncertain pose into a translational part and a part for the orientation, and thus showed a significant reduction in the number of required particles.

Furthermore, there are efficient techniques and libraries using parallel implementations for handling the computational complexity, for example: [Bas+03; HKG10; Chi+13; Zho14; Dem+14; Lin+17]. The parallel implementation also helps to reach sufficient observation rates for online applications with respect to real-time constraints.

Overall, methods for dealing with higher dimensions and handling the computational complexity are available and discussed in the filtering community, but must be carefully revisited and selected when larger scenes with multiple sources of uncertainties should be simultaneously observed.

A great advantage of the Bayesian framework is the possibility to incorporate various sources of measurement, which thus has the potential to improve the observability of states and improve scene awareness. In this work, the focus was put on intrinsic tactile sensing using joint torque sensors of a LWR, which has the advantage that no additional sensing device besides the robotic arm itself is needed. With the measurement model used, accurate estimates of object poses could be obtained from contact sensing.

[2] See [Che03, p.41] for an approximate measure of the effective sample size dependent on the volume of the search and target space. However, it strongly depends also on the given problem and used proposal densities.

However, the assumption was made that errors in the external joint torques are independent of the robot configuration and that they are distributed normally. This measurement model could be refined with experimental data obtained from measurements in various joint configurations to reach higher sensitivity when smaller contact forces are present. The alternative evaluation of the contact wrench in Cartesian space using a force/torque-sensor at the end-effector, which is independent of the joint configuration, is known from related work (e.g., [Mee+07]) and could also be considered again to reduce the effect of errors in the joint measurements and in order to support a larger class of robotic manipulators.

The combination of tactile and visual sensing in **Publication 4** showed that multiple modalities can be used to provide continuously accurate estimates throughout the overall assembly process. Therefore, the developed framework is suitable to incorporate various sensory inputs, which might be needed to increase the observability in more complex scenarios.

Once more awareness about the scene exists, it can be shared across mutliple components of the robotic system to enhance online planning and execution. In this work, the state estimates are already shared with a central world state component, which keeps track of the individual objects in the environment. However, the task planning components do not yet take uncertainties in the scene into account. In future developments, a goal could be to connect the probabilistic world state with reasoning entities on a logical layer, and connect the planning components closer to the perceived world state. In particular, this would be necessary if the system should not only deal with assembly but also with disassembly tasks, in which even more uncertainties are present in the process.

Overall, the combination of Bayesian estimators and a probabilistic representation of object states is a promising approach to provide more scene awareness and enable decisions at higher abstraction levels. This would potentially reduce the implementation effort even more and open up new fields of applications for autonomous robotic assembly systems.

5.3 From Adaptive Robotic Skills to Learning Systems

As demonstrated in this work, the awareness of a particular contact situation enables the implementation of adaptive assembly skills that can deal with present uncertainties (Section 3.6, **Publication 4**, **Publication 5**). The object-centric specification of assembly strategies together with a contact sensing implementation, which supports complex geometries, makes it possible to reuse skills in various situations and provides more flexibility and requires less implementation effort to solve assembly tasks. However, there are important aspects to be discussed in this section, which would make the system more performant and, furthermore, would allow dealing with a larger class of assembly scenarios.

Being flexible is definitely one of the key factors for successfully handling today's challenges in production with respect to fast and unforeseen market changes, larger product varieties, and smaller lot sizes (see Section 1.1). Besides the need for solutions that can be quickly and effortlessly adapted, it is also required that solutions are performant during execution. Applications in the industry show that dedicated and specialized production facilities can achieve very high performance when product volumes are high. For example, the *Sony Playstation 4*, with a volume of over 100 million sold units, is manufactured using an assembly line consisting of 32 robots that are finalizing one device every 30 seconds [Nit20]. In order to approach such performance levels with smaller quantities, an optimal setup is required as well as, in addition, the reliable and fast execution of manufacturing processes.

With respect to an optimal setup, the automated optimization of the layout of a robotic assembly system for given task sequences is investigated by [BNR21] for the two-arm version of the considered assembly system (see Section 3.1). With respect to the execution itself, a focus was put on handling geometric uncertainties in this work. Only single runs of skills were considered, and the goal was to achieve a successful execution at the first trial. Optimizing the performance with respect to execution time was not investigated, and data from past runs, which could be collected over multiple executions, was not yet used.

In related work, various approaches from the fields of machine learning and optimization are proposed that can be applied to speed up such robotic assembly processes automatically. For example, Nuttin and Van Brussel [NV97] demonstrated in simulation how the insertion time in a peg-in-hole task can be decreased while maintaining process quality using a reinforcement learning (RL) approach. [Hod+18] investigated how an automatic optimization of robot motions using reinforcement learning can be integrated into production environments. Recently, Johannsmeier et al. [JGH19] compared various learning algorithms for minimizing execution time of a manipulation skill, which is modeled as a sequence of parameterized motions according to a task-dependent structure.[3] It was shown that the required time can be efficiently reduced with a few trials and that human performance can be exceeded.

Overall, as shown in multiple works, performance can be optimized autonomously by the robotic systems, but challenging aspects are to provide an appropriate and safe training environment, to ensure a minimal time for optimization runs, and the transferability to new situations. In this context, the knowledge about the shape of the configuration space obtained during contact observation in this work could potentially also be used to support such an optimization.

Apart from the performance of a skill in a particular use case, it is generally desirable that skills can be applied to multiple problems in varying situations. As mentioned in Section 3.1.2, a skill implementation can be considered more powerful if it supports a larger range of tasks that can be solved. For example, a skill that can only handle screw connections using a

[3] See also [VJH21] for the latest extension of this approach.

specific type of screw is less powerful compared to a skill that can deal with screws of various types and lengths. In this work, the supported task range is mainly predetermined by the implemented solution and strategy envisioned by the developer. The implemented skills support a large class of objects, for example, objects for which grasps with a two-finger gripper are known or can be determined with the integrated grasp planner. The access to geometric planners in the developed system and also the approach in **Publication 5** enables the system to handle various geometries and assembly types. However, the current system is not able to autonomously expand its capabilities to handle new task types with new strategies. In future work, it should therefore be investigated how a learning assembly system can be implemented to reduce implementation efforts for new task types.

Skill acquisition and the learning of new assembly strategies could be based on different inputs. In **Publication 4** the model-based strategy of [SAH07] was applied to realize an efficient insertion process. Similarly, [JGH19; BDD95; Ino74] and others used modeled strategies for solving assembly tasks. Such modeled strategies are often inspired by human manipulation strategies but are implemented manually by the programmer or developer. Recent works in the field of kinesthetic teaching and imitation learning try to adopt such strategies directly from human demonstration, for example: [De 16; Kra+17; Vec+19; SRD19]. Another source for skill acquisition is represented by machine learning methods in general (also without human demonstration or supervision). For example, self-learning controllers for assembly were implemented early by using reinforcement learning and neural networks, for example, by [Sim+82; Asa90; GGB92; PC94]. Further variants of reinforcement learning and new approaches using deep learning and unsupervised learning became popular in recent years, for example, [Ino+17; Luo+18; Tho+18; Lee+19; Wu+21; Bel+22]. These rapidly evolving approaches could be an important component of autonomous learning systems that expand their capabilities through experience from past executions in simulated or real worlds.

6 Conclusion

Today's challenges in production motivate the development of flexible production systems. Larger product varieties are to be manufactured in a shorter time while using resources more efficiently. In addition to appropriate product design and general production planning and management, the use of flexible robotic assembly systems is a promising part of the solution to meet these requirements. Such systems could increase overall productivity by reducing manual effort throughout the process chain. In this context, robotic assembly systems have to deal with the uncertainty about which products are demanded in the future and the uncertainties during the actual execution of the physical manufacturing process. In this work, a framework for such a flexible robotic assembly system was proposed. This final conclusion provides a summary of the results and possible future research directions.

6.1 Summary

The first central research question in this work is how to assemble individual products with less manual effort using robots equipped with advanced sensing capabilities (Research Question 1). The proposed solution is a planning and execution framework that accepts assembly specifications of desired products as input and immediately manufactures them without any further manual steps. An integrated assembly planner finds suitable task sequences by analyzing the product structure and part geometry. In order to solve the tasks, robotic skills are developed that can be mapped to tasks with a task classification approach. These skills are a central component of the execution framework and are implemented once by a robotic expert in a reusable and object-centric fashion such that they can be applied in multiple situations and for various parts. The assembly motions within the skills are executed with respect to local features of the objects. The skills have access to planning units that provide solutions for dedicated problems, for example, for grasp selection or planning of collision free paths. It was shown that, without further manual programming efforts, the system can assemble individual products that are composed out of a set of known part types. This was demonstrated for structures made out of aluminum profiles with a varying number of parts and various geometric configurations (see Fig. 18).

In general, the fact that each product has an individual structure makes it impossible to optimize the setup and execution for all possible cases beforehand. The geometrically and logically planned assembly sequence therefore requires a robust physical execution as a supplement. Here, the impedance-based compliant control of the used LWR enables a robust and reliable assembly in the presence of uncertainties.

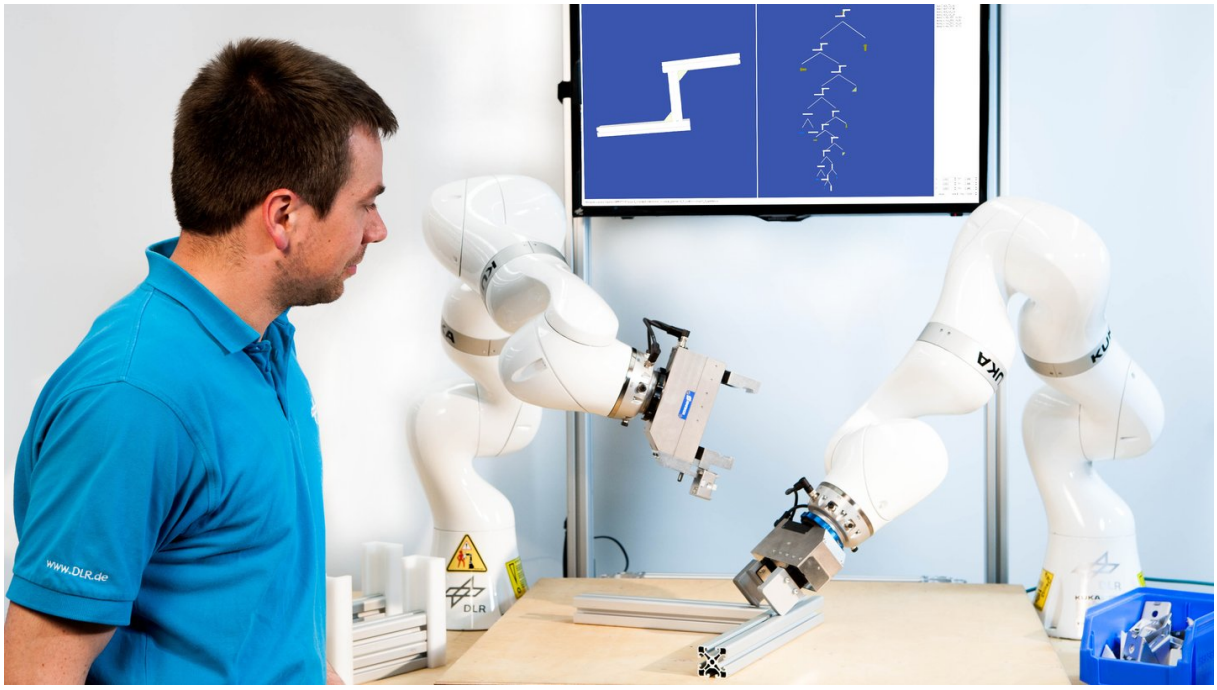


Figure 18: Assembly of aluminum profile structures with a flexible robotic assembly system. Source: Courtesy of DLR, CC-BY 3.0, also appearing in **Publication 4** (Fig. 1)

In addition to the passive compliant features of the used robot type in the implementation of the skills, it was investigated how the task execution can be observed using contact sensing and state estimation. Contact sensing and state estimation should enable the robotic system to actively react to the current situation and compensate for geometric uncertainties in the setup. This is of special interest for flexible scenarios where the need for accurate calibrations of workcell components should be reduced or specialized fixtures should be removed.

In particular, the investigated research question was how intrinsic tactile sensing using joint torque sensors can be implemented (Research Question 2). Due to its capabilities to handle nonlinear and non-Gaussian problems, as they are present in multiple forms during contact-rich tasks, a recursive Bayesian estimation framework based on SMC (i.e., particle filtering) was implemented. The primary focus was put on part localization during insertion tasks. Such tasks are a critical benchmark and are difficult to observe due to the tight constraints in the configuration space. However, the approach can similarly be applied to less constrained manipulation tasks and provides a general framework. It was shown how heuristics inspired by PRM (i.e., motion planning) can improve observation performance in narrow passages and

reduce the risk of sample impoverishment in contact-rich tasks. A contact model known from haptic rendering was integrated such that objects with complex and non-convex geometries can be handled.

Furthermore, methods dealing with moving objects were investigated, namely a constraint-based propagation model was derived and a constant velocity tracking model was applied. The estimator mainly uses the joint positions and external joint torques for the update. Additionally, it was shown how visual feature detection can be used in combination with contact sensing. The developed methods allow accurate online estimation in parallel to the execution. The overall observation approach uses explicit models of the kinematics and the contacts and thereby is suitable for industrial settings in which structured information about the tasks is available.

In a flexible production network, the robotic assembly skills should be reused in multiple situations with varying constraints and uncertainties such that the implementation effort for the specific solution vanishes. Therefore, the third research question refers to the adaptive execution of assembly tasks using the contact information from online observation (Research Question 3). In particular, methods were developed that incorporate the sensory information from intrinsic tactile sensing.

First, an adaptive motion generation algorithm was developed, which can be used to execute an assembly strategy with respect to the currently estimated part poses. The motion generator is demonstrated with an object-centric peg-in-hole skill, which is reusable for different part combinations, different initial positions, and also for scenarios with moving parts. The skill leverages a robust tilt-and-align assembly strategy implemented with a Cartesian impedance controller. In general, it is possible to execute arbitrary paths, which are defined with respect to an object feature frame. The path execution is monitored and the motion commands are chosen online according to the current progress. Compared to a solely passive compliant approach, the method tracks the parts over longer distances and supports various situations as it is able to deal with non-contact situations through the integration of visual perception and with occlusions with the help of tactile sensing.

While this method only uses the pose estimate obtained from contact sensing, the method studied second takes into account the local shape of the relative configuration space between the parts. The samples of the estimator approximate this space, and accordingly the next best motion can be selected in the neighborhood of the currently estimated relative part pose. This approach shows one possibility of how the information about the locally sampled configuration space can be exploited for motion generation. Overall, the methods for adaptive task execution developed in this work enable reusing skills in various situations.

6.2 Future Directions

The principal assembly framework consisting of a task planning unit and a skill-based execution, which was developed in this work, is also the basis for an extended dual-arm robotic assembly system [Rob20a], on which multiple topics associated with flexible assembly systems are researched in more depth [Rod+19; Sac+19; Rod+20; Bac+20; BNR21; Sch+21b]. Furthermore, concepts were transferred and expanded into the domain of the assembly of modular structures in space [Roa+17; Mar+21; Rod+21; Roa+22](see Fig. 19).

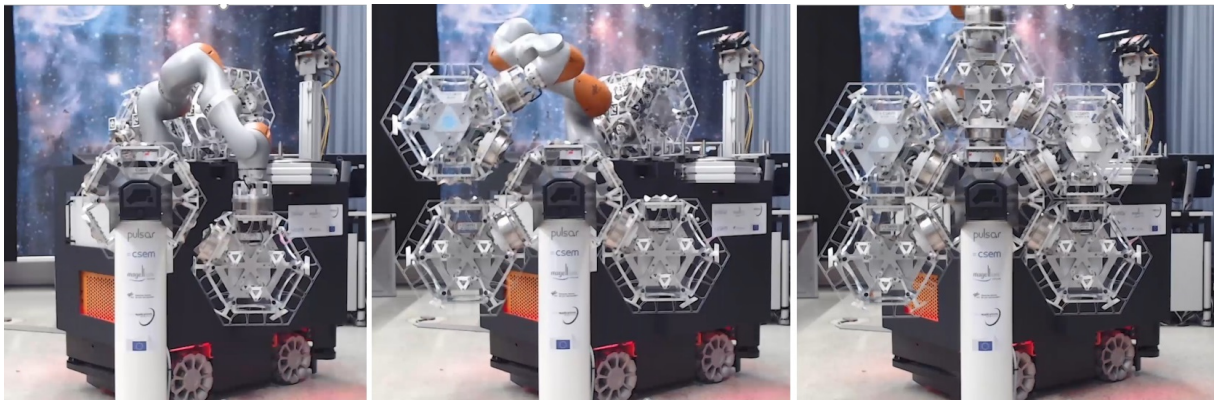


Figure 19: Application of the developed concepts for in-space assembly as showcased with demonstrator for precise assembly of mirror tiles of the H2020 PULSAR project using a KUKA KMR iiwa system and robust assembly skills for single, double and triple connections (left to right). Source: Figure from [Roa+22](Fig.8)

The investigation of flexible robotic assembly systems is continued within the DLR project *Factory of the Future* and associated research activities. In particular, variable workstations, consisting of the new lightweight robot SARA (Safe Autonomous Robotic Assistant) [Rob20b] at their core, will provide the capability to adapt the setup autonomously according to the given task, which includes the spatial arrangement of workcell components as well as the skill set of the robotic agent by interchanging tools and sensors flexibly in a production network (see Fig. 20, left). Thus, the flexibility of the robotic systems with respect to different products will significantly be increased compared to the system presented in this work.

The proposed combination of state estimation and motion generation in the implementation of a robotic skill has shown that it can increase the reusability and the robustness against uncertainties in various situations. A prerequisite for the approach is the availability of geometry and uncertainty models. As the used particle filtering method requires the evaluation of multiple hypotheses at each time step, a trade-off was implemented between physical accuracy and model complexity. Interesting in this context is the integration of more advanced simulation steps in the filtering framework, which could be investigated in future work. Furthermore, it could be researched how models can be enriched with data from previous executions or even completeley replaced by data-based approaches, reducing the need for exact and explicit models.

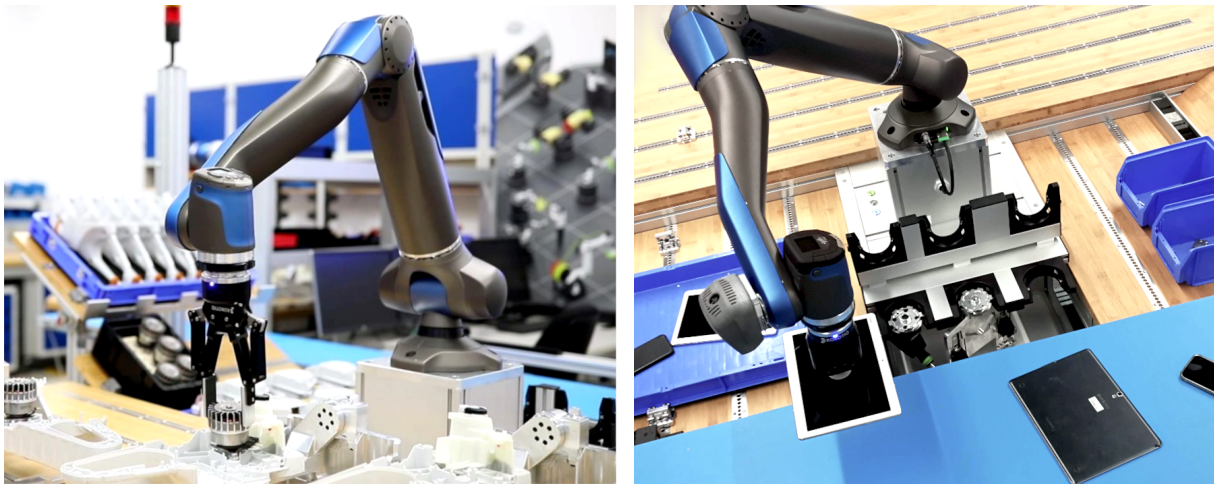


Figure 20: DLR SARA integrated in a variable workstation for flexible assembly execution (left). The autonomous handling of end-of-life products is an essential part for future production scenarios (right). Source: Courtesy of DLR

An important aspect, which was not yet investigated in this work, is to improve the performance of skill execution while maintaining flexibility and process quality. There are multiple possible directions to increase the overall performance. On the one hand, the methods for observation and online motion generation require an implementation such that they run fast and resource-efficient in order to keep track of the changes in more complex scenes in real-time. Several improvements are here possible in the implementation of the current methods. On the other hand, it is required that the robotic assembly system is able to autonomously improve the performance of the skill execution. Therefore, the overall framework of the assembly system requires the connection to a learning framework in which performances of the skill execution can be improved and novel solutions can be generated through learning and optimization methods.

The field of application of this work mainly deals with the manufacturing of new custom products. However, in order to achieve sustainability goals in future markets, it is critical to consider the entire product life-cycle. It can be expected that producers will have more responsibility to take back end-of-life products and establish a circular economy. Robotic disassembly is a technology that could support to realize a circular economy, see, for example, the disassembly robots of Apple Inc. [App21, p.45]. From a process perspective, disassembly has much in common with the assembly process, and methods from this work could be extended, see, for example, [Dan+22; HC22]. However, the robotic systems need to have the capability to deal with a much higher degree of uncertainty. It requires a close integration of perception, reasoning, and execution methods and is therefore an important research field from a methodical perspective. For example, the sorting and disassembly of electronic waste as shown in Fig. 20 (right). Providing the automation technologies to handle end-of-life products efficiently is therefore one of the most urgent, and at the same time, most challenging tasks. The idea of mass customization will only be sustainable if this is successful.

Bibliography

[Aja+19] A. Ajay, M. Bauza, J. Wu, N. Fazeli, J. B. Tenenbaum, A. Rodriguez, and L. P. Kaelbling. "Combining Physical Simulators and Object-Based Networks for Control". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 3217–3223. DOI: 10.1109/ICRA.2019.8794358 (cited on page 79).

[AK75] H. Akashi and H. Kumamoto. "Construction of discrete-time nonlinear filter by Monte Carlo methods with variance-reducing techniques". In: *Systems and Control* 19.4 (1975), pp. 211–221 (cited on page 51).

[Alb+07] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger. "The DLR lightweight robot: design and control concepts for robots in human environments". In: *Industrial Robot* 34.5 (2007), pp. 376–385. DOI: 10.1108/01439910710774386 (cited on pages 14, 36, 39).

[AOH07] A. Albu-Schäffer, C. Ott, and G. Hirzinger. "A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots". In: *The International Journal of Robotics Research* 26.1 (2007), pp. 23–39. DOI: 10.1177/0278364907073776 (cited on pages 8, 15, 39).

[AVK16] D. Almeida, F. E. Viña, and Y. Karayiannidis. "Bimanual folding assembly: Switched control and contact point estimation". In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. 2016, pp. 210–216. DOI: 10.1109/HUMANOIDS.2016.7803279 (cited on page 12).

[AS72] D. Alspach and H. Sorenson. "Nonlinear Bayesian estimation using Gaussian sum approximations". In: *IEEE Transactions on Automatic Control* 17.4 (1972), pp. 439–448. DOI: 10.1109/TAC.1972.1100034 (cited on page 51).

[AP75] A. P. Ambler and R. J. Popplestone. "Inferring the positions of bodies from specified spatial relationships". In: *Artificial Intelligence* 6.2 (1975), pp. 157–174. DOI: 10.1016/0004-3702(75)90007-7 (cited on pages 25, 26).

[AM79] B. Anderson and J. B. Moore. *Optimal filtering*. Prentice-Hall, 1979 (cited on page 50).

[AJT16] R. Andre, M. Jokesch, and U. Thomas. "Reliable robot assembly using haptic rendering models in combination with particle filters". In: *2016 IEEE International*

Conference on Automation Science and Engineering (CASE). 2016, pp. 1134–1139. DOI: 10.1109/COASE.2016.7743532 (cited on page 11).

[AD02] C. Andrieu and A. Doucet. “Particle filtering for partially observed Gaussian state space models”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 64.4 (2002), pp. 827–836. DOI: 10.1111/1467-9868.00363 (cited on page 51).

[App21] Apple. “Environmental Progress Report”. In: *Apple Inc. Website* (2021). accessed online: 25.02.2022. URL: https://www.apple.com/environment/pdf/Apple_Environmental_Progress_Report_2021.pdf (cited on page 88).

[Asa90] H. Asada. “Teaching and learning of compliance using neural nets: Representation and generation of nonlinear compliance”. In: *Proceedings., IEEE International Conference on Robotics and Automation*. 1990, pp. 1237–1244. DOI: 10.1109/ROBOT.1990.126167 (cited on page 83).

[BNR21] T. Bachmann, K. Nottensteiner, and M. A. Roa. “Automated Planning of Workcell Layouts Considering Task Sequences”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 12662–12668. DOI: 10.1109/ICRA48506.2021.9561831 (cited on pages 14, 21, 25, 77, 82, 87).

[Bac+20] T. Bachmann, K. Nottensteiner, I. V. Rodríguez Brena, and A. Stemmer. “Using Task-Specific Workspace Maps to Plan and Execute Complex Robotic Tasks in a Flexible Multi-Robot Setup”. In: *ISR 2020; 52nd International Symposium on Robotics*. VDE Verlag, 2020, pp. 161–168. ISBN: 978-3800754281 (cited on pages 14, 21, 32, 41, 67, 77, 87).

[Bad+91] F. Badano, M. Bétemps, T. Redarce, and A. Jutard. “Robotic assembly by slight random movements”. In: *Robotica* 9.1 (1991), pp. 23–29. DOI: 10.1017/S0263574700015538 (cited on page 12).

[Bal+17] S. Balakirsky, C. Schlenoff, S. Rama Fiorini, S. Redfield, M. Barreto, H. Nakawala, J. L. Carbonera, L. Soldatova, J. Bermejo-Alonso, F. Maikore, P. J. S. Goncalves, E. De Momi, V. R. Sampath Kumar, and T. Haidegger. “Towards a robot task ontology standard”. In: *Proceedings of the ASME 2017 International Manufacturing Science and Eng. Conference*. 2017. DOI: 10.1115/MSEC2017-2783 (cited on page 78).

[Bal98] R. S. Ball. *A Treatise on the Theory of Screws*. Cambridge: Cambridge University Press, 1998. ISBN: 978-0521636506 (cited on pages 16, 34, 36, 42, 64). Republication of: *A Treatise on the Theory of Screws*. 1900.

[Bar77] A. J. Barbera. *An Architecture for a Robot Hierarchical Control System*. Special Publication (NIST SP): 500-23. National Institute of Standards and Technology, 1977 (cited on page 24).

[Bas+03] A. Bashi, V. Jilkov, R. Li, and H. Chen. “Distributed Implementations of Particle Filters”. In: *Sixth International Conference of Information Fusion, 2003. Proceedings of the*. Vol. 2. 2003, pp. 1164–1171. DOI: 10.1109/ICIF.2003.177369 (cited on page 80).

[BRS18] W. Bauer, P. Rally, and O. Scholtz. “Schnelle Ermittlung sinnvoller MRK-Anwendungen”. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 113.9 (2018), pp. 554–559. DOI: 10.3139/104.111970 (cited on page 3).

[Bel+22] B. Belousov, B. Wibranek, J. Schneider, T. Schneider, G. Chalvatzaki, J. Peters, and O. Tessmann. “Robotic architectural assembly with tactile skills: Simulation and optimization”. In: *Automation in Construction* 133 (2022). DOI: 10.1016/j.autcon.2021.104006 (cited on page 83).

[Bes+17] A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou. “A stable particle filter for a class of high-dimensional state-space models”. In: *Advances in Applied Probability* 49.1 (2017), pp. 24–48. DOI: 10.1017/apr.2016.77 (cited on page 80).

[BD88] A. Bicchi and P. Dario. “Intrinsic Tactile Sensing for Artificial Hands”. In: *Proceedings of the 4th International Symposium on Robotics Research*. MIT Press, 1988, pp. 83–90 (cited on page 7).

[BSB93] A. Bicchi, J. K. Salisbury, and D. L. Brock. “Contact sensing from force measurements”. In: *The International Journal of Robotics Research* 12.3 (1993), pp. 249–262. DOI: 10.1177/027836499301200304 (cited on page 7).

[BSD89] A. Bicchi, J. K. Salisbury, and P. Dario. “Augmentation of grasp robustness using intrinsic tactile sensing”. In: *Proceedings, 1989 International Conference on Robotics and Automation*. 1989, pp. 302–307. DOI: 10.1109/ROBOT.1989.100005 (cited on page 7).

[BLB08] P. Bickel, B. Li, and T. Bengtsson. “Sharp failure rates for the bootstrap particle filter in high dimensions”. In: *IMS Collections: Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh*. Institute of Mathematical Statistics, 2008, pp. 318–329. DOI: 10.1214/074921708000000228 (cited on page 80).

[Bis+10] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger. “The KUKA-DLR Lightweight Robot arm – a new reference platform for robotics research and manufacturing”. In: *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE Verlag, 2010. ISBN: 978-3800732739 (cited on page 39).

[Bjö+11] A. Björkelund, L. Edström, M. Haage, J. Malec, K. Nilsson, P. Nugues, S. G. Robertz, D. Störkle, A. Blomdell, R. Johansson, M. Linderoth, A. Nilsson, A.

Robertsson, A. Stolt, and H. Bruyninckx. “On the integration of skilled robot motions for productivity in manufacturing”. In: *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*. 2011, pp. 1–9. DOI: 10.1109/ISAM.2011.5942366 (cited on pages 12, 24, 28, 78).

[Bjö+12] A. Björkelund, H. Bruyninckx, J. Malec, K. Nilsson, and P. Nugues. “Knowledge for Intelligent Industrial Robots”. In: *AAAI Spring Symposium Series*. Technical Report SS-12-02. 2012 (cited on page 78).

[BB87] M. Blauer and P. R. Bélanger. “State and Parameter Estimation for Robotic Manipulators Using Force Measurements”. In: *IEEE Transactions on Automatic Control* 32 (12 1987), pp. 1055–1066. DOI: 10.1109/TAC.1987.1104524 (cited on page 11).

[Bøg+12] S. Bøgh, O. S. Nielsen, M. R. Pedersen, V. Krüger, and O. Madsen. “Does your robot have skills?” In: *Proceedings of the 43rd International Symposium on Robotics*. VDE Verlag. 2012 (cited on pages 12, 24, 28, 29).

[BP73] R. Bolles and R. Paul. *The use of sensory feedback in a programmable assembly system*. Tech. rep. STAN-CS-396. Computer Science Department, Stanford University, 1973 (cited on page 12).

[BOS99] V. Boor, M. Overmars, and A. van der Stappen. “The Gaussian sampling strategy for probabilistic roadmap planners”. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation*. Vol. 2. 1999, pp. 1018–1023. DOI: 10.1109/ROBOT.1999.772447 (cited on pages 16, 59).

[Boo82] G. Boothroyd. “Economics of assembly systems”. In: *Journal of Manufacturing Systems* 1.1 (1982), pp. 111–127. DOI: 10.1016/S0278-6125(82)80072-1 (cited on page 3).

[BA92] G. Boothroyd and L. Alting. “Design for Assembly and Disassembly”. In: *CIRP Annals* 41.2 (1992), pp. 625–636. DOI: 10.1016/S0007-8506(07)63249-1 (cited on pages 4, 25).

[BCM82] G. Boothroyd, P. Corrado, and L. E. Murch. *Automatic Assembly*. Marcel Dekker, Inc., 1982. URL: https://archive.org/details/automaticassembl0000boot/ (cited on page 3).

[Bou84] A. Bourjault. “Contribution à une approche méthodologique de l’assemblage automatisé: élaboration automatique des séquences opératoires”. Thése d’Etat. Université de Franche-Comté, 1984 (cited on page 25).

[Bro84] R. W. Brockett. “Robotic Manipulators and the Product of Exponentials Formula”. In: *Mathematical Theory of Networks and Systems*. Springer, 1984, pp. 120–129. DOI: 10.1007/BFb0031048 (cited on page 37).

[Bro82] R. A. Brooks. “Symbolic error analysis and robot planning”. In: *The International Journal of Robotics Research* 1.4 (1982), pp. 29–78. DOI: 10.1177/027836498200100403 (cited on page 56).

[Bru+16] S. G. Brunner, F. Steinmetz, R. Belder, and A. Dömel. “RAFCON: A graphical tool for engineering complex, robotic tasks”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 3283–3290. DOI: 10.1109/IROS.2016.7759506 (cited on page 31).

[BDD93] H. Bruyninckx, J. De Schutter, and S. Dutré. “The "Reciprocity" and "Consistency" Based Approaches to Uncertainty Identification for Compliant Motions”. In: *1993 Proceedings IEEE International Conference on Robotics and Automation*. 1993, pp. 349–354. DOI: 10.1109/ROBOT.1993.292006 (cited on page 42).

[Bru95] H. Bruyninckx. “Kinematic models for robot compliant motion with identification of uncertainties”. PhD thesis. KU Leuven, 1995 (cited on page 42).

[BD96] H. Bruyninckx and J. De Schutter. “Specification of force-controlled actions in the "task frame formalism"-a synthesis”. In: *IEEE Transactions on Robotics and Automation* 12.4 (1996), pp. 581–589. DOI: 10.1109/70.508440 (cited on page 11).

[Bru+95] H. Bruyninckx, S. Demey, S. Dutré, and J. De Schutter. “Kinematic Models for Model-Based Compliant Motion in the Presence of Uncertainty”. In: *The International Journal of Robotics Research* 14.5 (1995), pp. 465–482. DOI: 10.1177/027836499501400505 (cited on pages 42, 56, 64).

[BDD95] H. Bruyninckx, S. Dutré, and J. De Schutter. “Peg-on-Hole: A Model Based Solution to Peg and Hole Alignment”. In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. 1995, pp. 1919–1924. DOI: 10.1109/ROBOT.1995.525545 (cited on page 83).

[BS71] R. S. Bucy and K. D. Senne. “Digital synthesis of non-linear filters”. In: *Automatica* 7.3 (1971), pp. 287–298. DOI: 10.1016/0005-1098(71)90121-X (cited on page 51).

[CCS10] S. Cabras, E. M. Castellanos, and E. Staffetti. “Contact-State Classification in Human-Demonstrated Robot Compliant Motion Tasks Using the Boosting Algorithm”. In: *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* 40.5 (2010), pp. 1372–1386. DOI: 10.1109/TSMCB.2009.2038492 (cited on page 11).

[Cac+99] F. Caccavale, C. Natale, B. Siciliano, and L. Villani. “Six-DOF impedance control based on angle/axis representations”. In: *IEEE Transactions on Robotics and Automation* 15.2 (1999), pp. 289–300. DOI: 10.1109/70.760350 (cited on page 39).

[Cac+09] J. Cachay, H. Gimpel, J. Kaar, G. Rumpel, and J. Sandau. “Made in Germany: Zukunftsperspektiven für die Produktion in Deutschland”. In: (2009) (cited on page 2).

[CGM07] O. Cappé, S. J. Godsill, and E. Moulines. “An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo”. In: *Proceedings of the IEEE* 95.5 (2007), pp. 899–924. DOI: 10.1109/JPROC.2007.893250 (cited on pages 11, 49, 51–53, 58, 80).

[Car21] C. Carey. “The Evolution of the iPhone Every Model from 2007-2020”. In: *iPhone Life Magazine* (2021). accessed online: 7.08.2021. URL: https://www.iphonelife.com/content/evolution-iphone-every-model-2007-2016 (cited on page 2).

[Cha+11] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans. *Fundamentals of object tracking*. Cambridge University Press, 2011 (cited on pages 16, 45, 50–52, 61).

[CRP13] M. Chalon, J. Reinecke, and M. Pfanne. “Online in-hand object localization”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 2977–2984. DOI: 10.1109/IROS.2013.6696778 (cited on pages 7, 11).

[Cha30] M. Chasles. “Note sur les propriétés générales du système de deux corps semblables entr’eux et placés d’une manière quelconque dans l’espace; et sur le déplacement fini ou infiniment petit d’un corps solide libre.” In: *Bulletin des Sciences Mathematiques, Astronomiques, Physiques et Chimiques* 14 (1830), pp. 321–326 (cited on page 34).

[Che+09] H. Chen, W. Eakins, J. Wang, G. Zhang, and T. Fuhlbrigge. “Robotic wheel loading process in automotive manufacturing automation”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 3814–3819. DOI: 10.1109/IROS.2009.5354048 (cited on pages 12, 61).

[Che03] Z. Chen. *Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond*. Tech. rep. McMaster University, Hamilton, Ontario, Canada, 2003 (cited on pages 45, 80).

[CB01] S. R. Chhatpar and M. S. Branicky. “Search strategies for peg-in-hole assemblies with position uncertainty”. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2001, pp. 1465–1470. DOI: 10.1109/IROS.2001.977187 (cited on page 12).

[CB03] S. R. Chhatpar and M. S. Branicky. “Localization for Robotic Assemblies with Position Uncertainty”. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2003, pp. 2534–2540. DOI: 10.1109/IROS.2003.1249251 (cited on page 58).

[CB05] S. R. Chhatpar and M. S. Branicky. “Localization for Robotic Assemblies Using Probing and Particle Filtering”. In: *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2005, pp. 1379–1384. DOI: 10.1109/AIM.2005.1511203 (cited on page 11).

[Chi09] G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups. Volume 1: Classical Results and Geometric Methods*. New York: Birkhäuser Boston, 2009. ISBN: 978-0817648022. DOI: 10.1007/978-0-8176-4803-9 (cited on pages 45, 48, 52).

[Chi+13] M. Chitchian, A. Simonetto, A. S. van Amesfoort, and T. Keviczky. “Distributed Computation Particle Filters on GPU Architectures for Real-Time Control Applications”. In: *IEEE Transactions on Control Systems Technology* 21.6 (2013), pp. 2224–2238. DOI: 10.1109/TCST.2012.2234749 (cited on pages 51, 80).

[CPR85] K. Collins, A. Palmer, and K. Rathmill. “The development of a European benchmark for the comparison of assembly robot programming systems”. In: *Robot Technology and Applications*. Springer, 1985, pp. 187–199. DOI: 10.1007/978-3-662-02440-9_18 (cited on page 10).

[Cra09] J. J. Craig. *Introduction to Robotics, Mechanics and Control*. 3. London: Pearson, 2009. ISBN: 978-8131718360 (cited on pages 36, 41, 57).

[CD02] D. Crisan and A. Doucet. “A survey of convergence results on particle filtering methods for practitioners”. In: *IEEE Transactions on signal processing* 50.3 (2002), pp. 736–746. DOI: 10.1109/78.984773 (cited on pages 51, 80).

[Dah+10] R. Dahiya, G. Metta, M. Valle, and G. Sandini. “Tactile Sensing - From Humans to Humanoids”. In: *IEEE Transactions on Robotics* 26.1 (2010), pp. 1–20. DOI: 10.1109/TRO.2009.2033627 (cited on page 7).

[Dai06] J. S. Dai. “An historical review of the theoretical development of rigid body displacements from Rodrigues parameters to the finite twist”. In: *Mechanism and Machine Theory* 41.1 (2006), pp. 41–52. DOI: 10.1016/j.mechmachtheory.2005.04.004 (cited on page 34).

[Dai21] Daimler. “"Factory 56". Mercedes-Benz Cars increases flexibility and efficiency in operations”. In: *Daimler AG Website* (2021). accessed online: 17.12.2021. URL: https://www.daimler.com/innovation/production/factory-56.html (cited on page 3).

[Dan+22] M. Daneshmand, F. Noroozi, C. Corneanu, F. Mafakheri, and P. Fiorini. “Industry 4.0 and Prospects of Circular Economy: A Survey of Robotic Assembly and Disassembly”. In: *The International Journal of Advanced Manufacturing Technology* (2022). DOI: 10.1007/s00170-021-08389-1 (cited on pages 25, 88).

[DH03] F. Daum and J. Huang. “Curse of dimensionality and particle filters”. In: *2003 IEEE Aerospace Conference Proceedings*. Vol. 4. 2003. DOI: 10.1109/AERO.2003.1235126 (cited on page 80).

[Dav87] S. M. Davis. *Future Perfect*. Addison Wesley, 1987. ISBN: 978-0201115130 (cited on page 2).

[De 16] G. P. L. De Chambrier. “Learning Search Strategies from Human Demonstrations”. PhD thesis. École Polytechnique Fédérale de Lausanne, 2016 (cited on page 83).

[DW87] T. De Fazio and D. Whitney. “Simplified generation of all mechanical assembly sequences”. In: *IEEE Journal on Robotics and Automation* 3.6 (1987), pp. 640–658. DOI: 10.1109/JRA.1987.1087132 (cited on page 25).

[De +99] J. De Schutter, H. Bruyninckx, S. Dutré, J. Katupitiya, S. Demey, and T. Lefebvre. “Estimating First-Order Geometric Parameters and Monitoring Contact Transitions during Force-Controlled Compliant Motion”. In: *The International Journal of Robotics Research* 18.12 (1999), pp. 1161–1184. DOI: 10.1177/02783649922067780 (cited on pages 10, 11, 66).

[De +07] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, Aertbeliën, K. Claes, and H. Bruyninckx. “Constrained-based Task Specification and Estimation for Sensor-Based Robot Systems in the Presence of Geometric Uncertainty”. In: *The International Journal of Robotics Research* 26.5 (2007), pp. 433–455. DOI: 10.1177/0278364907078091O7 (cited on pages 12, 56).

[DDH04] T. J. Debus, P. E. Dupont, and R. D. Howe. “Contact State Estimation Using Multiple Model Estimation and Hidden Markov Models”. In: *The International Journal of Robotics Research* 23.4-5 (2004), pp. 399–413. DOI: 10.1177/0278364904042195 (cited on page 10).

[Del94] P. Del Moral. “Résolution particulaire des problèmes d’estimation et d’optimisation non linéaires”. PhD thesis. Toulouse, 1994 (cited on page 51).

[Del96] P. Del Moral. “Nonlinear Filtering: Interacting Particle Resolution”. In: *Markov Processes and Related Fields* 2.4 (1996), pp. 555–580 (cited on page 51).

[DPV13] P. Del Moral, G. W. Peters, and C. Vergé. “An introduction to stochastic particle integration methods: with applications to risk and insurance”. In: *Monte Carlo and Quasi-Monte Carlo Methods 2012*. Springer, 2013, pp. 39–81. DOI: 10.1007/978-3-642-41095-6_3 (cited on page 51).

[Dem+14] Ö. Demirel, I. Smal, W. J. Niessen, E. Meijering, and I. F. Sbalzarini. “PPF — A parallel particle filtering library”. In: *IET Conference on Data Fusion Target*

Tracking 2014: Algorithms and Applications (DF TT 2014). 2014, pp. 1–8. DOI: 10.1049/cp.2014.0529 (cited on pages 51, 80).

[DV89] R. S. Desai and R. A. Volz. “Identification and Verification of Termination Conditions in Fine Motion in Presence of Sensor Errors and Geometric Uncertainties”. In: *Proceedings, 1989 International Conference on Robotics and Automation*. 1989, pp. 800–807. DOI: 10.1109/ROBOT.1989.100082 (cited on pages 10, 42).

[Des89] R. S. Desai. “On fine motion in mechanical assembly in presence of uncertainty”. PhD thesis. University of Michigan, 1989 (cited on pages 24, 41).

[Don86] B. Donald. “Robot motion planning with uncertainty in the geometric models of the robot and environment: A formal framework for error detection and recovery”. In: *Proceedings. 1986 IEEE International Conference on Robotics and Automation*. Vol. 3. 1986, pp. 1588–1593. DOI: 10.1109/ROBOT.1986.1087480 (cited on pages 12, 67).

[DMB93] K. L. Doty, C. Melchiorri, and C. Bonivento. “A Theory of Generalized Inverses Applied to Robotics”. In: *The International Journal of Robotics Research* 12.1 (1993). DOI: 10.1177/027836499301200101 (cited on pages 62, 65).

[DFG01] A. Doucet, N. de Freitas, and N. Gordon. “An Introduction to Sequential Monte Carlo Methods”. In: *Sequential Monte Carlo Methods in Practice*. Ed. by A. Doucet, N. de Freitas, and N. Gordon. Statistics for Engineering and Information Science. Springer, New York, NY, 2001. DOI: 10.1007/978-1-4757-3437-9_1 (cited on pages 11, 48, 49, 51–54).

[DGA00] A. Doucet, S. Godsill, and C. Andrieu. “On sequential Monte Carlo sampling methods for Bayesian filtering”. In: *Statistics and Computing* 10.3 (2000), pp. 197–208. DOI: 10.1023/A:1008935410038 (cited on page 51).

[DJ11] A. Doucet and A. M. Johansen. “A tutorial on particle filtering and smoothing: fifteen years later”. In: *The Oxford Handbook of Nonlinear Filtering*. Ed. by D. Crisan and B. Rozovskii. Oxford University Press, 2011, pp. 656–704. ISBN: 978-0-19-953290-2 (cited on pages 11, 15, 49–52, 55).

[Dra78] S. H. Drake. “Using Compliance in Lieu of Sensory Feedback for Automatic Assembly”. PhD thesis. Department of Mechanical Engineering, Massachusetts Institute of Technology, 1978 (cited on pages 8, 38).

[Duf+11] D. J. Duff, T. Mörwald, R. Stolkin, and J. Wyatt. “Physical simulation for monocular 3D model based tracking”. In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 5218–5225. DOI: 10.1109/ICRA.2011.5980535 (cited on page 62).

[Dut98] S. Dutré. “Identification and monitoring of contacts in force controlled robotic manipulation”. PhD thesis. Katholieke Universiteit Leuven, 1998 (cited on pages 55, 56).

[Eck87] R. Eckhardt. “Stan Ulam, John von Neumann and the Monte Carlo method”. In: *Los Alamos Science* 15 (1987), pp. 131–141 (cited on page 51).

[Eji+72] M. Ejiri, T. Uno, H. Yoda, T. Goto, and K. Takeyasu. “A Prototype Intelligent Robot that Assembles Objects from Plan Drawings”. In: *IEEE Transactions on Computers* C-21.2 (1972), pp. 161–170. DOI: 10.1109/TC.1972.5008921 (cited on pages 9, 24, 25).

[ETM21] J. Elfring, E. Torta, and R. van de Molengraft. “Particle Filters: A Hands-On Tutorial”. In: *Sensors* 21.2 (2021), p. 438. DOI: 10.3390/s21020438 (cited on pages 51–53, 80).

[ElM06] H. A. ElMaraghy. “Flexible and reconfigurable manufacturing systems paradigms”. In: *International Journal of Flexible Manufacturing Systems* 17.4 (2006), pp. 261–276. DOI: 10.1007/s10696-006-9028-7 (cited on pages 4, 78).

[Erd86] M. Erdmann. “Using backprojections for fine motion planning with uncertainty”. In: *The International Journal of Robotics Research* 5.1 (1986), pp. 19–45. DOI: 10.1177/027836498600500102 (cited on pages 12, 67).

[EGO94] A. Ericsson, P. Gröndahl, and M. Onori. “An Application with the MARK III Assembly System”. In: *Proceedings of the 25th International Symposium on Robotics (ISR)*. International Ferderation of Robotics (IFR). 1994 (cited on page 4).

[Eri98] G. Erixon. “Modular function deployment”. PhD thesis. KTH, Production Systems, 1998 (cited on page 2).

[Esk01] S. Eskilander. “Design for automatic assembly”. PhD thesis. KTH, Production Engineering, 2001 (cited on page 4).

[ELO98] S. Eskilander, B. Langbeck, and M. Onori. “Industrial Application Studies of a New Standard FAA Machine”. In: *IFAC Proceedings Volumes* 31.7 (1998), pp. 171–176. DOI: 10.1016/S1474-6670(17)40276-X (cited on page 4).

[Fah74] S. E. Fahlman. “A planning system for robot construction tasks”. In: *Artificial intelligence* 5.1 (1974), pp. 1–49. DOI: 10.1016/0004-3702(74)90008-3 (cited on page 25).

[FL16] P. Flores and H. M. Lankarani. *Contact force models for multibody dynamics*. Vol. 226. Springer, 2016. DOI: 10.1007/978-3-319-30897-5 (cited on page 40).

[For23] H. Ford. *My Life and Work*. Doubleday, Page & Company, 1923 (cited on page 2).

[Fox03] D. Fox. “Adapting the Sample Size in Particle Filters Through KLD-Sampling”. In: *The International Journal of Robotics Research* 22.12 (2003), pp. 985–1003. DOI: 10.1177/0278364903022012001 (cited on page 80).

[FG96] S. Franklin and A. Graesser. “Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents”. In: *Intelligent Agents III Agent Theories, Architectures, and Languages*. Ed. by J. Müller, M. Wooldridge, and N. Jennings. Vol. 1193. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence). Springer, 1996, pp. 21–35. DOI: 10.1007/BFb0013570 (cited on page 78).

[GLB05] K. Gadeyne, T. Lefebvre, and H. Bruyninckx. “Bayesian Hybrid Model-State Estimation Applied to Simultaneous Contact Formation Recognition and Geometrical Parameter Estimation”. In: *The International Journal of Robotics Research* 24.8 (2005), pp. 615–630. DOI: 10.1177/0278364905056196 (cited on pages 11, 51).

[Gei+06] T. Geier, M. Förg, R. Zander, H. Ulbrich, F. Pfeiffer, A. Brandsma, and A. van der Velde. “Simulation of a Push Belt CVT Considering Uni- and Bilateral Constraints”. In: *Journal of Applied Mathematics and Mechanics* 86 (10 2006), pp. 795–806. DOI: 10.1002/zamm.200610287 (cited on page 41).

[GS02] G. Gilardi and I. Sharf. “Literature Survey of Contact Dynamics Modelling”. In: *Mechanism and Machine Theory* 37.10 (2002), pp. 1213–1239. DOI: 10.1016/S0094-114X(02)00045-9 (cited on pages 40, 43).

[GT56] A. J. Goldmann and A. W. Tucker. “Polyhedral Convex Cones”. In: *Linear Inequalities and Related Systems, Annals of Mathematics Studies*. Ed. by H. W. Kuhn and A. W. Tucker. Vol. 38. Princeton University Press, 1956, pp. 19–40. ISBN: 978-1400881987 (cited on page 10).

[GSS93] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. “Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation”. In: *Radar and Signal Processing, IEE Proceedings F* 140.2 (1993), pp. 107–113 (cited on pages 51, 53, 54, 58).

[GTI80] T. Goto, K. Takeyasu, and T. Inoyama. “Control Algorithm for Precision Insert Operation Robots”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 1.10 (1980), pp. 19–25. DOI: 10.1109/TSMC.1980.4308347 (cited on page 8).

[GRL94] S. Gottschlich, C. Ramos, and D. Lyons. “Assembly and task planning: a taxonomy”. In: *IEEE Robotics and Automation Magazine* 1.3 (1994), pp. 4–12. DOI: 10.1109/100.326723 (cited on page 25).

[GM18] G. Graetz and G. Michaels. “Robots at work”. In: *The Review of Economics and Statistics* 100.5 (2018), pp. 753–768. DOI: 10.1162/rest_a_00754 (cited on page 3).

[GKR74] K. Grove, H. Karcher, and E. A. Ruh. “Jacobi fields and Finsler metrics on compact Lie groups with an application to differentiable pinching problems”. In: *Mathematische Annalen* 211.1 (1974), pp. 7–21. DOI: 10.1007/BF01344138 (cited on page 52).

[GOS09] N. Guarino, D. Oberle, and S. Staab. “What is an ontology?” In: *Handbook on Ontologies*. Ed. by S. Staab and R. Studer. International Handbooks on Information Systems. Springer, 2009, pp. 1–17. DOI: 10.1007/978-3-540-92673-3_0 (cited on page 78).

[GGB92] V. Gullapalli, R. A. Grupen, and A. G. Barto. “Learning reactive admittance control”. In: *Proceedings 1992 IEEE International Conference on Robotics and Automation*. 1992, pp. 1475–1480. DOI: 10.1109/ROBOT.1992.220143 (cited on page 83).

[Had+08] S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger. “Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction”. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008, pp. 3356–3363. DOI: 10.1109/IROS.2008.4650764 (cited on page 65).

[HLW00] D. Halperin, J.-C. Latombe, and R. H. Wilson. “A General Framework for Assembly Planning: The Motion Space Approach”. In: *Algorithmica* 26 (2000), pp. 577–601. DOI: 10.1007/s004539910025 (cited on pages 25, 26).

[HM69] J. E. Handschin and D. Q. Mayne. “Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering”. In: *International Journal of Control* 9.5 (1969), pp. 547–559. DOI: 10.1080/00207176908905777 (cited on page 51).

[Har+13] R. Hartley, J. Trumpf, Y. Dai, and H. Li. “Rotation Averaging”. In: *International Journal of Computer Vision* 103.3 (2013), pp. 267–305. DOI: 10.1007/s11263-012-0601-0 (cited on page 52).

[HST92] T. Hasegawa, T. Suehiro, and K. Takase. “A model-based manipulation system with skill-based execution”. In: *Robotics and Automation, IEEE Transactions on* 8.5 (1992), pp. 535–544. DOI: 10.1109/70.163779 (cited on pages 12, 24, 28).

[HKG10] G. Hendeby, R. Karlsson, and F. Gustafsson. “Particle Filtering: The Need for Speed”. In: *EURASIP Journal on Advances in Signal processing* 2010 (2010), pp. 1–9. DOI: 10.1155/2010/181403 (cited on pages 51, 80).

[Her+12] K. Hertkorn, M. A. Roa, C. Preusche, C. Borst, and G. Hirzinger. “Identification of Contact Formations: Resolving Ambiguous Force Torque Information”. In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 3278–3284. DOI: 10.1109/ICRA.2012.6225148 (cited on page 11).

[Hew70] C. Hewitt. *PLANNER: A language for manipulating models and proving theorems in a robot*. Artificial Intelligence Memo AIM-168. Artifical Intelligence Laborartory, Massachusetts Institute of Technology, 1970 (cited on page 25).

[HI92] S. Hirai and K. Iwata. “Recognition of Contact State Based on Geometric Model”. In: *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*. 1992, pp. 1507–1512. DOI: 10.1109/ROBOT.1992.220038 (cited on page 10).

[Hit88] H. Hitakawa. “Advanced parts orientation system has wide application”. In: *Assembly Automation* 8.3 (1988), pp. 147–150. DOI: 10.1108/eb004713 (cited on pages 4, 7).

[HC22] S. Hjorth and D. Chrysostomou. “Human–robot collaboration in industrial environments: A literature review on non-destructive disassembly”. In: *Robotics and Computer-Integrated Manufacturing* 73 (2022), pp. 1–18. DOI: 10.1016/j.rcim.2021.102208 (cited on page 88).

[HL64] Y. C. Ho and R. C. K. Lee. “A Bayesian Approach to Problems in Stochastic Estimation and Control”. In: *IEEE Transactions on Automatic Control* 9 (4 1964), pp. 333–339. DOI: 10.1109/TAC.1964.1105763 (cited on pages 50, 53).

[Hod+18] J. Hodapp, M. Andulkar, T. Reichling, and U. Berger. “Improvements in Robot Teaching for Handling Operations in Production Environments”. In: *ISR 2018; 50th International Symposium on Robotics*. 2018 (cited on page 82).

[Hog84] N. Hogan. “Impedance Control: An Approach to Manipulation”. In: *1984 American Control Conference*. 1984, pp. 304–313. DOI: 10.23919/ACC.1984.4788393 (cited on pages 8, 38, 39).

[HSG06] J. Hol, T. Schön, and F. Gustafsson. “On Resampling Algorithms for Particle Filters”. In: *2006 IEEE Nonlinear Statistical Signal Processing Workshop*. 2006, pp. 79–82. DOI: 10.1109/NSSPW.2006.4378824 (cited on pages 52, 54, 55).

[HS89] L. Homem de Mello and A. Sanderson. “A correct and complete algorithm for the generation of mechanical assembly sequences”. In: *Proceedings, 1989 International Conference on Robotics and Automation*. 1989, pp. 56–61. DOI: 10.1109/ROBOT.1989.99967 (cited on page 25).

[HS90] L. Homem de Mello and A. Sanderson. “AND/OR graph representation of assembly plans”. In: *IEEE Transactions on Robotics and Automation* 6 (2 1990), pp. 188–199. DOI: 10.1109/70.54734 (cited on page 26).

[HR91] A. Hörmann and U. Rembold. “Development of an advanced robot for autonomous assembly”. In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. 1991, pp. 2452–2457. DOI: 10.1109/ROBOT.1991.131992 (cited on page 9).

[Hun78] K. H. Hunt. *Geometry of Mechanisms*. Oxford: Clarendon Press, 1978. ISBN: 978-0198561248 (cited on page 34).

[Hun03] K. H. Hunt. “Review: Don’t Cross-Thread the Screw”. In: *Journal of Robotic Systems* 20.7 (2003), pp. 317–339. DOI: 10.1002/rob.10095 (cited on pages 35, 36).

[Hus+97] M. Husty, A. Karger, H. Sachs, and W. Steinhilper. *Kinematik und Robotik*. Springer, 1997. ISBN: 978-3642638220 (cited on page 34).

[Huy09] D. Q. Huynh. “Metrics for 3D rotations: Comparison and analysis”. In: *Journal of Mathematical Imaging and Vision* 35.2 (2009), pp. 155–164. DOI: 10.1007/s10851-009-0161-2 (cited on page 68).

[IEE15] IEEE Std 1872™-2015. *IEEE Standard Ontologies for Robotics and Automation*. Standard. IEEE Standards Association, 2015. DOI: 10.1109/IEEESTD.2015.7084073 (cited on pages 14, 78).

[Ino74] H. Inoue. *Force feedback in precise assembly tasks*. Artificial Intelligence Memo AIM-308. Artificial Intelligence Labratory, Massachusetts Institute of Technology, 1974 (cited on pages 8, 11, 83).

[Ino+17] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana. “Deep reinforcement learning for high precision assembly tasks”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 819–825. DOI: 10.1109/IROS.2017.8202244 (cited on page 83).

[IB96] M. Isard and A. Blake. “Contour tracking by stochastic propagation of conditional density”. In: *Computer Vision — ECCV ’96*. Ed. by B. Buxton and R. Cipolla. Lecture Notes in Computer Science 1064. Berlin / Heidelberg: Springer, 1996, pp. 343–356. ISBN: 978-3540611226 (cited on pages 50, 51).

[ISO12] ISO 8373:2012(E/F). *Robots and robotic devices – Vocabulary*. Standard. International Organization for Standardization, 2012 (cited on pages 9, 38).

[IX00] K. Ito and K. Xiong. “Gaussian filters for nonlinear filtering problems”. In: *IEEE Transactions on Automatic Control* 45.5 (2000), pp. 910–927. DOI: 10.1109/9.855552 (cited on page 50).

[JPH11] Z. Jakovljevic, P. B. Petrovic, and J. Hodolic. “Contact States Recognition in Robotic Part Mating Based on Support Vector Machines”. In: *The International Journal of Advanced Manufactoring Technology* 59 (2011), pp. 377–395. DOI: 10.1007/s00170-011-3501-5 (cited on page 11).

[Jas20] R. Jaster. *Agents'Abilities*. Berlin, Boston: De Gruyter, 2020. DOI: 10.1515/9783110650464-001 (cited on page 28).

[Jaz70] A. H. Jazwinski. *Stochastic processes and filtering theory*. Academic Press, 1970 (cited on pages 45, 47, 48, 50).

[Jim11] P. Jiménez. “Survey on assembly sequencing: a combinatorial and geometric perspective”. In: *Journal of Intelligent Manufacturing* 24 (2011), pp. 235–250. DOI: 10.1007/s10845-011-0578-5 (cited on page 25).

[JGH19] L. Johannsmeier, M. Gerchow, and S. Haddadin. “A Framework for Robot Manipulation: Skill Formalism, Meta Learning and Adaptive Control”. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 5844–5850. DOI: 10.1109/ICRA.2019.8793542 (cited on pages 82, 83).

[Joh09] A. M. Johansen. “SMCTC: Sequential Monte Carlo in C++”. In: *Journal of Statistical Software* 30.6 (2009), pp. 1–41. DOI: 10.18637/jss.v030.i06 (cited on page 51).

[Joh02] R. Johansson. “Implementation of Flexible Automatic Assembly in Small Companies – Flexibility and Process demands”. PhD thesis. KTH Stockholm, 2002 (cited on pages 4, 9).

[Jok+16] M. Jokesch, J. Suchý, A. Winkler, A. Fross, and U. Thomas. “Generic Algorithm for Peg-In-Hole Assembly Tasks for Pin Alignments with Impedance Controlled Robots”. In: *Robot 2015: Second Iberian Robotics Conference*. 2016, pp. 105–117. DOI: 10.1007/978-3-319-27149-1_9 (cited on page 12).

[JS09] K. Jorabchi and K. Suresh. “Nonlinear Algebraic Reduction for Snap-Fit Simulation”. In: *Journal of Mechanical Design* 131.6 (2009). DOI: 10.1115/1.3116342 (cited on page 79).

[Jör+00] S. Jörg, J. Langwald, J. Stelter, G. Hirzinger, and C. Natale. “Flexible Robot-Assembly Using a Multi-Sensory Approach”. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation*. 2000, pp. 3687–3694. DOI: 10.1109/ROBOT.2000.845306 (cited on pages 12, 61).

[JU04] S. Julier and J. Uhlmann. “Unscented filtering and nonlinear estimation”. In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422. DOI: 10.1109/JPROC.2003.823141 (cited on page 51).

[KL13] L. P. Kaelbling and T. Lozano-Pérez. “Integrated task and motion planning in belief space”. In: *The International Journal of Robotics Research* 32.9-10 (2013), pp. 1194–1227. DOI: 10.1177/0278364913484072 (cited on page 67).

[Kal60] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Journal of Basic Engineering* 82.1 (1960), pp. 35–45. DOI: 10.1115/1.3662552 (cited on page 50).

[KKR95] K. Kanazawa, D. Koller, and S. Russel. “Stochastic Simulation Algorithms for Dynamic Probalistic Networks”. In: *Proceedings of the Eleventh Conference on Uncertainty in Artifical Intelligence (UAI-95)*. 1995, pp. 346–351 (cited on page 51).

[KLB16] I. Kao, K. Lynch, and J. W. Burdick. “Contact Modeling and Manipulation”. In: *Springer Handbook of Robotics*. Ed. by B. Siciliano and O. Khatib. Springer, 2016, pp. 931–954. DOI: 10.1007/978-3-319-32552-1 (cited on page 41).

[Kar77] H. Karcher. “Riemannian center of mass and mollifier smoothing”. In: *Communications on pure and applied mathematics* 30.5 (1977), pp. 509–541. DOI: 10.1002/cpa.3160300502 (cited on page 52).

[Kas+19] B. Kast, S. Albrecht, W. Feiten, and J. Zhang. “Bridging the Gap Between Semantics and Control for Industry 4.0 and Autonomous Production”. In: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. 2019, pp. 780–787. DOI: 10.1109/COASE.2019.8843174 (cited on page 10).

[Kau+96] S. Kaufman, R. Wilson, R. Jones, T. Calton, and A. Ames. “The Archimedes 2 mechanical assembly planning system”. In: *Proceedings of IEEE International Conference on Robotics and Automation*. Vol. 4. 1996, pp. 3361–3368. DOI: 10.1109/ROBOT.1996.509225 (cited on page 25).

[Kav+96] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. “Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces”. In: *IEEE Transactions on Robotics and Automation* 12.4 (1996), pp. 566–580. DOI: 10.1109/70.508439 (cited on pages 15, 59).

[Ken90] W. S. Kendall. “Probability, convexity, and harmonic maps with small image I: uniqueness and fine existence”. In: *Proceedings of the London Mathematical Society* 3.2 (1990), pp. 371–406. DOI: 10.1112/plms/s3-61.2.371 (cited on page 52).

[Kit87] G. Kitagawa. “Non-Gaussian state—space modeling of nonstationary time series”. In: *Journal of the American Statistical Association* 82.400 (1987), pp. 1032–1041. DOI: 10.1080/01621459.1987.10478534 (cited on pages 49–51).

[Kit96] G. Kitagawa. “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models”. In: *Journal of Computational and Graphical Statistics* 5.1 (1996), pp. 1–25. DOI: 10.2307/1390750 (cited on page 51).

[Kit98] G. Kitagawa. “A Self-Organizing State-Space Model”. In: *Journal of the American Statistical Association* 93.443 (1998), pp. 1203–1215. DOI: 10.2307/2669862 (cited on page 58).

[Kle+20] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber. “A survey on learning-based robotic grasping”. In: *Current Robotics Reports* 1.4 (2020), pp. 1–11. DOI: 10.1007/s43154-020-00021-6 (cited on page 7).

[Kor+99] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel. “Reconfigurable manufacturing systems”. In: *CIRP annals* 48.2 (1999), pp. 527–540. DOI: 10.1016/S0007-8506(07)63232-6 (cited on pages 4, 78).

[KC02] D. Kragic and H. I. Christensen. *Survey on visual servoing for manipulation*. Tech. rep. ISRN KTH/NA/P-02/01-SE. Computational Vision and Active Perception Laboratory, KTH Stockholm, 2002 (cited on page 67).

[Kra+17] A. Kramberger, A. Gams, B. Nemec, D. Chrysostomou, O. Madsen, and A. Ude. “Generalization of orientation trajectories and force-torque profiles for robotic assembly”. In: *Robotics and Autonomous Systems* 98 (2017), pp. 333–346. DOI: 10.1016/j.robot.2017.09.019 (cited on page 83).

[Kui+17] B. Kuipers, E. A. Feigenbaum, P. E. Hart, and N. J. Nilsson. “Shakey: From Conception to History”. In: *AI Magazine* 38.1 (2017), pp. 88–103. DOI: 10.1609/aimag.v38i1.2716 (cited on page 9).

[KUK13] KUKA AG. “Premiere for the LBR iiwa”. In: *KUKA AG News* (Mar. 25, 2013). accessed online: 28.12.2021. URL: https://www.kuka.com/en-de/company/press/news/2013/03/premiere-fuer-den-lbr-iiwa (cited on page 39).

[KUK20] KUKA AG. “More efficiency through Assembly in Motion”. In: *KUKA AG Case Studies* (2020). accessed online: 10.04.2022. URL: https://www.kuka.com/en-de/industries/solutions-database/2020/04/assembly-in-motion (cited on page 60).

[Lan98] B. Langbeck. “Ett standardiserat monteringssystem för flexibel automatisk montering”. PhD thesis. KTH, Production Systems, 1998 (cited on page 4).

[LSH10] F. Lange, J. Scharrer, and G. Hirzinger. “Classification and prediction for accurate sensor-based assembly to moving objects”. In: *2010 IEEE International Conference on Robotics and Automation*. 2010, pp. 2163–2168. DOI: 10.1109/ROBOT.2010.5509659 (cited on pages 12, 61).

[Lee+19] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg. “Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks”. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 8943–8950. DOI: 10.1109/ICRA.2019.8793485 (cited on page 83).

[LBD05] T. Lefebvre, H. Bruyninckx, and J. De Schutter. “Polyhedral Contact Formation Identification for Autonomous Compliant Motion: Exact Nonlinear Bayesian Filtering”. In: *IEEE Transactions on Robotics* 21.1 (2005), pp. 124–129. DOI: 10.1109/TRO.2004.833804 (cited on page 10).

[LBD03] T. Lefebvre, H. Bruyninckx, and J. De Schutter. “Polyhedral Contact Formation Modeling and Identification for Autonomous Compliant Motion”. In: *IEEE Transactions on Robotics and Automation* 19.1 (2003), pp. 26–41. DOI: 10.1109/TRA.2002.805677 (cited on page 66).

[Lef+05] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. De Gersem. “Active Compliant Motion: A Survey”. In: *Advanced Robotics* 19.5 (2005), pp. 479–499. DOI: 10.1163/156855305323383767 (cited on page 10).

[LSW16] C. Lehmann, P. J. Städter, and K. Wächter. “A variant-flexible assembly cell for hydraulic valve sections using a sensitive lightweight robot”. In: *Proceedings of ISR 2016: 47st International Symposium on Robotics*. 2016 (cited on page 10).

[LA17] P. Lehner and A. Albu-Schäffer. “Repetition sampling for efficiently planning similar constrained manipulation tasks”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 2851–2856. DOI: 10.1109/IROS.2017.8206116 (cited on page 32).

[LBH12] D. Leidner, C. Borst, and G. Hirzinger. “Things Are Made for What They Are: Solving Manipulation Tasks by Using Functional Object Classes”. In: *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. 2012, pp. 429–435. DOI: 10.1109/HUMANOIDS.2012.6651555 (cited on pages 29, 31).

[Let+20] P. Letier, T. Siedel, M. Deremetz, E. Pavlovskis, B. Lietaer, K. Nottensteiner, M. A. Roa, J. Casarrubios, J. Romero, and J. Gancet. “HOTDOCK: Design and Validation of a New Generation of Standard Robotic Interface for On-Orbit Servicing”. In: *71st International Astronautical Congress (IAC)*. 2020 (cited on page 22).

[Li14] H. Li. *A Brief Tutorial On Recursive Estimation With Examples From Intelligent Vehicle Applications (Part II): System Models*. Tech. rep. hal-01054646. 2014 (cited on page 61).

[LLT15] S. Li, S. Lyu, and J. Trinkle. "State estimation for dynamic systems with intermittent contact". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 3709–3715. DOI: 10.1109/ICRA.2015.7139714 (cited on pages 11, 62).

[LLT18] S. Li, S. Lyu, and J. Trinkle. "Efficient State Estimation with Constrained Rao-Blackwellized Particle Filter". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 6682–6689. DOI: 10.1109/IROS.2018.8594268 (cited on page 62).

[LJ03] X. R. Li and V. P. Jilkov. "Survey of maneuvering target tracking. Part I. Dynamic models". In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4 (2003), pp. 1333–1364. DOI: 10.1109/TAES.2003.1261132 (cited on page 61).

[LW77] L. I. Lieberman and M. A. Wesley. "AUTOPASS: An automatic programming system for computer controlled mechanical assembly". In: *IBM Journal of Research and Development* 21.4 (1977), pp. 321–333. DOI: 10.1147/rd.214.0321 (cited on page 25).

[Lin+17] F. Lindsten, A. M. Johansen, C. A. Naesseth, B. Kirkpatrick, T. B. Schön, J. Aston, and A. Bouchard-Côté. "Divide-and-conquer with sequential Monte Carlo". In: *Journal of Computational and Graphical Statistics* 26.2 (2017), pp. 445–458. DOI: 10.1080/10618600.2016.1237363 (cited on pages 51, 80).

[LD85] H. Lipkin and J. Duffy. "The Elliptic Polarity of Screws". In: *Journal of Mechanisms, Transmissions, and Automation in Design* 107 (1985), pp. 377–387. DOI: 10.1115/1.3260725 (cited on page 35).

[Lip85] H. Lipkin. "Geometry and Mappings of Screws with Applications to the Hybrid Control of Robotic Manipulators". PhD thesis. University of Florida, 1985 (cited on pages 34, 36).

[LD82] H. Lipkin and J. Duffy. "Analysis of industrial robots via the theory of screws". In: *Proceedings of the 12th International Symposium of Industrial Robots* (1982), pp. 359–370 (cited on pages 36, 38, 42).

[LW93] T. Lozano-Pérez and R. H. Wilson. "Assembly sequencing for arbitrary motions". In: *Proceedings IEEE International Conference on Robotics and Automation*. 1993, pp. 527–532. DOI: 10.1109/ROBOT.1993.291904 (cited on page 25).

[Loz76] T. Lozano-Pérez. "The Design of a Mechanical Assembly System". PhD thesis. Massachusetts Institute of Technology, 1976 (cited on page 25).

[Loz81] T. Lozano-Pérez. “Automatic Planning of Manipulator Transfer Movements”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 11.10 (1981), pp. 681–698. DOI: 10.1109/TSMC.1981.4308589 (cited on page 41).

[Loz83] T. Lozano-Pérez. “Spatial Planning: A Configuration Space Approach”. In: *IEEE Transactions on Computers* C-32.2 (1983), pp. 108–120. DOI: 10.1109/TC.1983.1676196 (cited on page 41).

[LMT84] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor. “Automatic synthesis of fine-motion strategies for robots”. In: *The International Journal of Robotics Research* 3.1 (1984), pp. 3–24. DOI: 10.1177/027836498400300101 (cited on pages 12, 67).

[LR94] T. Lueth and U. Rembold. “Extensive Manipulation Capabilities and Reliable Behavior at Autonomous Robot Assembly”. In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. 1994, pp. 3495–3500. DOI: 10.1109/ROBOT.1994.351033 (cited on page 9).

[Luo+18] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino. “Deep Reinforcement Learning for Robotic Assembly of Mixed Deformable and Rigid Objects”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 2062–2069. DOI: 10.1109/IROS.2018.8594353 (cited on page 83).

[MDR21] D. Ma, S. Dong, and A. Rodriguez. “Extrinsic Contact Sensing with Relative-Motion Tracking from Distributed Tactile Measurements”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 11262–11268. DOI: 10.1109/ICRA48506.2021.9561781 (cited on page 7).

[Mah03] R. Mahler. “Multitarget Bayes filtering via first-order multitarget moments”. In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4 (2003), pp. 1152–1178. DOI: 10.1109/TAES.2003.1261119 (cited on page 51).

[Man04] J. H. Manton. “A globally convergent numerical algorithm for computing the centre of mass on compact Lie groups”. In: *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.* 2004, pp. 2211–2216. DOI: 10.1109/ICARCV.2004.1469774 (cited on page 52).

[Mar+07] F. Markley, Y. Cheng, J. Crassidis, and Y. Oshman. “Averaging Quaternions”. In: *Journal of Guidance, Control, and Dynamics* 30.4 (2007), pp. 1193–1196. DOI: 10.2514/1.28949 (cited on page 52).

[Mar+21] J. Martínez-Moritz, I. Rodríguez, K. Nottensteiner, J.-P. Lutze, P. Lehner, and M. A. Roa. “Hybrid Planning System for In-Space Robotic Assembly of Telescopes using Segmented Mirror Tiles”. In: *2021 IEEE Aerospace Conference*. 2021, pp. 1–16. DOI: 10.1109/AERO50100.2021.9438399 (cited on pages 22, 87).

[MBF18] J. Marvel, R. Bostelman, and J. Falco. “Multi-robot assembly strategies and metrics”. In: *ACM Computing Surveys (CSUR)* 51.1 (2018), pp. 1–32. DOI: 10.1145/3150225 (cited on page 12).

[MF12] J. Marvel and J. Falco. *Best practices and performance metrics using force control for robotic assembly*. NIST Interagency/Internal Report 7901. 2012. DOI: 10.6028/NIST.IR.7901 (cited on pages 8, 38).

[Mas81] M. T. Mason. “Compliance and Force Control for Computer Controlled Manipulators”. In: *IEEE Transactions on System, Man, and Cybernetics* 11.6 (1981), pp. 418–432. DOI: 10.1109/TSMC.1981.4308708 (cited on pages 8, 11).

[Mas01] M. T. Mason. *Mechanics of Robotic Manipulation*. Cambridge, Massachusetts: The MIT Press, 2001. ISBN: 978-0262133968 (cited on page 41).

[ML93] M. T. Mason and K. M. Lynch. “Dynamic Manipulation”. In: *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS ’93)*. 1993, pp. 152–159. DOI: 10.1109/IROS.1993.583093 (cited on page 33).

[May77] G. S. May. *R. E. Olds: Auto Industry Pioneer*. William B. Eerdmans Publishing Companys, 1977. ISBN: 978-0802870285 (cited on page 2).

[May+11] M. P. Mayer, C. M. Schlick, D. Ewert, D. Behnen, S. Kuz, B. Odenthal, and B. Kausch. “Automation of robotic assembly processes on the basis of an architecture of human cognition”. In: *Production Engineering* 5.4 (2011), pp. 423–431. DOI: 10.1007/s11740-011-0316-z (cited on page 10).

[MA93] B. J. McCarragher and H. Asada. “Qualitative Template Matching Using Dynamic Process Models State Transition Recognition of Robotic Assembly”. In: *Journal of Dynamic Systems, Measurement, and Control* 115 (1993), pp. 261–269. DOI: 10.1115/1.2899030 (cited on page 10).

[MPT99] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. “Six degree-of-freedom haptic rendering using voxel sampling”. In: *SIGGRAPH ’99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 1999, pp. 401–408. DOI: 10.1145/311535.311600 (cited on page 43).

[MPT05] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. “Six degree-of-freedom haptic rendering using voxel sampling”. In: *SIGGRAPH ’05: ACM SIGGRAPH 2005 Courses*. 2005. DOI: 10.1145/1198555.1198605 (cited on page 16).

[Mee+07] W. Meeussen, J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. De Schutter. “Contact-State Segmentation Using Particle Filters for Programming by Human Demonstration in Compliant-Motion Tasks”. In: *IEEE Transactions on Robotics* 23.2 (2007), pp. 218–231. DOI: 10.1109/TRO.2007.892227 (cited on pages 11, 43, 64, 81).

[Mee+08] W. Meeussen, E. Staffetti, H. Bruyninckx, J. Xiao, and J. De Schutter. “Integration of Planning and Execution in Force Controlled Compliant Motion”. In: *Robotics and Autonomous Systems* 56.5 (2008), pp. 437–450. DOI: 10.1016/j.robot.2007.09.009 (cited on pages 11, 12, 42).

[Mee+04] W. Meeussen, J. Xiao, J. De Schutter, H. Bruyninckx, and E. Staffetti. “Automatic Verification of Contact States Taking into Account Manipulator Constraints”. In: *IEEE International Conference on Robotics and Automation*. 2004, pp. 3583–3588. DOI: 10.1109/ROBOT.2004.1308808 (cited on page 11).

[MU49] N. Metropolis and S. Ulam. “The Monte Carlo Method”. In: *Journal of the American Statistical Association* 44.247 (1949), pp. 335–341. DOI: 10.1080/01621459.1949.10483310 (cited on page 51).

[MF94] N. Mimura and Y. Funahashi. “Parameter Identification of Contact Conditions by Active Force Sensing”. In: *Proceedings of the 1994 IEEE Intemational Conference on Robotics and Automation*. 1994, pp. 2645–2650. DOI: 10.1109/ROBOT.1994.351115 (cited on page 10).

[Moa02] M. Moakher. “Means and averaging in the group of rotations”. In: *SIAM journal on matrix analysis and applications* 24.1 (2002), pp. 1–16. DOI: 10.1137/S0895479801383877 (cited on page 52).

[MW01] H. Mosemann and F. Wahl. “Automatic decomposition of planned assembly sequences into skill primitives”. In: *IEEE Transactions on Robotics and Automation* 17.5 (Oct. 2001), pp. 709–718. DOI: 10.1109/70.964670 (cited on page 26).

[MA15] M. Muro and S. Andes. “Robots Seem to Be Improving Productivity, Not Costing Jobs”. In: *Harvard Business Review* (2015). accessed online: 6.08.2021. URL: https://hbr.org/2015/06/robots-seem-to-be-improving-productivity-not-costing-jobs (cited on page 3).

[MLS94] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, Florida, USA: CRC Press, 1994. ISBN: 978-0849379819 (cited on pages 34, 35, 37, 38, 41).

[NLS19] C. A. Naesseth, F. Lindsten, and T. B. Schön. “High-dimensional filtering using nested sequential Monte Carlo”. In: *IEEE Transactions on Signal Processing* 67.16 (2019), pp. 4177–4188. DOI: 10.1109/TSP.2019.2926035 (cited on page 80).

[Näg+20] L. Nägele, A. Hoffmann, A. Schierl, and W. Reif. “LegoBot: Automated Planning for Coordinated Multi-Robot Assembly of LEGO structures”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 9088–9095. DOI: 10.1109/IROS45743.2020.9341428 (cited on page 10).

[Neu+18] M. Neumann, K. Nottensteiner, I. Kossyk, and Z. Marton. “Material Classification through Knocking and Grasping by Learning of Structure-Borne Sound under Changing Acoustic Conditions”. In: *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. 2018, pp. 1269–1275. DOI: 10.1109/COASE.2018.8560527 (cited on pages 16, 21).

[New+99] W. S. Newman, M. S. Branicky, H. A. Podgurski, S. Chhatpar, Ling Huang, J. Swaminathan, and Hao Zhang. “Force-responsive robotic assembly of transmission components”. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation*. 1999, pp. 2096–2102. DOI: 10.1109/ROBOT.1999.770416 (cited on page 12).

[Nil69] N. J. Nilsson. “A mobile automaton: An application of artificial intelligence techniques”. In: *IJCAI’69: Proceedings of the 1st International Joint Conference on Artificial Intelligence*. 1969, pp. 509–520. URL: https://dl.acm.org/doi/abs/10.5555/1624562.1624607 (cited on page 9).

[Nit20] Y. Nitta. “PlayStation’s secret weapon: a nearly all-automated factory”. In: *Nikei Asia* (2020). accessed online: 6.08.2021. URL: https://asia.nikkei.com/Business/Companies/PlayStation-s-secret-weapon-a-nearly-all-automated-factory (cited on page 82).

[Not12] K. Nottensteiner. *Schätzung des Kontaktzustands bei robotergestützten Montageaufgaben*. DLR-Interner Bericht DLR-IB 572-2012/12. Diplomarbeit, TU München, 2012. URL: https://elib.dlr.de/79822/ (cited on pages 15, 42).

[Not+16a] K. Nottensteiner, T. Bodenmüller, M. Kaßecker, M. A. Roa, A. Stemmer, T. Stouraitis, D. Seidel, and U. Thomas. “A Complete Automated Chain for Flexible Assembly using Recognition, Planning and Sensor-Based Execution”. In: *Proceedings of ISR 2016: 47st International Symposium on Robotics* (Munich, Germany, June 21–22, 2016). VDE Verlag, 2016. ISBN: 978-3800742318 (cited on pages 13, 20, 72).

[NH17] K. Nottensteiner and K. Hertkorn. “Constraint-based Sample Propagation for Improved State Estimation in Robotic Assembly”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Singapore, May 29–June 3, 2017). IEEE, 2017, pp. 549–556. ISBN: 978-1509046331. DOI: 10.1109/ICRA.2017.7989069 (cited on pages 16, 20, 74).

[NSA21] K. Nottensteiner, A. Sachtler, and A. Albu-Schäffer. “Towards Autonomous Robotic Assembly: Using Combined Visual and Tactile Sensing for Adaptive Task Execution”. In: *Springer Journal of Intelligent & Robotic Systems (JINT)* 101.49 (3 Mar. 2021). DOI: 10.1007/s10846-020-01303-z (cited on pages 16, 20, 61, 64, 75).

[Not+16b] K. Nottensteiner, M. Sagardia, A. Stemmer, and C. Borst. “Narrow Passage Sampling in the Observation of Robotic Assembly Tasks”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)* (Stockholm, Sweden, May 16–21, 2016). IEEE, 2016, pp. 130–137. ISBN: 978-1467380263. DOI: 10.1109/ICRA.2016.7487125 (cited on pages 15, 20, 73).

[Not+13] K. Nottensteiner, A. Stemmer, K. Hertkorn, and C. Borst. “Sequential Monte Carlo in the Observation of Robotic Assembly Tasks”. In: *DGR-Tage 2013 der Deutschen Gesellschaft für Robotik* (Munich, Germany, Oct. 7–8, 2013). Poster. 2013 (cited on page 16).

[NSA20] K. Nottensteiner, F. Stulp, and A. Albu-Schäffer. “Robust Locally Guided Peg-in-hole with Impedance Controlled Robots”. In: *IEEE International Conference on Robotics and Automation (ICRA)* (Paris, France (online), May 31–Aug. 31, 2020). IEEE, 2020, pp. 5771–5777. ISBN: 978-1728173955. DOI: 10.1109/ICRA40945.2020.9196986 (cited on pages 17, 20, 76).

[NV97] M. Nuttin and H. Van Brussel. “Learning the peg-into-hole assembly operation with a connectionist reinforcement technique”. In: *Computers in Industry* 33.1 (1997), pp. 101–109. DOI: 10.1016/S0166-3615(97)00015-8 (cited on page 82).

[NKW10] P. Nyhuis, T. Klemke, and C. Wagner. “Wandlungsfähigkeit – ein systemischer Ansatz”. In: *Wandlungsfähige Produktionssysteme*. Ed. by P. Nyhuis. Berlin: GITO-Verlag, 2010, pp. 3–21. ISBN: 978-3942183154 (cited on page 2).

[OR81] M. S. Ohwovoriole and B. Roth. “An Extension of Screw Theory”. In: *Journal of Mechanical Design* 103.4 (1981), pp. 725–735. DOI: 10.1115/1.3254979 (cited on page 36).

[Oli+19] A. Olivares-Alarcos, D. Beßler, A. Khamis, P. Goncalves, M. K. Habib, J. Bermejo-Alonso, M. Barreto, M. Diab, J. Rosell, J. Quintas, J. Olszewska, H. Nakawala, E. Pignaton, A. Gyrard, S. Borgo, G. Alenyà, M. Beetz, and H. Li. “A review and comparison of ontology-based approaches to robot autonomy”. In: *The Knowledge Engineering Review* 34 (2019). DOI: 10.1017/S0269888919000237 (cited on page 78).

[OG98] M. Onori and P. Groendahl. “MARK III, A new approach to high-variant, medium-volume Flexible Automatic Assembly Cells”. In: *Robotica* 16.3 (1998), pp. 357–368. DOI: 10.1017/S0263574798000411 (cited on page 4).

[Ono02] M. Onori. “Evolvable assembly systems: A new paradigm?” In: *33rd International Symposium on Robotics (ISR)*. 2002. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-128405 (cited on page 4).

[OAH11] M. Onori, H. Akillioglu, and A. Hofmann. “An evolvable robotic assembly cell”. In: *IEEE, IES Conference on Applications of Electronics, Controls, Communications,*

Instrumentation and Computational Intelligence. 2011. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-39036 (cited on pages 4, 9).

[OBF06] M. Onori, J. Barata, and R. Frei. "Evolvable Assembly Systems Basic Principles". In: *Information Technology for Balanced Manufacturing Systems*. IFIP International Federation for Information Processing. Springer, Boston, MA. 2006, pp. 317–328. DOI: 10.1007/978-0-387-36594-7_34 (cited on page 4).

[Ott+08] C. Ott, A. Albu-Schäffer, A. Kugi, and G. Hirzinger. "On the Passivity-Based Impedance Control of Flexible Joint Robots". In: *IEEE Transactions on Robotics* 24.2 (2008), pp. 416–429. DOI: 10.1109/TRO.2008.915438 (cited on page 39).

[Pan+20] Y. Pane, M. H. Arbo, E. Aertbeliën, and W. Decré. "A system architecture for CAD-based robotic assembly with sensor-based skills". In: *IEEE Transactions on Automation Science and Engineering* 17.3 (2020), pp. 1237–1249. DOI: 10.1109/TASE.2020.2980628 (cited on page 78).

[PC94] Y. Park and H. Cho. "A self-learning rule-based control algorithm for chamferless part mating". In: *Control Engineering Practice* 2.5 (1994), pp. 773–783. DOI: 10.1016/0967-0661(94)90342-5 (cited on page 83).

[Per+19] A. Perzylo, M. Rickert, B. Kahl, N. Somani, C. Lehmann, A. Kuss, S. Profanter, A. B. Beck, M. Haage, M. Rath Hansen, M. T. Nibe, M. A. Roa, O. Sörnmo, S. Gestegård Robertz, U. Thomas, G. Veiga, E. A. Topp, I. Kessler, and M. Danzer. "SMErobotics: Smart Robots for Flexible Manufacturing". In: *IEEE Robotics and Automation Magazine* 26.1 (2019), pp. 78–90. DOI: 10.1109/MRA.2018.2879747 (cited on page 13).

[Per+15] A. Perzylo, N. Somani, S. Profanter, A. Gaschler, S. Griffiths, M. Rickert, and A. Knoll. "Ubiquitous Semantics: Representing and Exploiting Knowledge, Geometry, and Language for Cognitive Robot Systems". In: *Proceedings of the Workshop Towards Intelligent Social Robots - Current Advances in Cognitive Robotics, IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS)*. 2015. URL: https://mediatum.ub.tum.de/1280411 (cited on page 78).

[Pfa+18] M. Pfanne, M. Chalon, F. Stulp, and A. Albu-Schäffer. "Fusing Joint Measurements and Visual Features for In-Hand Object Pose Estimation". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3497–3504. DOI: 10.1109/LRA.2018.2853652 (cited on page 7).

[Pin93] B. J. Pine II. *Mass customization: the new frontier in business competition*. Harvard Business School Press, 1993. ISBN: 978-0875849461 (cited on page 2).

[PS99] M. K. Pitt and N. Shephard. "Filtering via Simulation: Auxiliary Particle Filters". In: *Journal of the American Statistical Association* 94.446 (1999), pp. 590–599. DOI: 10.1080/01621459.1999.10474153 (cited on page 51).

[RV15] P. Rebeschini and R. Van Handel. “Can local particle filters beat the curse of dimensionality?” In: *The Annals of Applied Probability* 25.5 (2015), pp. 2809–2866. DOI: 10.1214/14-AAP1061 (cited on page 80).

[Rem88] I. Rembold. “The Karlsruhe autonomous mobile assembly robot”. In: *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. 1988, pp. 598–603. DOI: 10.1109/ROBOT.1988.12117 (cited on page 9).

[Roa+22] M. A. Roa, C. E. S. Koch, M. Rognant, A. Ummel, P. Letier, A. Turetta, P. L. Negro, S. Trinh, I. V. Rodríguez Brena, K. Nottensteiner, J. Rouvinet, V. Bissonnette, G. Grunwald, and T. Germa. “PULSAR: Testing the Technologies for In-Orbit Assembly of a Large Telescope”. In: *ESA Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*. 2022 (cited on pages 14, 22, 87).

[Roa+19] M. A. Roa, K. Nottensteiner, G. Grunwald, P. L. Negro, A. Cuffolo, S. Andiappane, M. Rognant, A. Verhaeghe, and V. Bissonnette. “In-Space Robotic Assembly of Large Telescops”. In: *ESA Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*. 2019 (cited on pages 14, 22).

[Roa+17] M. A. Roa, K. Nottensteiner, A. Wedler, and G. Grunwald. “Robotic Technologies for In-Space Assembly Operations”. In: *ESA Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*. 2017 (cited on pages 14, 22, 87).

[Rob20a] Robotics and Mechatronics Center of the DLR (DLR RMC). *Autonomous robotic assembly of individual products made out of aluminum profiles*. Video summary of the AUTOMATICA 2018 demonstrator. Apr. 13, 2020. URL: https://youtu.be/XQhXGJbUURE (cited on pages 14, 26, 87).

[Rob20b] Robotics and Mechatronics Center of the DLR (DLR RMC). *SARA*. Description of the lightweight robot SARA on the website of the DLR RM institute, accessed online: 5.03.2022. 2020. URL: https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-11709 (cited on page 87).

[Rob21] Robotics and Mechatronics Center of the DLR (DLR RMC). *PULSAR (Prototype of an Ultra Large Structure Assembly Robot)*. Video summary of the EU H2020 PULSAR demonstrator at DLR (dPAMT). July 20, 2021. URL: https://youtu.be/1kyrO8k5rj0 (cited on page 14).

[Rod+20] I. V. Rodríguez Brena, K. Nottensteiner, D. Leidner, M. Durner, F. Stulp, and A. Albu-Schäffer. “Pattern Recognition for Knowledge Transfer in Robotic Assembly Sequence Planning”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3666–3673. DOI: 10.1109/LRA.2020.2979622 (cited on pages 14, 21, 26, 29, 77, 87).

[Rod+19] I. V. Rodríguez Brena, K. Nottensteiner, D. Leidner, M. Kaßecker, F. Stulp, and A. Albu-Schäffer. “Iteratively Refined Feasibility Checks in Robotic Assembly Sequence Planning”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1416–1423. DOI: 10.1109/LRA.2019.2895845 (cited on pages 14, 21, 26, 29, 31, 33, 77, 87).

[Rod+21] I. V. Rodríguez Brena, A. S. Bauer, K. Nottensteiner, D. Leidner, G. Grunwald, and M. A. Roa. “Autonomous Robot Planning System for In-Space Assembly of Reconfigurable Structures”. In: *2021 IEEE Aerospace Conference*. 2021, pp. 1–17. DOI: 10.1109/AERO50100.2021.9438257 (cited on pages 22, 87).

[Rog11] S. Rogalski. *Flexibility Measurement in Production Systems*. Springer, 2011. DOI: 10.1007/978-3-642-18117-7 (cited on page 2).

[RMW96] F. Röhrdanz, H. Mosemann, and F. Wahl. “Generating and evaluating stable assembly sequences”. In: *Advanced robotics* 11.2 (1996), pp. 97–126. DOI: 10.1163/156855397X00272 (cited on page 26).

[RBS05] J. Rosell, L. Basañez, and R. Suárez. “Predicting planar motion behavior under contact uncertainty”. In: *Advanced Robotics* 19.5 (2005), pp. 567–590. DOI: 10.1163/156855305323383802 (cited on pages 12, 55, 67).

[Rub88] D. B. Rubin. “Using the SIR algorithm to simulate posterior distributions”. In: *Bayesian statistics* 3 (1988), pp. 395–402 (cited on page 54).

[Sac16] A. Sachtler. *Thema 1: Manuelle Registrierung von Objekten in Roboterarbeitszellen unter Verwendung von Wissensdatenbanken, Thema 2: Automatisierte Selbstkalibrierung von Robotern in der Arbeitszelle*. Bericht zum Modul T2000. DHBW Baden-Würtemberg, Mannheim; DLR Institut für Robotik und Mechatronik, 2016 (cited on page 33).

[Sac+19] A. Sachtler, K. Nottensteiner, M. Kaßecker, and A. Albu-Schäffer. “Combined Visual and Touch-based Sensing for the Autonomous Registration of Objects with Circular Features”. In: *IEEE 19th International Conference on Advanced Robotics (ICAR)*. 2019, pp. 426–433. DOI: 10.1109/ICAR46387.2019.8981602 (cited on pages 16, 18, 21, 33, 57, 64, 66, 75, 87).

[Sag19] M. Sagardia Erasun. “Virtual Manipulations with Force Feedback in Complex Interaction Scenarios”. PhD thesis. Technische Universität München, 2019 (cited on pages 16, 43).

[Sah18] K. Saho. “Kalman Filter for Moving Object Tracking: Performance Analysis and Filter Design”. In: *Kalman Filters*. Ed. by G. L. de Oliveira Serra. IntechOpen, 2018. Chap. 12. DOI: 10.5772/intechopen.71731 (cited on page 61).

[Sal84] J. K. Salisbury Jr. “Interpretation of Contact Geometries from Force Measurements”. In: *Proceedings. 1984 IEEE International Conference on Robotics and Automation*. 1984, pp. 240–247. DOI: 10.1109/ROBOT.1984.1087180 (cited on page 10).

[Sal85] J. K. Salisbury Jr. “Interpretation of Contact Geometries from Force Measurements”. In: *Robot Hands and the Mechanics of Manipulation*. Ed. by M. T. Mason and J. K. Salisbury Jr. The MIT Press Series in Artificial Intelligence. The MIT Press, 1985, pp. 133–150. ISBN: 978-0262132053 (cited on page 10).

[SEL91] C. Samson, B. Espiau, and M. Le Borgne. *Robot Control: The Task Function Approach*. Oxford University Press, 1991. ISBN: 978-0198538059 (cited on page 11).

[Sav+18] T. R. Savarimuthu, A. G. Buch, C. Schlette, N. Wantia, J. Rossmann, D. Martinez, G. Alenya, C. Torras, A. Ude, B. Nemec, A. Kramberger, F. Worgotter, E. E. Aksoy, J. Papon, S. Haller, J. Piater, and N. Kruger. “Teaching a Robot the Semantics of Assembly Tasks”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48.5 (2018), pp. 670–692. DOI: 10.1109/TSMC.2016.2635479 (cited on page 10).

[Sch+21a] P. M. Schäfer, S. Schneyer, T. Bachmann, M. Knauer, F. Steinmetz, and K. Nottensteiner. *Factory of the Future Ontology*. Version 1.0. 2021. DOI: 10.5281/zenodo.4650309 (cited on page 28).

[Sch+21b] P. M. Schäfer, F. Steinmetz, S. Schneyer, T. Bachmann, T. Eiband, F. S. Lay, A. Padalkar, C. Sürig, F. Stulp, and K. Nottensteiner. “Flexible Robotic Assembly Based on Ontological Representation of Tasks, Skills, and Resources”. In: *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*. 2021, pp. 702–706. DOI: 10.24963/kr.2021/73 (cited on pages 14, 21, 78, 87).

[SRD19] S. Scherzinger, A. Roennau, and R. Dillmann. “Contact Skill Imitation Learning for Robot-Independent Assembly Programming”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 4309–4316. DOI: 10.1109/IROS40897.2019.8967523 (cited on page 83).

[SVB15] C. Scheuermann, S. Verclas, and B. Bruegge. “Agile Factory - An Example of an Industry 4.0 Manufacturing Process”. In: *2015 IEEE 3rd International Conference on Cyber-Physical Systems, Networks, and Applications*. 2015, pp. 43–47. DOI: 10.1109/CPSNA.2015.17 (cited on pages 4, 78).

[SP94] J. Schimmels and M. Peshkin. “Force-assembly with friction”. In: *IEEE Transactions on Robotics and Automation* 10.4 (1994), pp. 465–479. DOI: 10.1109/70.313097 (cited on page 12).

[SB14] F. Schmidt and R. Burger. “How we deal with software complexity in robotics: ’links and nodes’ and the ’robotkernel’”. In: *Workshop on Software Architectures and Methodologies for Developing Humanoid Robots. 14th IEEE-RAS International Conference on Humanoid Robots.* (Madrid, Spain, Nov. 18, 2014). 2014 (cited on page 29).

[SGN05] T. Schön, F. Gustafsson, and P.-J. Nordlund. “Marginalized particle filters for mixed linear/nonlinear state-space models”. In: *IEEE Transactions on Signal Processing* 53.7 (2005), pp. 2279–2289. DOI: 10.1109/TSP.2005.849151 (cited on page 51).

[SM21] R. Schönhof and M. Moosmann. “Maschinelles Lernen pusht Robotik”. In: *Robotik und Produktion* 6.6 (2021) (cited on page 3).

[Sim+82] J. Simons, H. Brussel, J. De Schutter, and J. Verhaert. “A self-learning automaton with variable resolution for high precision assembly by industrial robots”. In: *IEEE Transactions on Automatic Control* 27.5 (1982), pp. 1109–1113. DOI: 10.1109/TAC.1982.1103060 (cited on page 83).

[SV00] M. Skubic and R. A. Volz. “Acquiring robust, force-based assembly skills from human demonstration”. In: *IEEE Transactions on Robotics and Automation* 16.6 (2000), pp. 772–781. DOI: 10.1109/70.897788 (cited on page 24).

[Sny+08] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. “Obstacles to High-Dimensional Particle Filtering”. In: *Monthly Weather Review* 136.12 (2008), pp. 4629–4640. DOI: 10.1175/2008MWR2529.1 (cited on page 80).

[SDA18] J. Sola, J. Deray, and D. Atchuthan. “A micro Lie theory for state estimation in robotics”. In: *arXiv preprint arXiv:1812.01537* (2018) (cited on page 35).

[SKS16] H.-C. Song, Y.-L. Kim, and J.-B. Song. “Guidance algorithm for complex-shape peg-in-hole strategy based on geometrical information and force control”. In: *Advanced Robotics* 30.8 (2016), pp. 552–563. DOI: 10.1080/01691864.2015.1130172 (cited on page 12).

[SA70] H. W. Sorenson and D. L. Alspach. “Gaussian sum approximations for nonlinear filtering”. In: *1970 IEEE Symposium on Adaptive Processes (9th) Decision and Control*. 1970, pp. 193–193. DOI: 10.1109/SAP.1970.270017 (cited on page 51).

[Sta09] E. Staffetti. “Analysis of Rigid Body Interactions for Compliant Motion Tasks Using the Grassmann-Cayley Algebra”. In: *IEEE Transactions on Automation Science and Engineering* 6.1 (2009), pp. 80–93. DOI: 10.1109/TASE.2008.917041 (cited on pages 33, 42).

[SW16] F. Steinmetz and R. Weitschat. “Skill parametrization approaches and skill architecture for human-robot interaction”. In: *2016 IEEE International Conference*

on Automation Science and Engineering (CASE). 2016, pp. 280–285. DOI: 10.1109/COASE.2016.7743419 (cited on pages 12, 24, 28).

[SWW18] F. Steinmetz, A. Wollschläger, and R. Weitschat. "RAZER—A HRI for Visual Task-Level Programming and Intuitive Skill Parameterization". In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1362–1369. DOI: 10.1109/LRA.2018.2798300 (cited on page 24).

[SNS19] F. Steinmetz, V. Nitsch, and F. Stulp. "Intuitive task-level programming by demonstration through semantic skill recognition". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3742–3749. DOI: 10.1109/LRA.2019.2928782 (cited on page 24).

[SAH07] A. Stemmer, A. Albu-Schäffer, and G. Hirzinger. "An Analytical Method for the Planning of Robust Assembly Tasks of Complex Shaped Planar Parts". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 317–323. DOI: 10.1109/ROBOT.2007.363806 (cited on pages 12, 16, 67, 75, 83).

[Ste+06] A. Stemmer, G. Schreiber, K. Arbter, and A. Albu-Schäffer. "Robust assembly of complex shaped planar parts using vision and force". In: *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. 2006, pp. 493–500. DOI: 10.1109/MFI.2006.265671 (cited on pages 8, 12).

[Sto15] A. Stolt. "On robotic assembly using contact force control and estimation". PhD thesis. Lund University, 2015 (cited on pages 12, 79).

[SZP18] F. Suárez-Ruiz, X. Zhou, and Q.-C. Pham. "Can robots assemble an IKEA chair?" In: *Science Robotics* 3.17 (2018). DOI: 10.1126/scirobotics.aat6385 (cited on page 10).

[Sun+05] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. Reif. "Narrow Passage Sampling for Probabilistic Roadmap Planning". In: *IEEE Transactions on Robotics* 21.6 (2005), pp. 1105–1115. DOI: 10.1109/TRO.2005.853485 (cited on pages 16, 59).

[Sür21] C. Sürig. *Achieving Autonomous Task Execution by Integrating a Combination of Ontologies and Semantic Languages Into a Human Factory Interface*. DLR-Interner Bericht DLR-IB-RM-OP-2021-56. Masterarbeit, TU München, 2021. URL: https://elib.dlr.de/141639/ (cited on page 26).

[Tab17] M. Taboga. *Lectures on probability theory and mathematical statistics*. CreateSpace Independent Publishing Platform, 2017 (cited on page 45).

[TMO10] Y. Taguchi, T. K. Marks, and H. Okuda. "Rao-Blackwellized Particle Filtering for Probing-Based 6-DOF Localization in Robotic Assembly". In: *2010 IEEE*

International Conference on Robotics and Automation. IEEE, 2010, pp. 2610–2617. DOI: 10.1109/ROBOT.2010.5509478 (cited on pages 11, 58, 80).

[Tan07] P. Tang. "Representation and automatic generation of contact state graphs between general solid objects". PhD thesis. The University of North Carolina at Charlotte, 2007 (cited on pages 11, 42).

[TX08] P. Tang and J. Xiao. "Automatic Generation of High-level Contact State Space between 3D Curved Objects". In: *The International Journal of Robotics Research* 27.7 (2008), pp. 832–854. DOI: 10.1177/0278364908092463 (cited on page 11).

[TR88] R. H. Taylor and V. T. Rajan. "The Efficient Computation Of Uncertainty Spaces For Sensor-based Robot Planning". In: (1988), pp. 231–236. DOI: 10.1109/IROS.1988.593281 (cited on page 56).

[Tay76] R. H. Taylor. "The synthesis of manipulator control programs from task-level specifications." PhD thesis. Stanford University, 1976 (cited on pages 23, 25, 28, 56).

[Tho+18] G. Thomas, M. Chien, A. Tamar, J. A. Ojea, and P. Abbeel. "Learning robotic assembly from cad". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 3524–3531. DOI: 10.1109/ICRA.2018.8460696 (cited on page 83).

[TBW03] U. Thomas, M. Barrenscheen, and F. Wahl. "Efficient assembly sequence planning using stereographical projections of C-space obstacles". In: *Proceedings of the IEEE International Symposium on Assembly and Task Planning*. 2003, pp. 96–102. DOI: 10.1109/ISATP.2003.1217194 (cited on page 26).

[Tho+03] U. Thomas, B. Finkemeyer, T. Kröger, and F. Wahl. "Error-tolerant execution of complex robot tasks based on skill primitives". In: *2003 IEEE International Conference on Robotics and Automation*. Vol. 3. 2003, pp. 3069–3075. DOI: 10.1109/ROBOT.2003.1242062 (cited on pages 12, 24, 28).

[Tho+13] U. Thomas, G. Hirzinger, B. Rumpe, C. Schulze, and A. Wortmann. "A new skill based robot programming language using UML/P Statecharts". In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 461–466. DOI: 10.1109/ICRA.2013.6630615 (cited on pages 12, 24, 28).

[TW01] U. Thomas and F. Wahl. "A system for automatic planning, evaluation and execution of assembly sequences for industrial robots". In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 3. 2001, pp. 1458–1464. DOI: 10.1109/IROS.2001.977186 (cited on page 26).

[Tho+07] U. Thomas, S. Molkenstruck, R. Iser, and F. Wahl. "Multi Sensor Fusion in Robot Assembly Using Particle Filters". In: *Proceedings 2007 IEEE International*

Conference on Robotics and Automation. 2007, pp. 3837–3843. DOI: 10.1109/ROBOT.2007.364067 (cited on page 11).

[TSR15] U. Thomas, T. Stouraitis, and M. A. Roa. "Flexible assembly through integrated assembly sequence planning and grasp planning". In: *2015 IEEE International Conference on Automation Science and Engineering (CASE)*. 2015, pp. 586–592. DOI: 10.1109/CoASE.2015.7294142 (cited on pages 26, 31, 32, 72).

[TW10] U. Thomas and F. M. Wahl. "Assembly Planning and Task Planning — Two Prerequisites for Automated Robot Programming". In: *Robotic Systems for Handling and Assembly*. Ed. by D. Schütz and F. M. Wahl. Vol. 67. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2010, pp. 333–354. DOI: 10.1007/978-3-642-16785-0_19 (cited on pages 26, 72).

[TBF05] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. The MIT Press, 2005. ISBN: 978-0262201629 (cited on pages 11, 51).

[Vec+19] M. Vecerik, O. Sushkov, D. Barker, T. Rothörl, T. Hester, and J. Scholz. "A practical approach to insertion with variable socket position using deep reinforcement learning". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 754–760. DOI: 10.1109/ICRA.2019.8794074 (cited on page 83).

[VD16] L. Villani and J. De Schutter. "Force control". In: *Springer Handbook of Robotics*. Ed. by B. Siciliano and O. Khatibm. Springer, 2016, pp. 931–954. DOI: 10.1007/978-3-319-32552-1 (cited on pages 12, 38).

[Vil+18] V. Villani, F. Pini, F. Leali, C. Secchi, and C. Fantuzzi. "Survey on Human-Robot Interaction for Robot Programming in Industrial Applications". In: *IFAC-PapersOnLine* 51.11 (2018). 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018, pp. 66–71. DOI: 10.1016/j.ifacol.2018.08.236 (cited on page 4).

[VJH21] F. Voigt, L. Johannsmeier, and S. Haddadin. "Multi-Level Structure vs. End-to-End-Learning in High-Performance Tactile Robotic Manipulation". In: *Proceedings of the 2020 Conference on Robot Learning*. 2021, pp. 2306–2316. URL: https://proceedings.mlr.press/v155/voigt21a.html (cited on page 82).

[Wah+15] A. Wahrburg, S. Zeiss, B. Matthias, J. Peters, and H. Ding. "Combined pose-wrench and state machine representation for modeling Robotic Assembly Skills". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 852–857. DOI: 10.1109/IROS.2015.7353471 (cited on pages 12, 24, 28, 79).

[Wal72] K. J. Waldron. "A method of studying joint geometry". In: *Mechanism and Machine Theory* 7.3 (1972), pp. 347–353. DOI: 10.1016/0094-114X(72)90043-2 (cited on page 42).

[Wat78] P. C. Watson. “Remote Center Compliance System”. US4098001. The Charles Stark Draper Laboratory. 1978 (cited on page 38).

[WKW14] I. Weidauer, D. Kubus, and F. Wahl. “A hierarchical extension of manipulation primitives and its integration into a robot control architecture”. In: *2014 IEEE International Conference on Robotics and Automation*. 2014, pp. 5401–5407. DOI: 10.1109/ICRA.2014.6907653 (cited on pages 12, 24, 28).

[WN79] D. Whitney and J. L. Nevins. “What is the Remote Center Compliance (RCC) and what can it do”. In: *Proceedings of the 9th International Symposium on Industrial Robots* (Washington D.C., USA, Mar. 13–15, 1979). Dearborn: Society of Manufacturing Engineers, 1979. ISBN: 978-0872630482 (cited on page 8).

[Wie+07] H.-P. Wiendahl, H. A. ElMaraghy, P. Nyhuis, M. F. Zäh, H.-H. Wiendahl, N. Duffie, and M. Brieke. “Changeable manufacturing-classification, design and operation”. In: *CIRP Annals* 56.2 (2007), pp. 783–809. DOI: 10.1016/j.cirp.2007.10.003 (cited on pages 4, 78).

[Wil92] R. H. Wilson. “On geometric assembly planning”. PhD thesis. Stanford University, 1992 (cited on page 25).

[Win71] T. Winograd. “Procedures as a representation for data in a computer program for understanding natural language”. PhD thesis. Massachusetts Institute of Technology, 1971 (cited on page 25).

[Wir+18] F. Wirnshofer, P. S. Schmitt, W. Feiten, G. v. Wichert, and W. Burgard. “Robust, Compliant Assembly via Optimal Belief Space Planning”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 5436–5443. DOI: 10.1109/ICRA.2018.8460995 (cited on pages 12, 67).

[Wir+19] F. Wirnshofer, P. S. Schmitt, P. Meister, G. v. Wichert, and W. Burgard. “State Estimation in Contact-Rich Manipulation”. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 3790–3796. DOI: 10.1109/ICRA.2019.8793572 (cited on pages 11, 62).

[Wir21] F. Wirnshofer. “State estimation and planning under uncertainty for robot manipulation”. PhD thesis. Universität Freiburg, 2021. DOI: 10.6094/UNIFR/219450 (cited on page 79).

[Wol89] J. D. Wolter. “On the automatic generation of assembly plans”. In: *Proceedings, 1989 International Conference on Robotics and Automation*. 1989, pp. 62–68. DOI: 10.1109/ROBOT.1989.99968 (cited on pages 24, 25).

[Wu+21] Z. Wu, W. Lian, V. Unhelkar, M. Tomizuka, and S. Schaal. “Learning Dense Rewards for Contact-Rich Manipulation Tasks”. In: *2021 IEEE International*

Conference on Robotics and Automation (ICRA) (2021), pp. 6214–6221. DOI: 10.1109/ICRA48506.2021.9561891 (cited on page 83).

[Wüt+15] M. Wüthrich, J. Bohg, D. Kappler, C. Pfreundt, and S. Schaal. "The coordinate particle filter-a novel particle filter for high dimensional systems". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 2454–2461. DOI: 10.1109/ICRA.2015.7139527 (cited on page 80).

[XJ01] J. Xiao and X. Ji. "Automatic Generation of High-level Contact State Space". In: *The International Journal of Robotics Research* 20.7 (2001), pp. 584–606. DOI: 10.1177/02783640122067552 (cited on pages 11, 42).

[Xia90] J. Xiao. "On uncertainty handling in robot part-mating planning". PhD thesis. University of Michigan, 1990 (cited on pages 12, 24).

[Xia93] J. Xiao. "Automatic Determination of Topological Contacts in the Presence of Sensing Uncertainties". In: *Proceedings IEEE International Conference on Robotics and Automation*. 1993, pp. 65–70. DOI: 10.1109/ROBOT.1993.291962 (cited on pages 10, 42).

[XV89] J. Xiao and R. A. Volz. "On replanning for assembly tasks using robots in the presence of uncertainties". In: *1989 IEEE International Conference on Robotics and Automation*. 1989, pp. 638–645. DOI: 10.1109/ROBOT.1989.100056 (cited on page 12).

[Yas17] Yaskawa America, Inc. *Conveyor Tracking Setup*. Application Note AN.MP3300iec.03. Yaskawa, Nov. 16, 2017. URL: https://www.yaskawa.com/downloads/search-index/details?showType=details&docnum=AN.MP3300iec.03 (cited on page 60).

[YR18] K.-T. Yu and A. Rodriguez. "Realtime State Estimation with Tactile and Visual Sensing for Inserting a Suction-held Object". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 1628–1635. DOI: 10.1109/IROS.2018.8594077 (cited on page 11).

[Zäh+09] M. F. Zäh, M. Beetz, K. Shea, G. Reinhart, K. Bender, C. Lau, M. Ostgathe, W. Vogl, M. Wiesbeck, M. Engelhard, C. Ertelt, T. Rühr, M. Friedrich, and S. Herle. "The Cognitive Factory". In: *Changeable and Reconfigurable Manufacturing Systems*. Ed. by H. A. ElMaraghy. Springer Series in Advanced Manufacturing. Springer London, 2009, pp. 355–371. DOI: 10.1007/978-1-84882-067-8_20 (cited on pages 4, 78).

[ZSŠ75] V. Zaritskii, V. Svetnik, and L. Šimelevič. "Monte-Carlo technique in problems of optimal information processing". In: *Avtomatika i Telemekhanika, Autom. Remote Control,* 36.12 (1975), pp. 95–103. URL: http://mi.mathnet.ru/eng/at8123 (cited on page 51).

[ZT12] L. E. Zhang and J. C. Trinkle. “The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing”. In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 3805–3812. DOI: 10.1109/ICRA.2012.6225125 (cited on pages 11, 62).

[Zho14] Y. Zhou. “vSMC: Parallel Sequential Monte Carlo in C++”. In: *Journal of Statistical Software* 62.9 (2014), pp. 1–49. DOI: 10.18637/jss.v062.i09 (cited on pages 51, 80).

[Zis17] S. Zistl. “The Future of Manufacturing: Prototype Robot Solves Problems without Programming”. In: *Siemens AG News* (Dec. 11, 2017). accessed online: 12.08.2021. URL: https://new.siemens.com/global/en/company/stories/research-technologies/artificial-intelligence/prototype-robot-solves-problems-without-programming.html (cited on page 10).

Full Text of Publications

Publication 1

K. Nottensteiner, T. Bodenmüller, M. Kaßecker, M. A. Roa, A. Stemmer, T. Stouraitis, D. Seidel, and U. Thomas (2016): "*A Complete Automated Chain for Flexible Assembly using Recognition, Planning and Sensor-Based Execution*". 47st International Symposium on Robotics (ISR). VDE Verlag, 2016.

Version Note

The following attached version corresponds to the accepted manuscript of the publication.

The final published version is available under:

- https://www.vde-verlag.de/proceedings-en/454231052.html
- https://ieeexplore.ieee.org/document/7559140

Please refer to the final published version for citation:

```
@InProceedings{Nottensteiner2016a,
 author =       {Nottensteiner, Korbinian and Bodenmüller, Tim and
                 Kaßecker, Michael and Máximo A. Roa and Stemmer,
                 Andreas and Stouraitis, Theodoros and Seidel, Daniel
                 and Thomas, Ulrike},
  title =       {A Complete Automated Chain for Flexible Assembly
                 using Recognition, Planning and Sensor-Based
                 Execution},
  booktitle =   {Proceedings of ISR 2016: 47st International
                 Symposium on Robotics},
  year =        2016,
  venue =       {Munich, Germany},
  eventdate =   {2016-06-21/2016-06-22},
  publisher =   {VDE Verlag},
  isbn =        {978-3800742318},
  eid =         {2-s2.0-84983738930},
```

A Complete Automated Chain for Flexible Assembly using Recognition, Planning and Sensor-Based Execution

Korbinian Nottensteiner[1], Tim Bodenmüller[1], Michael Kaßecker[1], Máximo A. Roa[1], Andreas Stemmer[1], Theodoros Stouraitis[1], Daniel Seidel[2], and Ulrike Thomas[3]
[1] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), korbinian.nottensteiner@dlr.de
[2] Sensor Based Robotic Systems and Intelligent Assistance Systems, TU Munich, daniel.seidel@tum.de
[3] Robotics and Human Machine Interaction Lab, TU Chemnitz, ulrike.thomas@etit.tu-chemnitz.de

Abstract

This paper presents a fully automated system for automatic assembly of aluminum profile constructions. This high grade of automation of the entire process chain requires novel strategies in recognition, planning and execution. The system includes an assembly sequence planner integrated with a grasp planning tool, a knowledge-based reasoning method, a skill-based code generation, and an error tolerant execution engine. The modular structure of the system allows its adaptation to new products, which can prove especially useful for SMEs producing small lot sizes. The system is robust and stable, as demonstrated with the repeated execution of different geometric assemblies.

1 Introduction

In large-scale manufacturing, automation with industrial robots is a key technology for production at sustained high quality with low cost. Compared to this well-established domain, the application of robots in small and medium-sized enterprises (SMEs) is still challenging. Here, the production is typically dominated by small lot sizes and a high number of variations in the products. Altogether, this requires the possibility of fast changeovers and reconfiguration of production processes [5]. Conventional automation solutions are far too complex, specialized for a certain product variant, and often involve expensive facilities with several assembly stations. Reconfiguration of assembly lines from one variant to another is time-consuming and usually requires careful re-engineering and re-programming, hence it is too expensive and often dominates the production costs. Therefore, SMEs demand more flexible systems that can be installed and configured by factory floor workers rather than robotic experts, thus reducing programming and engineering efforts.

In recent years, new robotic technologies like the lightweight-robot arm (LWR, [2]) became available, which meet the special requirements of flexible environments like the production facilities of SMEs. Robots equipped with torque measurement units in their joints allow safe interaction with the factory floor worker [6]. Assembly strategies using the variable stiffness provided by the impedance control show that a robust handling of tight fittings and sensitive parts is possible without specialized and therefore expensive additional hardware [15]. Despite this new robotic technology, there are still barriers for applications in real SME productions, as nowadays the programming and installation of these robots is complex and requires the intervention of robotic experts.

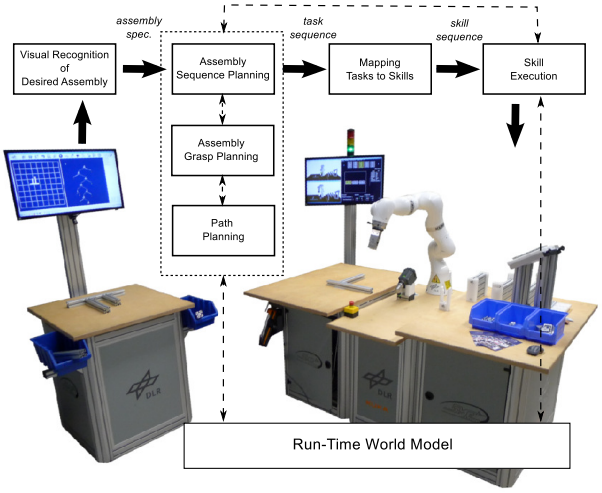


Figure 1: The user initiates the assembly process by showing the desired product on the demonstration table (left). This assembly is visually recognized, planned and automatically transferred to a sequence of robotic skills executed in the workcell (right).

This work tries to fill such gap by providing a new assembly planning algorithm tightly integrated with a grasp planner, a robotic skill engine, and a reasoning system to map the assembly tasks to executable skills. Fig. 1 provides an overview of the integrated system, the implemented components and the interaction between them. This system is applied to the flexible assembly of structures based on the *item MB Building Kit System*, consisting of aluminum profiles and angle brackets as connectors. First, the desired assembly is recognized and the poses of individual parts are located (Section 3). A knowledge base provides a set of grasps for each single object, calculated by a generic grasp planner (Section 4). Then, an assembly planner generates feasible assembly sequences (Section 5). It calls the grasp planner when

needed to prune potential sequences and retain only the feasible ones. The output is an assembly sequence consisting of an order of tasks. Based on a task to skill mapping algorithm, the tasks are instantiated automatically with executable skills whose parameters come from the actual desired assembly. After this instantiation of skills, the robot program is executed and the product is assembled by the application of assembly skills (Section 6). The approach is implemented and applied to the generation of geometrically different assemblies (Section 7). The following sections summarize the related work and provide the relevant details of the system.

2 Related Work

Providing a fully automated chain from assembly specification to execution tackles many different issues, both in the planning and execution domains. Assembly sequence planners are available since the early 80s, based for instance on information gathered from questions for end-users [4], on the automatic analysis of directional and non-directional-blocking graphs [10], or on geometrical information obtained with the analysis of the motion space [7]. However, these planners are still not widely used in industry nowadays, mainly due to the difficulty of dealing with uncertainties and to the inherent complexity of assembly sequencing. Although modern CAD software assists the engineers with the planning process, the validation of a plan in the available robotic system is still weak and requires high amount of manual labor and expert knowledge. In order to cope with these limitations, we have developed a new planning algorithm and a skill execution engine that offers a symbolic interface to the assembly system components.

The sequence planner is coupled with a grasp planner which is required to deal with the large number of product variants in our automated one-of-a-kind production. Grasp planners automatically generate feasible grasps for a part based on geometrical and physical analysis [3]. The generic grasp planner used here was developed for multi-fingered hands, and generates force closure grasps for any gripper and object geometry [14]. It is adapted to provide fast answers to grasp feasibility queries, which help to prune the search of an executable assembly sequence.

Various concepts of manipulation skills have been introduced to tackle uncertainties during robotic executions [8, 12, 18]. For assembly tasks, [15] demonstrates a method to increase robustness against pose uncertainties. Robustness is especially important when actions are selected and parameterized automatically and success rates have to be guaranteed in changing contexts. In the presented system, the sensing and control capabilities of the LWR are used to implement a library of assembly skills ensuring high execution rates.

To the best of the authors' knowledge, the system described in this paper is the first one integrating all the described components to automatically generate assembly programs for each individual product.

3 Visual Assembly Recognition

A completely untrained layman is able to instruct the proposed assembly system to manufacture a wide variety of product variants. This section discusses how to achieve this goal without actually feeding blueprints to the system. According to the social cognitive learning theory, humans learn by observing behaviors, thus an effective and intuitive way to teach people is by using demonstrations. The same theory was applied to the proposed system, where the demonstration consists of arranging profiles and brackets on a workbench (Fig. 1), while the system is supervising the process with a camera. A visual feedback of the systems' understanding of the desired assembly is given on a monitor (Fig. 2).

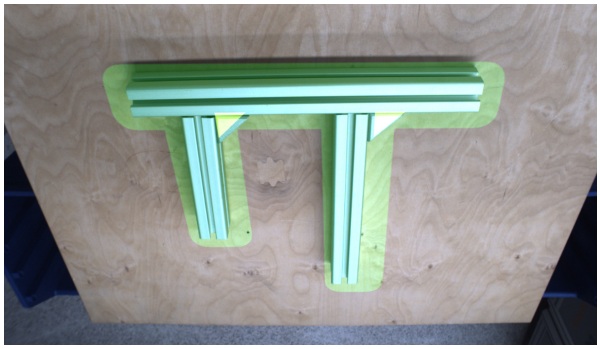


Figure 2: Recognition of the desired assembly by the camera above the workbench, and visual feedback to the user highlighting the detected structure (in green).

The recognition process is made up of three major phases. In a perceptive phase, profiles and brackets are visually identified, and the pose of each structural element is estimated. The result is a structure containing an error attributed to both sensoric inaccuracies and human impreciseness. The former error mainly stems from lighting effects, whereas the latter is caused by human inability to accurately position elements. Therefore, a rule set with heuristics is applied in the next phase to obtain the nominal assembly requested by the user.

In the cognitive phase, the detected parts are aligned with respect to constraints coming from the fact that the demonstration contains an arrangement of profiles lying on a flat table. Thus, it is assumed that the elements are in-plane, not rotated, connected orthogonally and strictly joined. To keep the character of the whole structure as it was supposedly *meant* by the user, an optimization step is introduced to minimize the shifting error. Without this step, geometrical errors would be accumulated for the position of the element that was aligned in the last step. For that purpose L-, I- and T-shaped substructures are identified and aligned bottom-up, and are successively grouped into larger structures, with priority to L- and T-shapes. At last, brackets are detected and if necessary aligned, shifted and rotated. Hidden elements in the camera view, such as screws and nuts, are added according to predefined rules, in order to complete the assembly.

The third and final phase creates an assembly specification, including type and relative poses of the parts.

4 Assembly Grasp Planning

A generic grasp planning module was developed to generate feasible grasps for any object described as a point-cloud, using any type of end effector [17]. The grasp planner generates a database of feasible grasps for a free-floating object using a sampling-based approach that creates different possibilities for grabbing the object. These possible grasps can be later filtered in an online process to discard possible collisions with supporting surfaces or obstacles in the scene.

In the assembly grasp planning module, the objects are grouped in two categories: objects that allow only a reduced set of grasp configurations (according to the gripper selected for the application), like the angle bracket and slot nut, and objects where a variety of grasps can be applied on continuous regions, like the profiles and all the subassemblies. For the objects of the first category, feasible grasps can be predefined by the user and employed throughout the assembly process. The second category requires an automatic determination of feasible grasp regions and a process to generate grasps on it. The geometry of the subassemblies is known only at planning time, therefore the system must provide a quick answer to queries on feasible grasps for different groups of objects.

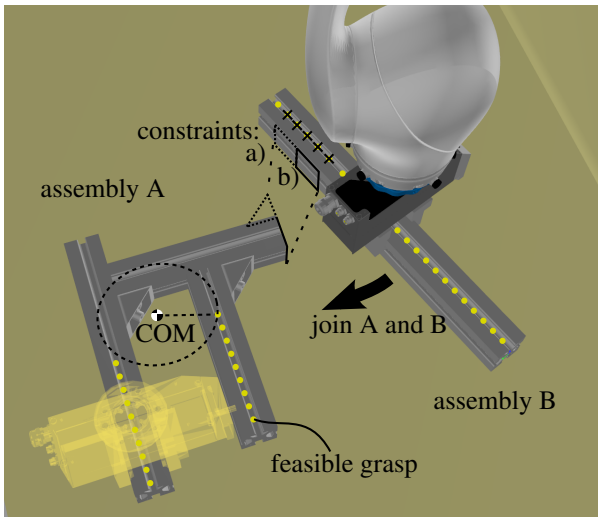


Figure 3: Grasp planning for assemblies with constraints. Feasible grasps are sampled for assembly A and sorted by their distance to the COM. Assembly B is grasped considering the constraints from the joining operation.

The grasp planning process is illustrated in Fig. 3. A large number of samples allows the planner to choose an appropriate grasp for different scenarios and assembly sequences. As the end-effector for this system is a 2-finger gripper, one point defines the position of the end-effector. For example, samples for possible grasp poses are taken along the main axes of the profiles. All the suggested grasping poses are good grasps as long as the object is isolated. When the part must be grasped to join a subassembly, all candidate grasps in the database have to be verified with respect to the constraints that might arise. The constraints are represented as restriction areas whose size and location depend on the nature of the constraint, and have two different sources: a) Constraints due to the subassembly (the space required for the angle-bracket that joins the parts must be free), and b) Constraints due to the joining action (the areas required for mating the parts must not be occupied by the gripper). Fig. 3 illustrates both types of constraints.

During the assembly planning process, described in the next section, two or more parts are combined to create subassemblies. As there is a previously generated grasp database for all the individual parts, the union of the individual grasp sets generates a pool of candidate grasps for the subassembly. A filtering process discards unfeasible grasps due to collision with other parts in the group. This strategy is conservative, in the sense that it does not generate new grasps for combined parts, but provides a sufficient number of possible grasps. The remaining candidate grasps are then evaluated and sorted according to a convenient robustness criterion. In the current system, the considered criterion is the distance of the grasping point to the center of mass (COM) of the subassembly; the smallest the distance to the COM, the lowest the torque on the end effector.

5 Assembly Sequence Planning

The aim of assembly sequence planner is to find a suitable sequence to assemble a product. This is, in general, an NP-hard problem [11]. Moreover, the geometric assembly planner must also cope with uncertainties in shape and poses of the parts. The proposed solution to deal with this problems is presented in this section.

5.1 Disassembly Analyzer

One common way to generate feasible sequences is to start with the entire product and sequentially remove single parts or subassemblies. For each pair of parts (p_i, p_j) from the assembly, the so-called disassembly maps are generated, based on a geometric approach [16]. Fig. 4 illustrates the disassembly algorithm, which follows these steps:

1. Decompose the surfaces of the objects into convex patches; here, let p_i consists of $S_i = \{s_1, s_2, \ldots s_p\}$ and p_j of $S_j = \{s_1, s_2, \ldots s_q\}$ convex patches.
2. In order to deal with uncertainties, the convex patches are shrinked according to a tolerance value.
3. The C-space obstacle of the active part p_i due to the passive part p_j is computed using the union of the Minkowski differences of the surface patches:
$$s_{i1} \ominus s_{j1} \cup s_{i1} \ominus s_{j2} \cup \ldots \cup s_{ip} \ominus s_{jq}.$$
4. The above steps lead to a mesh soup that could be united. However, this is an expensive calculation; instead, all the triangles are inserted in a voxel-space for fast ray-casting tests.

5. Each disassembly direction corresponds to a ray that intersects the triangle soup. If there is no collision, free motion is possible. All the directions are registered in a disassembly map.

For n parts, $\frac{1}{2}\ (n^2 - n)$ disassembly maps are obtained. The next step is the generation of the AND/OR-graph, a well known graph to efficiently store variants of sequences [9].

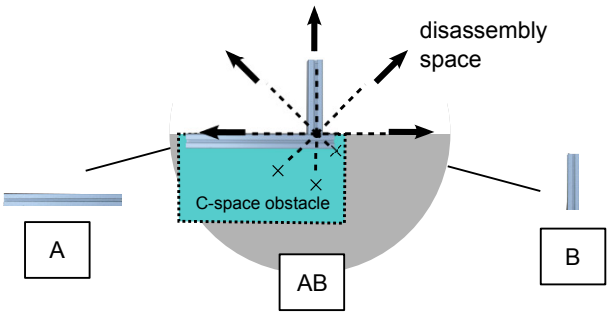


Figure 4: Calculation of disassembly maps with the Minkowski operation and ray casting tests. Profile B can be removed in any direction of the disassembly space.

5.2 Generation of the AND/OR-graph

The disassembly maps are used to efficiently determine the geometric feasibility during the generation of the AND/OR-graph. An excerpt of an AND/OR-graph is shown in Fig. 5, where the circles describe the disassembly maps, and white color in them corresponds to free disassembly directions. The feasibility is obtained by the union of the disassembly maps. The generation of the AND/OR-Graph uses the following algorithm [17]:

1. It starts with an assignment of standard priorities to single parts, and predefined priorities to subassemblies. The subassemblies and parts are sorted according their priorities.

2. The algorithm tries to cut the head from the tail (i.e. create two separate entities). If this is a valid cut, a new edge and two nodes are inserted into the AND/OR-graph, if they do not exist already.

3. With the knowledge of the cuts, the priorities are assigned again. For example, if the very last cut caused a screwing operation, then all screws in short distance to this screw obtain high priorities.

4. The queue is reordered according to the newly assigned priorities.

5. If a cut is not feasible, the priority of the subcomponents is assigned again, and the head is inserted at the appropriate position in the queue.

6. When the entire queue has been investigated, a shuffle method is called to reassign priorities such that the order of the queue can be adapted to a new situation.

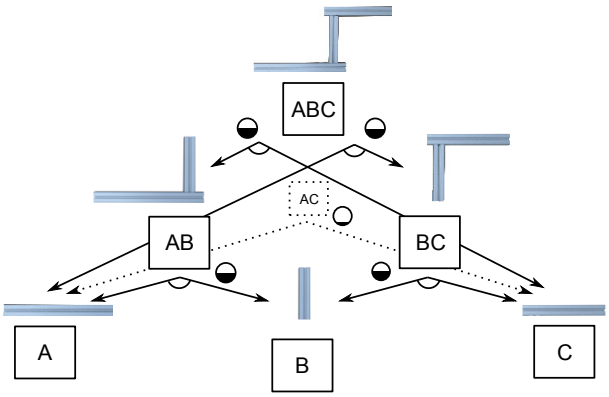


Figure 5: Excerpt of the AND/OR Graph for a particular example, where unconnected subassemblies are pruned.

Thus, priorities can be considered as a penalty function, which causes the algorithm to behave in a similar way to a Markov decision process. Note that the assembly sequence planner is so far fully deterministic; the only variation can be caused by objects with the same priority. The process is repeated until no change in the queue occurs. The process stops when a predefined number of sequences has been analyzed. After that, the AND/OR-graph keeps all possible sequences. The final assembly sequence is obtained after searching through the AND/OR-graph in a deterministic order, according to certain evaluation functions.

5.3 Evaluating the AND/OR-graph

The AND/OR-graph has to be evaluated in order to find a feasible assembly sequence. A major criterion is that each node holds at least one feasible grasp, nodes which do not own feasible grasp configurations are discarded [17]. The remaining AND/OR-graph is searched according to further criteria. Factors such as preferring mating actions with the lowest difficulty, or a minimization of the number of tool changes, are considered. Finally, a sequence is selected through a search in the AND/OR graph. Fig. 6 illustrates for a particular example the assembly tree generated by the assembly sequence planner, which specifies the task sequence.

6 Robotic Assembly Skills

The assembly system in an SME production has to provide robust robotic skills that are able to incorporate the uncertainties in the workcell and that are highly reusable for a wide set of assembly tasks. This section describes the skill-based execution engine and the skill library used in the proposed system.

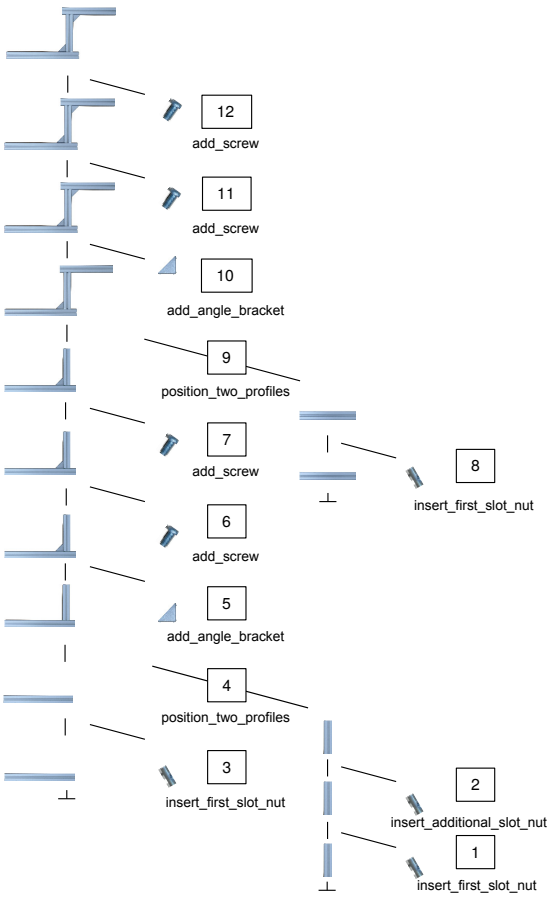


Figure 6: An automatically generated assembly sequence tree with its classification of tasks. In total, five different task types are identified by analyzing the pairs of assembly groups.

6.1 Skill Library

In the current work, robotic skills are considered as enhanced robot actions that use the device capabilities in order to solve certain types of tasks. They are built upon device primitive functions, and are instantiated through parameters that makes them adaptable for specific situations. For instance, a typical assembly task is the peg-in-hole task, which can be realized by a LWR or by a generic industrial manipulator. Both robots can solve the task, but will have different strategies and implementations. Nevertheless, the high level parameters given by the task specification are the same. Fig. 7 shows the levels of abstraction and the connected parameter types used in the presented assembly system.

On the lowest level there are skills that implement the basic capabilities of the system, such as a collision free motion of the robot arm in the workcell or a controller for doing a peg-in-hole insertion. At this level, parameters are spatial coordinates or controller values. Close to this level is an object level, in which direct actions on physical objects are implemented and parameters are on a symbolic level, e.g. available grasps that are symbolically annotated and referenced. These skills are logically composed on the higher abstraction levels, where the dependency on the devices disappears and parameters of the assembly process itself are explicitly stated. The final application level contains the interface to the assembly planner, where references to certain nodes in the assembly tree are given. A skill library structured according to this abstraction levels guarantees re-usability for different task sequences. Altogether, the implemented system uses about 25 skills.

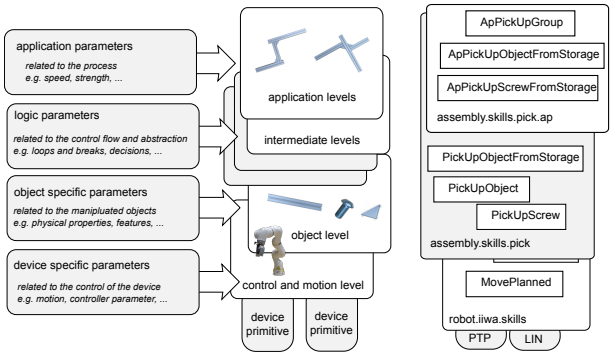


Figure 7: Hierarchical abstraction of skills and types of parameters.

6.2 Skill Integration

The assembly skills require information about the products and resources in the workcell. This knowledge should be accessible and updated directly by the robotic system without needing human assistance during the nominal execution. In the presented assembly system, a world representation [13] is used in order to keep track of properties and states of the parts and workcell. It consists of a database with general object properties filled with prior knowledge, e.g. from the product data-sheet, and a runtime model of object instances, where properties can be automatically updated according to the observations made by the robotic system. This knowledge is accessible by all system components through services on a symbolic level.

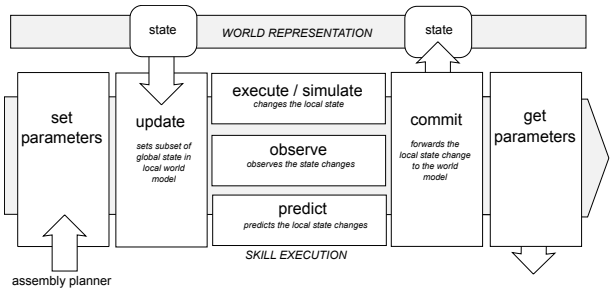


Figure 8: Conceptual execution flow of a skill; parametrization and synchronization of the local state to an external world representation.

In particular, the skills synchronize their state changes as depicted in Fig. 8, which shows a conceptual execution flow diagram. At the beginning, the skill is configured by the required parameters from an external source, i.e., the assembly planner or the user input. Then comes an

update step, where a local state of the affected objects is obtained, which is committed back to the world representation after the actual skill execution. The skill execution itself features various modes that interact in different ways with the local state and which can be executed in a parallel manner. The execute/simulate-mode implements the complete actions of the skills, either on the real system or on a virtual simulation. An observer mode is used to monitor the progress and evaluate the outcome of the execution, and a predict-mode is used to preview the (logical) outcome. If not only the state but also parameters are changed, e.g. after a performance optimization in the simulate-mode, parameters can be transferred back to the external source. The globally shared knowledge guarantees an easy integration of additional system components, and the various skill modes support a situation-dependent execution of the skill, e.g. the simulation mode is used to validate the generated assembly plan in the current workcell setup, while the observation-mode is important in an autonomous assembly system for error detection and handling.

6.3 Skill Robustness

In the autonomous assembly scenario, the robot system must reliably cope with uncertainties present in a typical SME environment. As classical assembly tasks are typically sequences of various contact states, it is especially important to deal with pose and contact uncertainties. Fig. 9 shows a collection of particular device capabilities of the LWR, enabled by its intrinsic torque-sensing [1], and which are used throughout the skill implementations to achieve the required robustness during the execution. The impedance controller allows the passive alignment with objects in the environment with inaccurately known poses by setting a virtual stiffness, as used for instance in the *PlaceObject*-skill; compliant search strategies actively reduce larger uncertainties, and oscillating force patterns decrease the risk of jamming in peg-in-hole tasks, e.g. in *PickUpScrew*- and *PlacePegInHole*-skill.

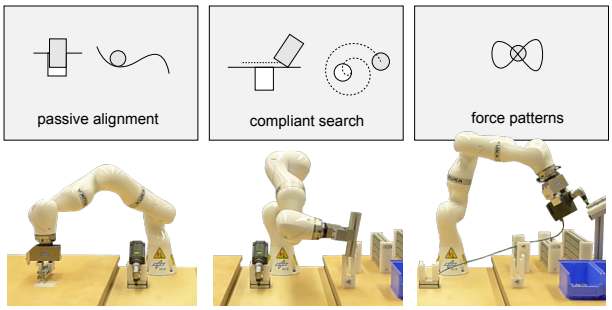


Figure 9: Manipulation capabilities of the LWR robot, enabled by its torque-sensing and impedance control mode, which are used to implement robust assembly skills, e.g. in the *PlaceObject*-, the *PlacePegInHole*- and the *PickUpScrew*-skill.

6.4 Mapping Tasks to Skills

The assembly of product variants requires an automatic mapping from specified assembly tasks in the generated sequence plan to skills that are available in the skill library. The reasoning about which skill to choose for a specific task is achieved here using a task classification approach. The concept requires the identification of specific patterns in the assembly sequence and its relation to a skill chart, which captures the knowledge of an expert. These patterns can incorporate properties and relations of the subassemblies from the current, the previous and the future assembly steps.

In the presented application, the pairs of subassemblies at each assembly step are classified into certain group types. The combinations of these group types determine then the type of assembly task. In the testbed there are five major tasks, e.g. inserting a slot nut in a profile or adding an angle bracket for joining the profiles. In order to automatically classify the tasks, the patterns, i.e. the combinations of group types, can be collected in a simple look-up-table where the rows and columns are specific group types and the cells of the matrix are task types, as illustrated in Fig. 10.

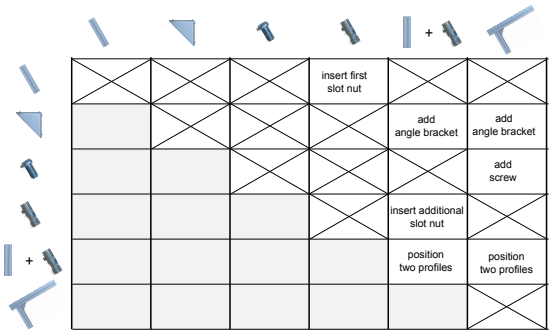


Figure 10: Task pattern matrix in which specific combinations of group types from the assembly sequence tree are related to assembly task types. Every cell represents a specific task type that is later mapped to an appropriate skill execution.

For example, if one step in the assembly sequence is a joining operation between a group with a single profile and a group with a single slot nut, the task *insert first slot nut* is selected. If in the next step another slot nut should be added, the task *insert additional slot nut* is selected. Each task type is mapped to a known skill sequence that can solve the specific task with the available devices. In this specific use case of a single robot arm and a given workcell layout, it is straightforward to have a one-to-one (injective) function, but nevertheless a generalization would allow having different mappings according to the target robot system. Fig. 11 shows for example the skill sequence in the testbed for solving the task *insert additional slot nut*. The complete task classification for an S-shaped structure is shown in Fig. 6.

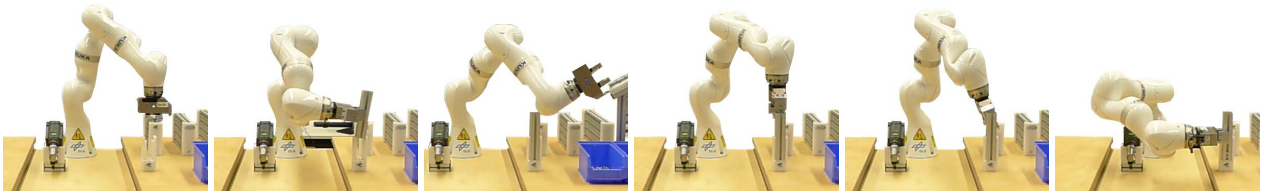


Figure 11: An example of skill sequence for the *insert first slot nut*-task, from left to right: *ApPickUpGroupFromStorage, ApPlacePegInHole, ApPickUpGroupFromStorage, ApPlaceSlotNutIntoProfile, ApMoveSlotNutInProfile, ApPickUpAndPlaceGroupInAssemblyFixture*.

7 System Evaluation

The presented assembly system is applicable to product families which are based on a modular construction design. Basic parts are tailored or combined according to individual requests of the customers.

7.1 Experimental Setup

The system is evaluated in an one-of-a-kind production scenario realized in the SMErobotics-testbed (Fig. 1), shown for the first time at AUTOMATICA 2014. A demonstration module allows to intuitively specify the desired assembly, as described in Section 3. In particular, the available basic parts for the assembly are a set of 40×40 aluminum profiles with various lengths. They can be joined together to create complex assemblies with angle brackets, button head M8×16 screws and slot nuts. CAD-data from the vendor is available for each single part, which is used during assembly planning and for the object representation in the run-time world model. The robotic workbench contains the robotic devices, part storage and primitive fixtures, as it might be available in a SME workshop. The available devices are a *KUKA LBR iiwa* arm, a *Schunk WSG 50* gripper, and a modified off-the-shelf power screwdriver, which can be grasped and used by the robot. There is one fixture in which a single profile can be inserted for the slot nut assembly. Another fixture on the assembly table prevents sliding of the parts during assembly operations.

7.2 Assembly Execution

Our robotic assembly system is capable of autonomously building a large variety of demonstrated structures. Fig. 12 shows some examples obtained with the system. The execution system can build profile connections with one- or two-sided angle bracket connections. Multiple profiles can be attached on one or two sides of a profile. A theoretical infinite number of structures can be assembled with this system.
As there is only a single robot arm in the setup, the assemblies need to be reoriented in the fixtures of the workcell. These operations are included in the skill implementations, and the system automatically decides the corresponding steps and computes the required grasps and paths. The simulation mode of the skills allows the precomputation of a sufficient number of assembly steps to validate the feasibility of the plan in the current state of the workcell, which can differ from the state during assembly sequencing.

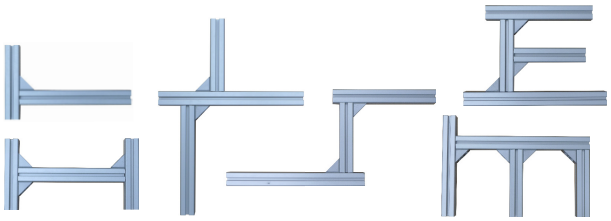


Figure 12: Examples of assemblies that were tested and successfully assembled.

A robust assembly process is achieved by using the impedance controller mode of the robot. Multiple runs showed that the main sources of failure are due to a not optimal workcell setup and to currently unobservable events. In order to increase the ability to detect errors, the integration of vision systems in the workcell is considered. The current strategy for solving intractable assembly steps, which can in certain cases be discovered in the simulation, is the assistance from a human worker.
The execution time of the test-bed has not been optimized, as the main objective was not increasing the execution speed in order to outperform a skilled human worker. In the current implementation, the execution time lies in the range of 5 minutes per joining operation. Instead, the setup demonstrates how a robotic system can autonomously solve different and complex assembly tasks with minimal human intervention, thus increasing the flexibility of a SME production facility. The time required from a client order for an individualized product to the final delivery with the proposed system is considered competitive, as no human labor is needed at this stage anymore.

8 Conclusions

This paper describes a complete system that demonstrates how variations of a relatively complex assembly task can be automatically solved. It demonstrates how the robot can create new assemblies of aluminum profiles without the intervention of a robotic expert, thus meeting the requirements of a SME production facility. A layman is able to show the desired product without needing any knowledge about the robotic system. The rec-

ognized assembly is automatically transferred to the assembly sequence planner; using CAD data as direct input is also possible. The system autonomously generates the code required to produce the desired assembly, considering the feasibility of the assembly sequence, reachability of the parts and robust and feasible grasping of the parts. A novel reasoning system to map assembly tasks to executable robot skills allows the direct generation of robot executable code. A skill execution engine was developed for dealing with uncertainties; it adapts to the current state of the work cell as it is connected to a world representation, and provides methods to presimulate and observe the assembly progress.
Altogether, the fully automated assembly system is capable of building new assemblies from combinations of basic parts. The developed system can be easily reconfigured for generating new product families. Current limitations of the system are due to the work cell setup, e.g. location of the fixturing units. In the future, the system will be extended to deal with more complex assemblies, e.g. 3D constructions. Finally, the efficiency of the automated workflow will be compared to manually programmed systems to prove its competitiveness for high variation and low volume productions.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 287787, project SMErobotics. Furthermore, the authors want to thank Marinus Danzer, Christian Scheurer and Uwe Zimmermann from KUKA AG for providing the collision free motion planner used in this work.

References

[1] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, G. Hirzinger: *The DLR Lightweight Robot: Design and Control Concepts for Robots in Human Environments*, Industrial Robot, Vol. 34(5), Emerald Group, 2007

[2] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, G. Hirzinger: *The KUKA-DLR Lightweight Robot Arm*, Proc. of the 41st Int. Symposium on Robotics (ISR) 2010 and 6th German Conf. on Robotics (ROBOTIK), 2010

[3] J. Bohg, A. Morales, T. Asfour, D. Kragic: *Data-Driven Grasp Synthesis - A Survey*, IEEE Trans. Robotics, Vol. 30(2), 2014

[4] A. Bourjault: *Methodology of Assembly Automation: A New Approach*, Springer-Verlag, 1987

[5] T. Dietz, U. Schneider, M. Barho, S. Oberer-Treitz, M. Drust, R. Hollmann, M. Haegele: *Programming System for Efficient Use of Industrial Robots for Deburring in SME Environments*, Proc. 7th German Conf. on Robotics (ROBOTIK), 2012

[6] S. Haddadin, M. Suppa, S. Fuchs, T. Bodenmüller, A. Albu-Schäffer, G. Hirzinger: *Towards the Robotic Co-Worker*, Robotics Research, Springer Tracts in Advanced Robotics, Vol. 70, Springer-Verlag, 2011

[7] D. Halperin, J. Latombe, R. Wilson: *A General Framework for Assembly Planning: The Motion Space Approach*, Algorithmica, Vol. 26(3), Springer-Verlag, 2000

[8] T. Hasegawa, T. Suehiro, K. Takase: *A Model-Based Manipulation System with Skill-Based Execution*, IEEE Trans. Robotics and Automation, Vol. 8(5), 1992

[9] L. Homem de Mello, A. Sanderson: *AND/OR Graph Representation of Assembly Plans*, IEEE Trans. Robotics and Automation, Vol. 6(2), 1990

[10] S. Kaufman, R. Wilson, R. Jones, T. Calton, A. Ames: *The Archimedes 2 Mechanical Assembly Planning System*, Proc. IEEE Int. Conf. Robotics and Automation, 1996

[11] L. E. Kavraki, J.-C. Latombe, R. H. Wilson: *On the Complexity of Assembly Partitioning*, Information Processing Letters, Vol. 48(5), 1993

[12] T. Kröger, B. Finkemeyer, U. Thomas, F. Wahl: *Compliant Motion Programming: The Task Frame Formalism Revisited*, Mechatronics and Robotics, 2004

[13] D. Leidner, C. Borst, G. Hirzinger: *Things Are Made for What They Are: Solving Manipulation Tasks by Using Functional Object Classes*, Proc. IEEE/RAS Int. Conf. Humanoid Robots, 2012

[14] M. A. Roa, M. Argus, D. Leidner, C. Borst, G. Hirzinger: *Power Grasp Planning for Anthropomorphic Robot Hands*, Proc. IEEE Int. Conf. Robotics and Automation, 2012

[15] A. Stemmer, A. Albu-Schaeffer, G. Hirzinger: *An Analytical Method for the Planning of Robust Assembly Tasks of Complex Shaped Planar Parts*, Proc. IEEE Int. Conf. on Robotics and Automation, 2007

[16] U. Thomas, M. Barrenscheen, F. Wahl: *Efficient assembly sequence planning using stereographical projections of C-space obstacles*, Proc. IEEE Int. Symp. Assembly and Task Planning, 2003

[17] U. Thomas, T. Stouraitis, M. A. Roa: *Flexible Assembly through Integrated Assembly Sequence Planning and Grasp Planning* Proc. IEEE Int. Conf. on Automation Science and Engineering, Gothenburg, 2015

[18] I. Weidauer, D. Kubus, F. Wahl: *A Hierarchical Extension of Manipulation Primitives and its Integration into a Robot Control Architecture*, Proc. IEEE Int. Conf. Robotics and Automation, 2014

Publication 2

K. Nottensteiner, M. Sagardia, A. Stemmer, and C. Borst (2016): "*Narrow Passage Sampling in the Observation of Robotic Assembly Tasks*". IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016.

Version Note

The following attached version corresponds to the accepted manuscript of the publication.

The final published version is available under:

- https://ieeexplore.ieee.org/document/7487125.

Please refer to the final published version for citation:

```
@InProceedings{Nottensteiner2016b,
  author =      {Nottensteiner, Korbinian and Sagardia, Mikel and
                 Stemmer, Andreas and Borst, Christoph},
  title =       {Narrow Passage Sampling in the Observation of
                 Robotic Assembly Tasks},
  booktitle =   {2016 IEEE International Conference on Robotics and
                 Automation (ICRA)},
  year =        2016,
  venue =       {Stockholm, Sweden},
  eventdate =   {2016-05-16/2016-05-21},
  publisher =   {IEEE},
  pages =       {130-137},
  isbn =        {978-1467380263},
  eid =         {2-s2.0-84977494035},
  doi =         {10.1109/ICRA.2016.7487125}}
```

Copyright Note

Eratum

- In equation (8) read $\boldsymbol{H}_k{}^{(i)} \in \tilde{\mathcal{C}} \cup \partial\tilde{\mathcal{C}}$ instead of $\boldsymbol{H}_k{}^{(i)} \in \tilde{\mathcal{C}}$.

Narrow Passage Sampling in the Observation of Robotic Assembly Tasks

Korbinian Nottensteiner[1] , Mikel Sagardia[1], Andreas Stemmer[1], and Christoph Borst[2]

***Abstract*— The observation of robotic assembly tasks is required as feedback for decisions and adaption of the task execution on the current situation. A sequential Monte Carlo observation algorithm is proposed, which uses a fast and accurate collision detection algorithm as a reference model for the contacts between complex shaped parts. The main contribution of the paper is the extension of the classic random motion model in the propagation step with sampling methods known from the domain of probabilistic roadmap planning in order to increase the sample density in narrow passages of the configuration space. As a result, the observation performance can be improved and a risk of sample impoverishment reduced. Experimental validation is provided for a peg-in-hole task executed by a lightweight-robot arm equipped with joint torque sensors.**

I. Introduction

Typical assembly processes consist of sequences of various contacts transferring parts from free configuration space to constrained goal configurations. In robotic assembly execution (Fig. 1), the observation of such processes is required to detect and handle errors, adapt action parameters and robot controllers, or even to learn strategies and perform high-level reasoning. A major goal of assembly observation is to provide information about partially hidden states, e.g. geometric uncertainties or contact states. In recent years, especially sequential Monte Carlo (SMC) approaches showed their advantages in the observation of robotic assembly processes. The SMC framework allows not only to deal with non-linear models, but also with non-Gaussian and multi-modal distributions, cf. [1] and [2], which are obviously present when multiple contact states are possible. Within this probabilistic framework, state estimates are obtained by the sequential propagation, weighting and selection of a set of samples according to their consistency with the sensory information and the assumed system and measurement models.

In particular, Chhatpar and Branicky [3] propose a probing based localization method and demonstrate it in a lock-key assembly. Hypotheses of relative poses are initially spread in a pre-generated map of the configuration space and further updated by subsequent probes until the samples converge, i.e. the lock is localized. Thomas et al. [4] also apply pre-generated maps to localize the parts in an assembly task. The so-called force-torque maps contain additional information about expected contact forces and torques, which is used as further input in the weighting of the samples. A drawback of the pre-generated maps is the high preparation effort needed to ensure completeness in a sufficient level of detail.

[1] Institute of Robotics and Mechatronics (RMC-RM), German Aerospace Center (DLR), 82234 Wessling, Germany
[2] KUKA Roboter GmbH, 86165 Augsburg, Germany
Contact: korbinian.nottensteiner@dlr.de

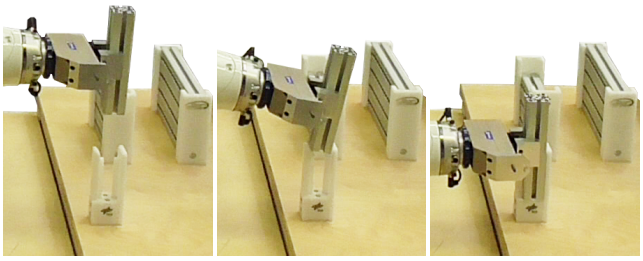


Fig. 1. Peg-in-hole insertion of an aluminum profile into an assembly fixture with a LWR robot arm. The peg is tilted, aligned in contact and inserted with impedance control.

Further SMC approaches use contact state graphs [5], which add a higher semantic layer to the observation by partitioning the configuration space into distinct regions, so-called contact formations. The knowledge of the properties and the transitions between certain contact formations can then be used to improve the performance of the observation algorithm [6] or resolve ambiguous force and torque measurements [7]. While these methods perform well, it is not clear how they scale to realistic examples, as they are only demonstrated for simple geometrical objects with a few possible contact states.

A general issue of the SMC methods is the number of samples required for reliable observation results, what consequently becomes a question of computational resources for high state dimensions and complex models. Taguchi et al. [8] reduce the number of samples significantly by the help of a Rao-Blackwellized particle filter (RBPF) that combines a sample based position estimation with a Kalman filter for orientation uncertainties. The additional usage of vision systems to initialize the sample distribution [9] or the fusion of tactile and vision measurements, [4] and [10], can also help to improve the observation performance. Although progress has been made, problems remain in cases with highly constrained configurations, as also concluded in [10]. Especially the transitions from free space into narrow passages in the configuration space, as in peg-in-hole assembly tasks, are still challenging with respect to sample impoverishment. This particular effect is characterized by the selection of only a few high-weighted samples in the resampling step. In the further observation, the samples might then not be sufficient to approximate the full posterior distribution and thus lead to strong biases in the estimates.

Avoiding the narrow passage during the localization, which is suggested by [3] and [8] by probing before the insertion, is not always feasible considering practical constraints

in real applications, and focuses not on the observation of the actual assembly process. Therefore, in the presented work, the particular cases of transitions into narrow passages before the final convergence of the sample set are considered, and two alternative sampling policies are compared with the commonly used random motion model. Both policies are rooted in probabilistic roadmap planning (PRM, [11]) and are used here for the first time in the context of sampling based observation of assembly processes. The Gaussian sampler [12] allows the generation of higher sample densities near the border of the configuration space. The random bridge builder (RBB), presented by [13], increases the sample density especially in narrow passages. By increasing the density in these critical regions a reduced risk of sample impoverishment and a higher performance is expected.

The sampling policies are adapted and integrated directly in the propagation model of the SMC algorithm. Here, the configuration space is sampled locally around each contact hypothesis and thus only the relevant space is considered and no pre-generated maps are required. As a contact model, the voxelmap-pointshell algorithm (VPS, [14]) is used, which is a penalty based collision algorithm, commonly used in haptic rendering for the computation of contact forces. Our fast and generic implementation [15] of the VPS is able to handle complex geometries given as CAD data. The developed observation framework is experimentally validated with sensory information of a lightweight-robot arm (LWR) executing a peg-in-hole task with complex shaped aluminum parts. The measurements include the current joint position and torques and therefore allow for the observation of contact states. The LWR is able to robustly execute such peg-in-hole tasks by the help of the impedance control mode [16]. But, on a higher abstraction level, these assembly strategies are feed-forward controllers, which do not directly incorporate the current state of the execution and thus are not capable of adapting the strategy to the current situation. This motivates the presented observer algorithm that monitors the assembly progress and provides state information for high level controllers and reasoning systems in future applications that increase the overall performance of the assembly system.

The work is presented as following, the applied robot and contact models are presented in Section II, together with the observation framework, and followed by the description of the sampling policies in the propagation step in Section III. Simulations and experimental results are provided for a peg-in-hole insertion in Section IV. Finally, the results are discussed in Section V and concluded in Section VI.

II. Models for Contact Observation

In contact tasks, the uncertainties from the robot arm combine with further uncertainties in the kinematic chain, which is closed in the contact between an environment object (EO) and a manipulated object (MO). In this section, the robot and uncertainty models are presented, as well as the contact models used for the observation of the assembly task. An overview of the observation framework is given, for which different sampling policies are investigated later.

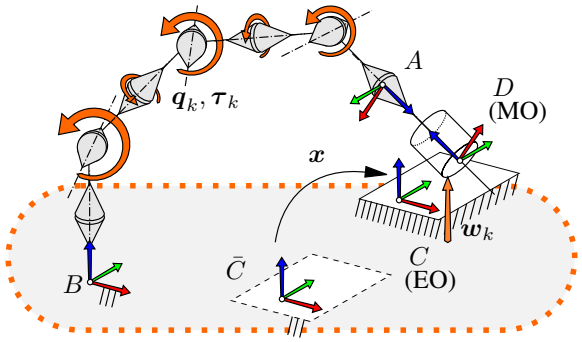


Fig. 2. Kinematic chain consisting of the robot flange A, the base B, the initial reference frame $\bar{C}$ and the frame C of the EO, and the frame D of the MO. The geometric uncertainties are summarized in the state $\boldsymbol{x}$ as parameter in the transformation $\boldsymbol{H}_{\bar{C}C}$. The contact wrench $\boldsymbol{w}_k$ induces a torque $\boldsymbol{\tau}_k$ in the robot joints; together with the current robot configuration $\boldsymbol{q}_k$ the state $\boldsymbol{x}$ is estimated.

A. Robot and Uncertainty Model

A general formalism on how to model the uncertainties in a kinematic chain can be found in [17]. As the observation of the assembly task is in focus of this work, not the identification of individual uncertainties is of interest, but rather the cumulative relative uncertainty between MO and EO, where the kinematic chain will be closed during the assembly. Exemplary in this work, the state $\boldsymbol{x} \in \mathbb{R}^m$ shall therefore be defined as geometric uncertainty in the transformation $\boldsymbol{H}_{\bar{C}C}(\boldsymbol{x}) \in SE(3)$ between an initial reference pose $\bar{C}$ and the actual pose C of the EO. It is assumed that the EO and MO are statically fixed to the environment and the robot, respectively. In the nominal case, $\boldsymbol{x}$ contains $m \leq 6$ constant, but unknown parameters affecting positions and orientations. In the real case, the specific source of uncertainty vanishes within the state $\boldsymbol{x}$, i.e. inaccurately known object poses or model errors are not separable without adding further error models in the relevant parts of the kinematic chain. The here chosen approach of a cumulative uncertainty is reasonable as long as unmodelled effects in the remaining kinematic chain are bound on a small scale. Note also that the presence of unmodelled effects, e.g. a not completely fixed EO, can produce a time varying state. Therefore, the state value at a specific instant of time k will be denoted as $\boldsymbol{x}_k$.

A rigid body model of the robot arm is assumed. The forward kinematics are then given by the transformation function $\boldsymbol{H}_{BA}(\boldsymbol{q}) \in SE(3)$ with the joint configuration $\boldsymbol{q} \in \mathbb{R}^n$, where B represents the base and A the flange of a robot arm with $n \geq 6$ degrees of freedom (Fig. 2). The MO is attached to the robot flange with a known and fixed transformation $\boldsymbol{H}_{AD} \in SE(3)$, where D is the coordinate system of the MO. At time k, the relative pose $\boldsymbol{H}_k := \boldsymbol{H}_{CD} \in SE(3)$ between MO and EO is obtained for a given state $\boldsymbol{x}_k$ and the current joint configuration $\boldsymbol{q}_k$ from the closed kinematic chain by

$$\boldsymbol{H}_k(\boldsymbol{q}_k, \boldsymbol{x}_k) = \boldsymbol{H}_{C\bar{C}}(\boldsymbol{x}_k)\boldsymbol{H}_{\bar{C}B}\boldsymbol{H}_{BA}(\boldsymbol{q}_k)\boldsymbol{H}_{AD}, \quad (1)$$

where $\boldsymbol{H}_{\bar{C}B}$ is given by the inverse of the initially assumed

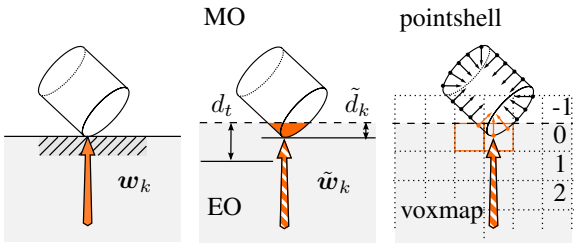


Fig. 3. (a) Real contact situation. (b) Contact model. (c) Implementation of contact model with VPS. $\tilde{d}_k$ denotes the contact distance and d_t a threshold on the maximal feasible virtual penetration, $\tilde{\boldsymbol{w}}_k$ the virtual contact wrench.

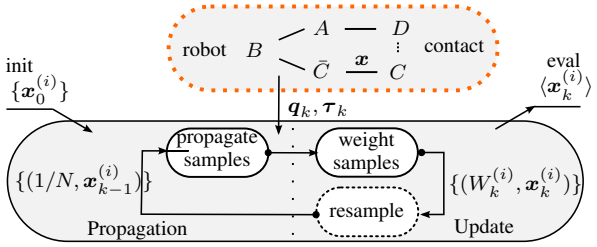


Fig. 4. The observation cycle of the basic SMC algorithm applied for assembly observation. Hypotheses $\{\boldsymbol{x}_k{}^{(i)}\}$ with $i = 1, ..., N$ of the state $\boldsymbol{x}$ in the kinematic chain of the robot, are propagated, then weighted according their consistency with the measurements and finally resampled.

pose $\boldsymbol{H}_{B\bar{C}} \in SE(3)$ of the EO relative to B, see Fig. 2.

The considered robot arms are equipped with torque sensors in their joints. When MO and EO are in contact, a torque $\boldsymbol{\tau}_k \in \mathbb{R}^{\mathrm{n}}$ can be obtained from the intrinsic sensors, which is dependent on the robot configuration by the well known relation:

$$\boldsymbol{\tau}_k = \boldsymbol{J}_k^T \boldsymbol{w}_k, \tag{2}$$

where $\boldsymbol{J}_k := \boldsymbol{J}_{BD}^D(\boldsymbol{q}_k) \in \mathbb{R}^{6 \times \mathrm{n}}$ is the Jacobian of the robot arm with respect to D and the external wrench $\boldsymbol{w}_k :=_D \boldsymbol{w}_k = (\boldsymbol{F}_k{}^{\mathrm{T}}, \boldsymbol{M}_k)^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R})^6$ acting on the MO at D.

B. Virtual Contact Model

A contact model is needed as a reference model in the observation process. It connects a state value $\boldsymbol{x}_k$ and the joint measurement $\boldsymbol{q}_k$ to a hypothetical contact situation, which provides information about the expected measured torques. The model is based on our improved implementation [15] of VPS [14]. The basics of VPS are visualized in Fig. 3 for a simplified contact situation. A cylindrical object is pressed on a flat surface, which induces a wrench $\boldsymbol{w}_k$ in the contact. The virtual model represents the wrench in form of the force $\tilde{\boldsymbol{w}}_k$, which acts on the intersecting volume of the virtual representations of the objects, and is in its physical meaning comparable to the buoyant force [18]. While this is a simplification of the possible effects in a real contact, the directions of the virtual wrenches provide reasonable information for the comparison of various contact hypotheses. The VPS implementation approximates the force by the help of so-called pointshells and voxmaps, which are, respectively, discretizations of the geometry in surface points including surface normals and cubic volume elements which conform signed distance fields. These data structures are generated from CAD for arbitrary complex parts.

Dependent on the relative pose $\boldsymbol{H}_k$ of the objects, the virtual contact model provides the contact forces and torques $\tilde{\boldsymbol{w}}_k(\boldsymbol{H}_k) = (\tilde{\boldsymbol{F}}_k^T, \tilde{\boldsymbol{M}}_k^T)^T \in \mathbb{R}^6$, and additionally the contact distance between the virtual objects $\tilde{d}_k = \tilde{d}(\boldsymbol{H}_k) \in \mathbb{R}$. The contact distance defines implicitly the relative configuration space $\tilde{\mathcal{C}}$ between the virtual representations of the MO and EO:

$$\begin{cases} \textit{no contact } (\boldsymbol{H}_k \in \tilde{\mathcal{C}}): & \tilde{d}_k < 0 \\ \textit{contact } (\boldsymbol{H}_k \in \partial\tilde{\mathcal{C}}): & 0 \leq \tilde{d}_k < d_t \\ \textit{invalid } (\boldsymbol{H}_k \notin \tilde{\mathcal{C}}): & d_t < \tilde{d}_k, \end{cases} \tag{3}$$

where $d_t > 0$ is a threshold on the maximal feasible virtual penetration. As no additional force/torque-measurement device will be used, $\tilde{\boldsymbol{w}}_k$ is transformed to a virtual contact torque $\tilde{\boldsymbol{\tau}}_k$ according to Eq. (2).

C. Framework for Contact Task Observation

At each time step k the robot provides the measurements $\boldsymbol{y}_k = (\boldsymbol{q}_k, \boldsymbol{\tau}_k)$. The observation model uses this sensory information together with the contact model in order to estimate the current state value $\boldsymbol{x}_k$. Formulated in terms of Bayesian estimation, the objective is to infer the current state distribution density $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$ given the set of past measurements $\boldsymbol{y}_{0:k} = \{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_k\}$. Changes in the contact state are in general nonlinear and the distributions of the uncertain state are not completely describable by methods based solely on Gaussian distributions. Therefore, the observation is carried out with a SMC approach, which can deal with nonlinear and non-Gaussian systems, compare [1] and [2] for a general introduction. A main idea of SMC is that probability densities, like the posterior density $p\,(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$, are approximated by a set of samples or so-called particles instead of using continuous model functions. The general procedure of the applied SMC algorithm for assembly observation is visualized in Fig. 4.

The relevant uncertainty space is initially approximated by a set of N samples $\{\boldsymbol{x}_0{}^{(\mathrm{i})}\}$, followed by sequential processing of the set. The basic SMC algorithm for filtering, as it is described by [2], consists of a sampling or propagation step, a weighting and a resampling step. In the first step, samples are drawn from an importance distribution $q(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{x}_{k-1}{}^{(\mathrm{i})})$. A commonly used importance model for the estimation of geometric uncertainties assumes a normal distributed random diffusion of the samples, e.g. [3] and [8],

$$q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \Sigma_{\boldsymbol{x}_k}), \tag{4}$$

with the covariance matrix $\Sigma_{\boldsymbol{x}_k}$. In stiff and accurate kinematic systems, the geometric uncertainties are often rather constant parameter values and the sampling step produces mainly artificial noise in order to avoid particle impoverishment [8]. This random diffusion model will be compared to two further propagation models, which will be presented in the next section in more detail.

The second step is the weighting step, in which the samples obtain weights $W_k{}^{(\mathrm{i})}$ according to their consistency

with the observation

$$W_k^{(i)} \propto \frac{g(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}) f(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})}{q(\boldsymbol{x}_k^{(i)}|y_k, \boldsymbol{x}_{k-1}^{(i)})}. \tag{5}$$

Here, $g(\boldsymbol{y}_k|\boldsymbol{x}_k)$ is the observation density and $f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ the transition density of the (hidden) system dynamics. Choosing $q(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{x}_{k-1}) = f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ leads to the classic weight computation proposed in the bootstrap filter, compare [1]:

$$W_k^{(i)} \propto g(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}). \tag{6}$$

In the presented approach, the observation density distribution consists of two factors. First, the measured torques $\boldsymbol{\tau}_k$ are compared to the virtual contact torques provided by the VPS algorithm assuming normal distributed errors in the measurements:

$$p_{\boldsymbol{\tau}}(\boldsymbol{\tau}_k|\boldsymbol{x}_k^{(i)}) = \mathcal{N}(\boldsymbol{\tau}_k|\tilde{\boldsymbol{\tau}}_k^{(i)}, \Sigma_{\boldsymbol{\tau}}). \tag{7}$$

Compared to other approaches, where the residuum is calculated in the wrench space, cf. [4] and [6], a formulation in joint space is chosen in order to directly refer to the errors in the joint torque measurements and in order to avoid the nonuniform metrics of the wrench space. Second, the contact distance is used to filter out the impossible configurations that violate the relative configuration space between the objects:

$$p_{\boldsymbol{q}}(\boldsymbol{q}_k|\boldsymbol{x}_k^{(i)}) = (\sigma_d\sqrt{2\pi})^{-1}.\begin{cases} 1, & \boldsymbol{H}_k^{(i)} \in \tilde{\mathcal{C}} \\ \exp(-\frac{(\tilde{d}_k - d_t)^2}{(2\sigma_d^2)}), & \boldsymbol{H}_k^{(i)} \notin \tilde{\mathcal{C}}. \end{cases} \tag{8}$$

This density ensures that the virtual objects stay in the valid configuration space given by the threshold d_t on the virtual contact distance $\tilde{d}_k$. The usage of a contact distance in the update step is also described by [6], where combinations of relevant elemental contacts for a given discrete contact state are evaluated. [9] proposes a combination of distance measures in the inhand-localization of objects with robotic hands, which incorporates the information from tactile sensing in a further zero-mean Gaussian distance density. In contrast, the tactile information is explicitly considered in the torque residuum Eq. (7) above. Thus, also the direction of the contact forces are evaluated and certain contact states can be distinguished, which is important for the convergence of the filter in the peg-in-hole task. Under the assumption of independence, the total observation density is then given by the product:

$$g(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}) = p_{\boldsymbol{q}}(\boldsymbol{q}_k|\boldsymbol{x}_k^{(i)}) \ \cdot \ p_{\boldsymbol{\tau}}(\boldsymbol{\tau}_k|\boldsymbol{x}_k^{(i)}). \tag{9}$$

Finally, the weights are used in the resampling step in order to select best fitting hypotheses with a higher probability. Expected values of a function $V(\boldsymbol{x}_k)$ can be approximated by the evaluation of the particle distribution:

$$\langle V_k \rangle \approx \sum_{i=1}^{N} W_k^{(i)} V(\boldsymbol{x}_k^{(i)}), \tag{10}$$

which can be used to indicate the current progress of the assembly process and to infer further information from the current state. Note that after resampling, the weights are set to $W_k{}^{(\mathrm{i})} = 1/N$.

III. Sampling Policies

In this section alternative sampling policies for the propagation step of the observer are introduced. They are known from PRM and increase the sample density at the border and in narrow passages of the configuration space $\mathcal{C}$. In the context of contact task observation, it is expected that they decrease the number of required samples for convergence of the estimates and reduce the effects of sample impoverishment.

A. Pure Diffusion Sampling

The first policy is based on the random diffusion model, which was already presented shortly in the previous section, see Eq. (4). It is a default model for unknown or uncertain state dynamics. The samples are propagated according to a given covariance matrix $\Sigma_{\boldsymbol{x},p}$:

function PureDiffusion($\boldsymbol{x}_{k-1}{}^{(\mathrm{i})}$)
  $\boldsymbol{x} \leftarrow \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}{}^{(\mathrm{i})}, \Sigma_{\boldsymbol{x},p})$.
  return $\boldsymbol{x}$

If no joint torque is measurable, then the samples will evolve in a diffusion process governed by $\Sigma_{\boldsymbol{x},p}$. In this sampling model, no contact information is evaluated with the advantage of a fast implementation. The diffusion is limited in the constrained regions through the observation density given by Eq. (8) in the update step of the observer.

B. Gaussian Sampler

The Gaussian sampler is known from robot motion planning with PRM [12]. It concentrates samples close to $\partial\mathcal{C}$ by selecting only samples for which a second sample outside of $\mathcal{C}$ can be found within a certain distance. In the here presented work, the method is adapted to concentrate the samples in $\partial\tilde{\mathcal{C}}$ as defined in (3). Therefore, the function EvaluateContact tests whether a sample with state $\boldsymbol{x}$ is within the configuration space of the virtual contact model:

function EvaluateContact($\boldsymbol{x}$, $\boldsymbol{q}_k$)
  if $d_t < \tilde{d}(\boldsymbol{x}, \boldsymbol{q}_k)$ **then**
    return $invalid$
  else if $0 \leq \tilde{d}(\boldsymbol{x}, \boldsymbol{q}_k) < d_t$ **then**
    return $contact$
  else if $\tilde{d}(\boldsymbol{x}, \boldsymbol{q}_k) < 0$ **then**
    return $no\ contact$

The return value is used to decide on the propagation strategy. If a sample A is already in $contact$ it will be propagated according to the PureDiffusion-function above. If the sample A is not in $contact$, a second sample B is drawn according to a distance measure. This is repeated until the locally sampled B is in $contact$ or a maximal number of trials L_{max} is reached. This sampling policy is applied for every hypothesis $\boldsymbol{x}_{k-1}{}^{(\mathrm{i})}$ and is summarized as:

function GaussianSampler($\boldsymbol{x}_{k-1}{}^{(\mathrm{i})}$, $\boldsymbol{q}_k$)
  $\boldsymbol{x}_A := \boldsymbol{x}_{k-1}{}^{(\mathrm{i})}$
  $A \leftarrow$ EvaluateContact($\boldsymbol{x}_A, \boldsymbol{q}_k$)
  if $A \neq contact$ **then**
    for j := 1 to L_{max} **do**
      $\boldsymbol{x}_B \leftarrow \mathcal{N}(\boldsymbol{x}_B|\boldsymbol{x}_{k-1}{}^{(\mathrm{i})}, \Sigma_{\boldsymbol{x},g})$.

```
        B ← EVALUATECONTACT(x_B, q_k)
        if B = contact then
            return x_B
    return PUREDIFFUSION(x_A)
```

In contrast to the classical policy of [12], the samples A are not discarded completely in the case that the collision conditions are not satisfied, but rather resampled locally. L_{max} controls the effort in order to find a solution and finally how dense the border will be filled with samples. For $L_{max} \rightarrow \infty$ all samples will be concentrated in $\partial\tilde{\mathcal{C}}$; for small values of L_{max} the pure diffusion dominates the sample propagation. The parameter $\Sigma_{\boldsymbol{x},g}$ controls the search distance in the local resampling loop.

C. Bridge Test Sampling

The following policy is also motivated by PRM. The RBB [13] favors samples in narrow passages of the configuration space. The method is also adopted for the propagation context. Here again, a second sample B is created for a initial hypothesis A:

```
function BRIDGETEST(x_{k-1}^(i), q_k)
    x_A := x_{k-1}^(i)
    for j := 1 to L_max do
        x_B ← N(x_B | x_A, Σ_{x,b}).
        B ← EVALUATECONTACT(x_B, q_k)
        if B = invalid then
            x_C ← (x_A + x_B)/2
            C ← EVALUATECONTACT(x_C, q_k)
            if C ≠ invalid then
                return x_C
    return PUREDIFFUSION(x_A)
```

In the case B is $invalid$, a bridge point C will be created which is located at the half distance between A and B. If this bridge point is valid, i.e. $contact$ or $no\ contact$, it will be propagated as the new hypothesis. This pulls samples into the narrow passages of the local configuration space. The covariance matrix $\Sigma_{\boldsymbol{x},b}$ is related to the gap size of the expected narrow passages. Here, a Gaussian density is chosen. As the authors of [13] note, if prior knowledge about the narrow passage is available, a more suitable function could be used. Similarly to the GAUSSIANSAMPLER-function a higher number of maximal iterations L_{max} increases the admissible effort, but also the density in the narrow passage.

IV. EXPERIMENTS AND RESULTS

The three sampling policies are validated based on measurements obtained from the KUKA LBR iiwa robot, which is the new industrial version of the LWR developed by [19]. The observation is carried out for a peg-in-hole task, where an aluminum profile is inserted in an assembly fixture (Fig. 1). The profile is tilted, aligned and inserted with enabled impedance control mode, which allows implementing a robust assembly strategy. Nevertheless, an observation of the process is requested in flexible production facilities, as the workcell setup is not guaranteed to be fixed and high-level monitoring and decisions are desired.

A. Experimental Setup

The peg-in-hole task is part of a larger assembly testbed previously shown at AUTOMATICA 2014[1]. The MO is a 200 mm long 40 mm × 40 mm aluminum profile of the *item MB Building Kit System*[2] and the EO is the assembly fixture, which provides form-closure with the longitudinal slots of the profile at two sides. The depth of the fixture is 100 mm, the clearance is below 1 mm. The voxmap of the EO in the VPS has a resolution of 1 mm, and the pointshell of the MO contains in total 5666 points. A world model [20] provides the assumed pose of the fixture relative to the robot base $\boldsymbol{H}_{B\bar{C}}$ and the grasp transformation $\boldsymbol{H}_{AD}$. The desired goal pose at the bottom of the fixture is given by the nominal transformation $\boldsymbol{H}_{CD,goal}$. Measurements of the current joint pose $\boldsymbol{q}_k$ and the torque $\boldsymbol{\tau}_k$ are taken at a rate of 33 Hz and evaluated offline in order to compare the effects of the sampling policies.

B. Simulated Sample Evolution at Narrow Passage

Simulations are carried out in order to validate the spread of the samples. Therefore, the relative transformation $\boldsymbol{H}_{CD}$ is kept constant at the entrance of the narrow passage at ${}_{C}x_{CD} = 95$ mm (Fig. 5). In this configuration, no torque is measured and the samples mainly evolve according to the propagation model and the boundaries of the configuration space. In the given example, the threshold on the maximal penetration is set to $d_t = 2$ mm and the number of allowed iterations is $L_{max} = 5$. Position uncertainties $\boldsymbol{x} = {}_{C}(x, y, z)_{\bar{C}C}$ of the fixture relative to its assumed pose are considered here to show the basic results. The initial samples are uniformly drawn from a cubic uncertain volume around $\bar{C}$ with a side length of 20 mm and a shifted mean of one third of the dimensions. The covariance matrices are assumed to be diagonal $\Sigma_{\boldsymbol{x}} = \mathrm{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2)$ with $\sigma_x = \sigma_y = \sigma_z = \sigma$. For the pure diffusion sampling is chosen $\sigma = \sigma_p = 0.1$ mm, for the Gaussian sampler $\sigma = \sigma_g = 2$ mm, and for the bridge test $\sigma = \sigma_b = 6$ mm. The Gaussian sampler should place the samples close to the border, therefore, we set $\sigma_g \leq d_t$. The bridge test apparently requires a high mobility of the samples in order to build a bridge in the narrow passage, thus $\sigma_b \geq 2d_t$ provides reasonable results.

On the right side of Fig. 5, the evolution is displayed as the history of $N = 1000$ samples within 10 repetitions of the observer cycle. The pure diffusion case shows an almost uniform distribution in the free space The Gaussian sampler captures samples near $\partial\tilde{\mathcal{C}}$, whereas the bridge test policy increases significantly the density at the entrance of the narrow passage and also near the border of the configuration space. Although obtained in simulation and with local resampling, these results show clearly the increased sample density in the critical regions as expected from the known analysis of [12] and [13].

[1] Video: http://youtu.be/2jYhdmk-pMg

[2] item Industrietechnik GmbH, http://www.item24.de

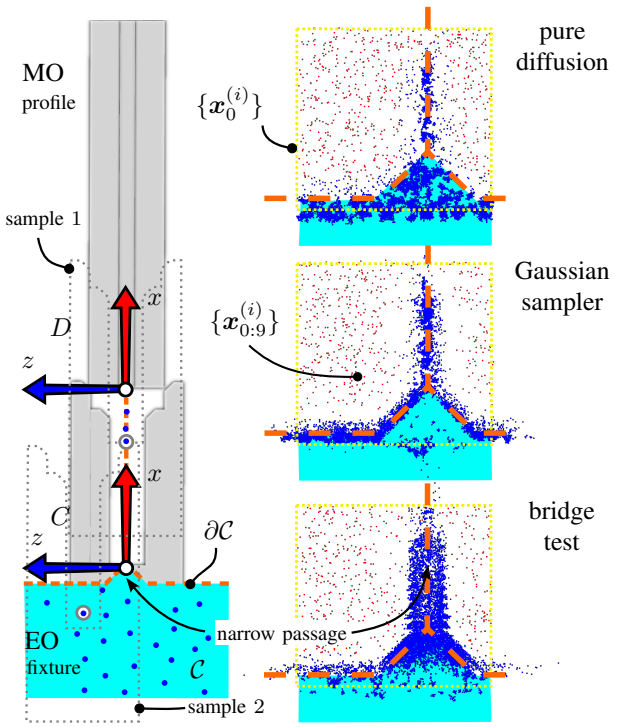


Fig. 5. The sampling policies at the entrance of the narrow passage. Here in simulation, the pose of the profile is kept constant, while the samples evolve. Note that each drawn dot represents a sample of the relative configuration space $\tilde{\mathcal{C}}$ at time k, i.e. a hypothetical pose of the fixture. Blue dots denote samples in $\tilde{\mathcal{C}}$, red marked samples violate $\tilde{\mathcal{C}}$. The yellow dotted line encloses the initial sample set $\{\boldsymbol{x}_0{}^{(i)}\}$. The orange dashed line visualizes the ideal border $\partial\mathcal{C}$, which is approximated by the sample distribution. Sample 1 is in the narrow passage, sample 2 in free space.

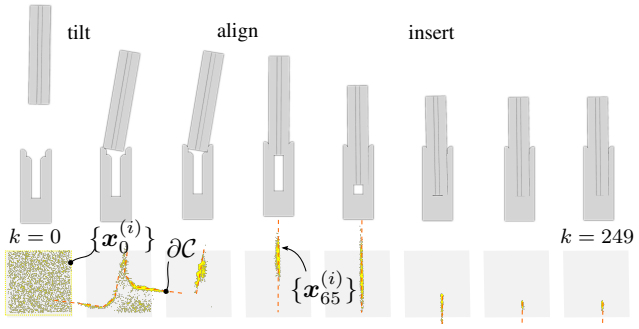
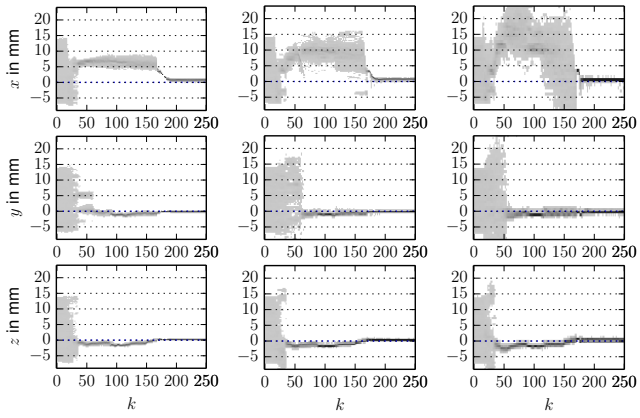


Fig. 6. The evolution of the samples based on measurements from the real assembly process for the bridge test propagation ($N = 1000$, front-view), as also presented in the video attachment. The dotted line encloses the initial sample set $\{\boldsymbol{x}_0{}^{(i)}\}$, whereas the single yellow dots represent the sample set $\{\boldsymbol{x}_k{}^{(i)}\}$ at time k. The samples condense at $\partial\tilde{\mathcal{C}}$, the border region of the local configuration space (orange dashed line).



(a) pure diffusion (b) Gaussian sampler (c) bridge test

Fig. 7. Exemplary sample evolution for the three propagation sampling policies ($N = 1000$). The sampled distribution of the state $\boldsymbol{x} = {}_{\bar{C}}(x, y, z)_{\bar{C}C}$ over the time step k is shown. All distributions condense when the profile is close to the narrow passage of the fixture at $k \approx 25$. Dark gray values indicate a high sample density. The x-values of the samples are distributed wider for the bridge test policy after $k \approx 100$, which features an improved filling of the narrow passage during the insertion of the profile.

C. Peg-in-Hole Experiment

In the observation of the real assembly process, the sample evolution is additional affected by the measured joint torques dependent on the phase of the assembly process. The process is visualized in Fig. 6 together with the distribution of the state samples with the bridge test as propagation function. The task execution and exemplary results are also presented in the video attachment. Like above, the samples condense at the entrance of the narrow passage in the configuration space. An exemplary evolution of all state variables is plotted in Fig. 7 for the three policies, with $N = 1000$. All distributions shrink when the profile is close to the narrow passage of the fixture at $k \approx 25$. At $k \approx 60$ the profile enters the more constrained region, where the slots at the sides are aligned inside with the fixture; at $k \approx 170$ the peg reaches the bottom. The sample populations of the Gaussian sampler and the bridge test show a faster filling of the narrow passage after an intermediate contraction during the alignment. Whereas the sample population for the pure diffusion is spreading only slowly and concentrated in clusters, thus indicating a stronger influence of sample impoverishment and a worse coverage of the relevant configuration space.

For the further evaluation of convergence and robustness, the insertion is recorded 5 times and evaluated multiple-times for each policy and each number of samples on a logarithmic scale $N \in \{10, 32, 100, 316, 1000, 3162\}$. The propagation models are equally parametrized like in Section IV-B. In the observation model, it is assumed that the measurement errors of the torques are independent for each joint. Therefore, a diagonal covariance matrix $\Sigma_{\boldsymbol{\tau}} = \mathrm{diag}(\sigma_\tau^2, \ldots, \sigma_\tau^2)$ is chosen in the observation density of the joint torques. With $\sigma_\tau = 5\,\mathrm{Nm}$, a large tolerance with respect to unmodeled effects and errors in the torque measurements is given. The residuum of the contact distance is $\sigma_d = 0.1\,\mathrm{mm}$.

For the analysis of the error in the observation, a goal pose $\boldsymbol{H}_{DC,goal}$ without clearance is assumed and therefore a uni-modal target distribution of the samples is present. The ground truth of the transformation at the end of the assembly process is given with

$$\boldsymbol{H}_{\bar{C}C,ground} := \boldsymbol{H}_{\bar{C}B}\boldsymbol{H}_{BA}(\boldsymbol{q}_{end})\boldsymbol{H}_{AD}\boldsymbol{H}_{DC,goal}, \quad (11)$$

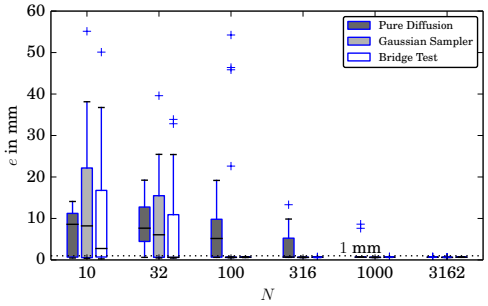


Fig. 8. Boxplot of errors with respect to the ground truth at the final observation step. A set of 5 records is evaluated 5 times for each policy and each value of sample number N. The error decrease with increased N. The bridge test shows consistent valid results with $e < 1\,\text{mm}$ for $N \geq 100$, the Gaussian sampler for $N \geq 316$, the pure diffusion sampler for $N \geq 3162$,

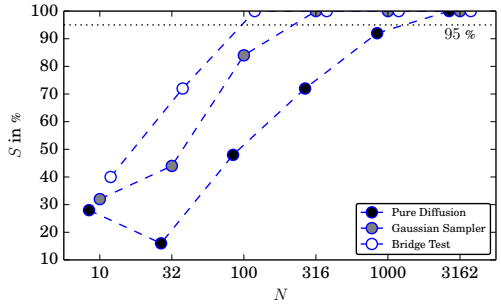


Fig. 9. Reliability S measured by the percentage of observations with $e < e_t = 1\,\text{mm}$ over N. For each sample number N and each policy, a set of 5 records is evaluated 5 times; the reliability is calculated with respect to the total number of evaluations, e.g. the bridge test shows reliable results with $S > 95\,\%$ for $N \geq 100$.

where the peg has successfully reached the goal pose and the robot configuration is $\boldsymbol{q}_{end}$, obtained from the joint measurements. With the help of Eq. (10), the expected value of the transformation from the assumed pose of the fixture to the observed value $\langle \boldsymbol{H}_{\bar{C}C}(\boldsymbol{x}_k) \rangle$ can be obtained. The error at the final observation step is then calculated by

$$e = ||\langle \boldsymbol{r}_{\bar{C}C}(\boldsymbol{x}_k) \rangle - \boldsymbol{r}_{\bar{C}C,ground}||_2, \tag{12}$$

with $\boldsymbol{r}_{\bar{C}C} \in \mathbb{R}^3$, the position component of $\boldsymbol{H}_{\bar{C}C}$. Fig. 8 shows the error for the policies for the repeated evaluations over the same sample number N and Fig. 9 the reliability measured by the percentage S of observations with $e < e_t = 1\,\text{mm}$. None of the policies show a reliable convergence for low numbers of particles $N < 100$ as $S < 95\,\%$. For $N \leq 100$, the Gaussian sampler and the bridge test appear to have the highest spread in the error distribution, which is due to the higher mobility of the samples ($\sigma_g, \sigma_b > \sigma_p$) and the potential to favor alternative narrow passages if the initial density is not sufficient. The error converges finally at $e \approx 0.5\,\text{mm}$ for all policies and $N > 1000$. The minimal required number of samples for a significant reliability of the convergence is different. The observations made with the pure diffusion model requires high sample numbers above $N > 1000$ for $S > 95\,\%$. A considerably lower number is achieved with the bridge test policy, where the expected value of the error is already close to the achievable accuracy at $N = 100$ and $S > 95\,\%$. The results of the Gaussian sampler lie in between.

V. Discussion

In the previous section, it was shown that the number of required samples needed for convergence can be reduced significantly by the application of more advanced sampling strategies in the propagation step of the SMC filter. The number of required samples could be reduced by a factor of $f \approx 1/10$ if the bridge test is applied instead of the pure diffusion model. The benefit of using the Gaussian sampler is comparable, but slightly smaller. The question is whether the computational effort introduced by the additional collision checks in the propagation model is feasible. In an optimal implementation of the observation algorithm, most time is spent in the collision checks. The update model needs a single call of VPS in the weighting of a sample according to the distance and torque consistency; in the worst case, further $L_{max} + 1$ calls are required in the propagation step of the Gaussian sampler and $2 \cdot L_{max}$ calls in the bridge test. Under these assumptions and given a sequential processing of the samples, the total time of a single observation step with N samples is:

$$T_{p,max} \approx N \cdot T_{vps}, \tag{13}$$

$$T_{g,max} \approx (L_{max} + 2) \cdot f \cdot N \cdot T_{vps}, \tag{14}$$

$$T_{b,max} \approx (2 \cdot L_{max} + 1) \cdot f \cdot N \cdot T_{vps}, \tag{15}$$

for the pure diffusion model, the Gaussian sampler, and the bridge test. The average time needed for a single contact computation is denoted as T_{vps}. In the given example with $N = 1000$ samples and $L_{max} = 5$ and $T_{vps} \approx 1\,\text{ms}$, we obtain $T_{p,max} = 1\,\text{s}$, $T_{g,max} = 0.7\,\text{s}$ and $T_{b,max} = 1.1\,\text{s}$. In the worst case, the advantages of the alternative policies in the needed computational time are not directly visible, but assuming that in average less than L_{max} iterations are required, an improvement can be achieved. Table I shows the actual rate of processed particles per second (pps) and the effective rate which incorporates the reduction of required particles averaged over all exemplary evaluations. An effective speedup of ≈ 2.5 is achieved by the bridge test, and ≈ 6 by the Gaussian sampler, when the reduced number of required samples is incorporated. A comparison to the performance of [8] is not directly possible as they use simpler geometries for probing, and the application for a peg-in-hole with more complex geometries is only given as an outlook. Nevertheless, the combination of the approach with a RBPF is reasonable when also orientation uncertainties are incorporated.

TABLE I

Rate of processed particles per second.

propagation model	actual rate	reduction factor f	effective rate
pure diffusion	180 pps	1.0	180 pps
Gaussian sampler	110 pps	0.1	1100 pps
bridge test	45 pps	0.1	450 pps

In any case, a general speedup of the computation is necessary in order to accomplish an online observation of the assembly process. The current implementation does not run in real-time, but also does not yet use the potential of an optimized implementation and a parallelized architecture for which a speedup greater than 10 is technically feasible; compare [21] for an analysis of the theoretical speedup in the distributed implementation of SMC algorithms. Therefore, a minimal observation rate of 10 Hz seems likely in the current scenario. This rate might be too low for the direct usage as input in the lower torque level controllers of the robot, but is sufficient for high level task control and reasoning in future applications. Especially the larger volume of samples within the narrow passage increases the knowledge about the local configuration space which might be exploited for parametrization of low level controllers, e.g. the stiffness of the impedance controller could be adapted according to the current phase of the insertion of the peg and the shape of the local configuration space. And of course, also the quality of repetitive tasks can be improved when initially uncertain goal frames are updated by the assembly observation system.

VI. Conclusions

An observation method based on SMC was presented, which improves the capability of a robot system to use joint torque sensing as an input for the observation of robotic assembly tasks. It makes use of the fast haptic rendering algorithm VPS for complex geometrical objects and therefore is suitable for a large variety of assembly parts. The observation method was evaluated with joint torque measurements obtained from a peg-in-hole assembly with a LWR robot arm. Especially it was shown how samples of the uncertain space can be dragged into narrow passages of the configuration space in order to increase the sample density in these critical areas. Thus, the number of total samples needed for convergence can be reduced significantly and a local sampled map of the configuration space can be obtained. This information can be used in future high-level controllers in order to adapt strategies to the current execution. The potential of using these sampling methods in the propagation step of the observation was demonstrated, but further investigations have to be carried out in order to weight the benefits against the time constraints that are given in the real process. Further analysis is required in the presence of large relative orientational errors, as only results on position uncertainties were presented here. In addition, alternative adaptive methods should be investigated, which could improve the performance of the observation more efficiently.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287787, project SMErobotics.

References

[1] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential monte carlo," *Proc. of the IEEE*, vol. 95, no. 5, pp. 899 – 924, 2007.

[2] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: fifteen years later," *Handbook of Nonlinear Filtering*, vol. 12, pp. 656–704, 2009.

[3] S. Chhatpar and M. Branicky, "Localization for robotic assemblies using probing and particle filtering," in *Adv. Intelligent Mechatronics. Proc., 2005 IEEE/ASME Int. Conf. on*, July 2005, pp. 1379–1384.

[4] U. Thomas, S. Molkenstruck, R. Iser, and F. Wahl, "Multi sensor fusion in robot assembly using particle filters," in *Robotics and Automation, 2007 IEEE Int. Conf. on*, April 2007, pp. 3837–3843.

[5] P. Tang and J. Xiao, "Automatic generation of high-level contact state space between 3d curved objects," *The Int. Journal of Robotics Research*, vol. 27, no. 7, 2008.

[6] W. Meeussen, J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. De Schutter, "Contact-state segmentation using particle filters for programming by human demonstration in compliant-motion tasks," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 218–231, 2007.

[7] K. Hertkorn, M. Roa, C. Preusche, C. Borst, and G. Hirzinger, "Identification of contact formations: Resolving ambiguous force torque information," in *Robotics and Automation (ICRA), 2012 IEEE Int. Conf. on*, May 2012, pp. 3278–3284.

[8] Y. Taguchi, T. Marks, and H. Okuda, "Rao-blackwellized particle filtering for probing-based 6-dof localization in robotic assembly," in *Robotics and Automation (ICRA), 2010 IEEE Int. Conf. on*, May 2010, pp. 2610–2617.

[9] M. Chalon, J. Reinecke, and M. Pfanne, "Online in-hand object localization," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ Int. Conf. on*, Nov 2013, pp. 2977–2984.

[10] L. E. Zhang and J. C. Trinkle, "The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing," in *Proc. of the 2012 IEEE Int. Conf. on Robotics and Automation*. IEEE, 2012, pp. 3805–3812.

[11] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[12] V. Boor, M. Overmars, and A. van der Stappen, "The gaussian sampling strategy for probabilistic roadmap planners," in *Robotics and Automation, 1999. Proc.. 1999 IEEE Int. Conf. on*, vol. 2, 1999, pp. 1018–1023 vol.2.

[13] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. Reif, "Narrow passage sampling for probabilistic roadmap planning," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1105–1115, Dec 2005.

[14] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *ACM SIGGRAPH 2005 Courses*, ser. SIGGRAPH '05. New York, USA: ACM, 2005.

[15] M. Sagardia, T. Stouraitis, and J. a. Lopes e Silva, "A new fast and robust collision detection and force computation algorithm applied to the physics engine bullet," in *Conf. and Exhibition of the European Association of Virtual and Augmented Reality (EuroVR)*, 2014.

[16] A. Stemmer, A. Albu-Schäffer, and G. Hirzinger, "An analytical method for the planning of robust assembly tasks of complex shaped planar parts," in *Proc. of the 2007 IEEE Int. Conf. on Robotics and Automation*. IEEE, 2007, pp. 317 – 323.

[17] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decr, R. Smits, Aertbelin, K. Claes, and H. Bruyninckx, "Constrained-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The Int. Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[18] J. Barbič, "Real-time reduced large-deformation models and distributed contact for computer graphics and haptics," Ph.D. dissertation, Carnegie Mellon University, 2007.

[19] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, "The DLR lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: An Int. Journal*, vol. 34, no. 5, pp. 376–385, 2007.

[20] D. Leidner, C. Borst, and G. Hirzinger, "Things are made for what they are: Solving manipulation tasks by using functional object classes," in *Humanoid Robots,12th IEEE-RAS Int. Conf. on*, 2012, pp. 429–435.

[21] A. Bashi, V. Jilkov, R. Li, and H. Chen, "Distributed implementations of particle filters," in *Proceedings of the 6th International Conference of Information Fusion*, 2003, pp. 1164–1171.

Publication 3

K. Nottensteiner and K. Hertkorn (2017): "*Constraint-based Sample Propagation for Improved State Estimation in Robotic Assembly*". IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017.

Version Note

The following attached version corresponds to the accepted manuscript of the publication.

The final published version is available under:

- https://ieeexplore.ieee.org/document/7989069

Please refer to the final published version for citation:

```
@InProceedings{Nottensteiner2017,
  author =      {Nottensteiner, Korbinian and Hertkorn, Katharina},
  title =       {Constraint-based Sample Propagation for Improved
                 State Estimation in Robotic Assembly},
  booktitle =   {2017 IEEE International Conference on Robotics and
                 Automation (ICRA)},
  year =        2017,
  venue =       {Singapore},
  eventdate =   {2017-05-29/2017-06-03},
  publisher =   {IEEE},
  pages =       {549-556},
  isbn =        {978-1509046331},
  eid =         {2-s2.0-85027994902},
  doi =         {10.1109/ICRA.2017.7989069}}
```

Copyright Note

Eratum

- In equation (31) read $\boldsymbol{x}_k{}^{(i)} \sim \mathcal{N}(\bar{\boldsymbol{x}}_k{}^{(i)}, \Sigma_{\boldsymbol{x}})$ instead of $\boldsymbol{x}_k{}^{(i)} = \mathcal{N}(\bar{\boldsymbol{x}}_k{}^{(i)}, \Sigma_{\boldsymbol{x}})$.

Constraint-based Sample Propagation for Improved State Estimation in Robotic Assembly

Korbinian Nottensteiner[1], Katharina Hertkorn[2]

***Abstract*— In fast changing assembly scenarios, it is required to adapt the task execution to the current state of the setup without extensive calibration routines. Therefore, it is important to estimate the geometric uncertainties and contact states during the assembly execution. We use a sequential Monte Carlo (SMC) method to track the relative poses between workpieces during a robotic assembly based on joint torque and position measurements only. In contrast to existing approaches, we focus on assembly tasks where the workpiece is not fixed in the workcell, but can, for example, slide on a table surface. We propose a new constraint-based propagation model for the SMC approach: a compensation motion for the samples dependent on the violation of contact constraints is derived. This allows us to track the motion of the workpieces in cases where a common random diffusion model fails. The method is evaluated with experiments using an assembly scenario with two KUKA LBR iiwa robot arms and shows accurate tracking performance.**

I. Introduction

During assembly processes, workpieces pass through various contact states, i.e. from no contact to fully constrained goal configurations. It is important to observe such contact states especially in fast changing assembly setups where uncertainties in the workcell make it difficult to rely on absolute accuracy. The observation methods presented in our work can be used to estimate geometric uncertainties in a workcell setup where the poses of the workpieces are not statically fixed or only roughly known. The state estimates can be integrated in high-level control, quality management, and automatic strategy adaption.

The estimation of geometric uncertainties and contact states has a long tradition in compliant motion control [1]. In particular, sequential Monte Carlo (SMC) approaches were applied in recent years as they can handle non-linear models, non-Gaussian and multimodal distributions [2]. Within the domain of assembly observation, the SMC framework was first applied in the probing-based localization of a lock-key assembly [3], in the simultaneous contact state segmentation and estimation of geometric uncertainties [4], and in the fusion with vision and force torque measurements in various peg-in-hole tasks [5]. Further improvements in performance were then achieved by [6] through the combination of sample-based particle filtering and an extended Kalman filter. Recently, we showed how narrow passage sampling methods can be applied for better performance in the observation of peg-in-hole tasks [7]. All of the works have in common that they assume a statically fixed environment in which no motion of the objects occurs.

[1]korbinian.nottensteiner@dlr.de
Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany
[2]katharina@hertkorn.com

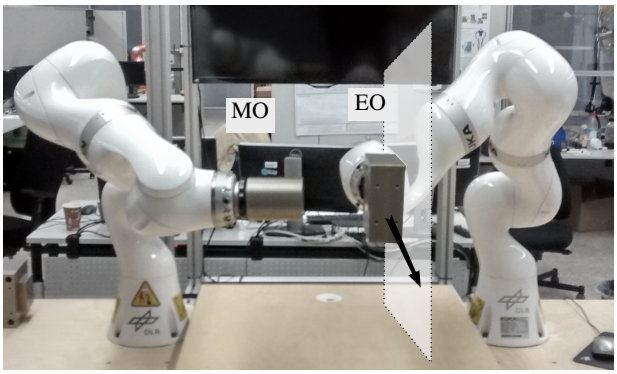


Fig. 1. The assembly scenario uses two KUKA LBR iiwa robot arms mounted on a workbench. The left robot assembles the peg (manipulated object, MO) into the hole (environment object, EO). The right arm simulates the sliding of the EO on a table surface (transparent grey plane).

The general SMC algorithm has two major steps in a sequential estimation loop: first, samples of an unknown state are propagated using a propagation model and then updated according to their consistency with the available sensory information and the underlying observation models [2]. Most of the aforementioned methods focus on the second step and assume a rather simple random motion model in the first. This approach falls short when the workpiece in the environment is not fixed, e.g. in an assembly without special fixtures (see Fig. 1). One of the reasons for the failure is that the current robot motion is not incorporated in the propagation step. It is only used in the update to check consistency of the samples with the observed velocity. During a standard propagation step, samples are equally spread in all spatial directions. This is especially critical when the estimates have already converged in a narrow region of the configuration space, e.g. during a peg-in-hole task. The samples might then be too slow to follow a sudden motion when the piece in the environment starts to slip away.

We therefore present a new motion model for the samples in order to improve the tracking capabilities during the assembly. The proposed method is derived from the fundamental property of a rigid body in static equilibrium saying that no energy is produced, i.e. the body is passive. Following the concept of time domain passivity control [8], we compute a compensation motion if energy is generated. This allows us to correct the virtual model of the contact before the update step, i.e. make a better proposal of the

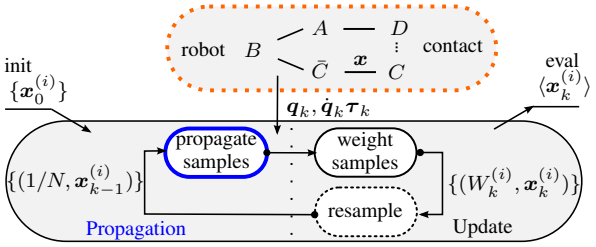


Fig. 2. The observation algorithm: samples $\boldsymbol{x}^{(i)}$ of the hidden state $\boldsymbol{x}$ are sequentially propagated, weighted according to their consistency with the observation $(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \boldsymbol{\tau}_k)$, i.e. the robotic data , and re-sampled.

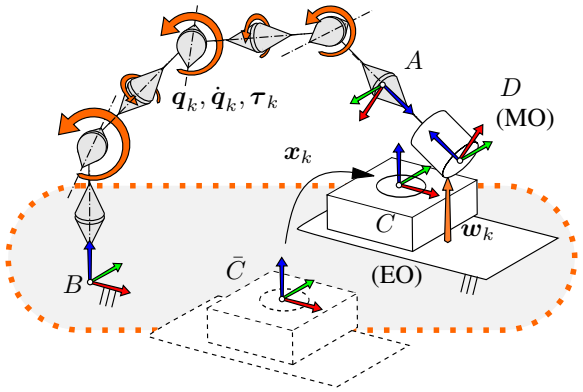


Fig. 3. The manipulator model describes the kinematic relation between the manipulated object (D, MO) and the environment object (C, EO). It uses the measured joint angles $\mathbf{q}_k$ and torques $\boldsymbol{\tau}_k$. $\bar{C}$ is the assumed initial pose of EO.

future state. We derive a general method to compute the compensation motion for the samples using the measured robot motion.

This work is structured as follows. First, the basic observation algorithm and the mechanical models are introduced in Section II and then in Section III, the new concept for sample propagation is derived. We show experiments with the dual-arm setup and discuss the results in Section IV. Finally, the paper is summarized in Section V.

II. Models for Contact Observation

In the following section, we present the basic models used in our observation framework as depicted in Fig. 2. First, the mechanical model is presented including a kinematic model and a short description of the applied contact model. Then, a brief introduction into the SMC algorithm is given. This section is the basis for the further derivation of our improved propagation model in Section III. Compare our previous work [7] for more details on the used models.

A. Mechanical Model

The considered manipulator is a single lightweight robot arm with n joints. A rigid body model is applied that closes the kinematic chain between the robot arm, a manipulated object (MO) attached to the robot flange and an object in the environment (EO). The kinematics are illustrated in Fig. 3 for a peg-in-hole where the EO can slide along a plane. The sliding motion of the EO is in general not directly observable and therefore, we model the pose of the EO as a hidden uncertainty state $\boldsymbol{x}$, see [10] for a general notion on the specification of geometric uncertainties in kinematic chains. Here, the state $\boldsymbol{x} = (r_x, r_y, r_z, \alpha, \beta, \gamma) \in \mathbb{R}^6$ will represent the translation and rotation uncertainties of the EO frame C with respect to an initial reference frame $\bar{C}$ in a transformation $\boldsymbol{H}_{\bar{C}C}(\boldsymbol{x})$, where $\bar{C}$ is given with respect to the robot base B as constant transformation $\boldsymbol{H}_{B\bar{C}} \in SE(3)$. We assume that the MO is statically fixed with respect to the robot flange A. Thus, the current pose of the reference frame D of the MO relative to the robot base B is given by the forward kinematics $\boldsymbol{H}_{BD}(\boldsymbol{q})$ with joint configuration $\boldsymbol{q} \in \mathbb{R}^n$. At time $t = t_k$, the kinematic closure equation defines the relative transformation $\boldsymbol{H}_k$ between C (EO) and D (MO) as:

$$\boldsymbol{H}_k = \boldsymbol{H}_{CD}(\boldsymbol{q}_k, \boldsymbol{x}_k) = \boldsymbol{H}_{C\bar{C}}(\boldsymbol{x}_k)\boldsymbol{H}_{\bar{C}B}\boldsymbol{H}_{BD}(\boldsymbol{q}_k), \quad (1)$$

which will be used in the observation model to compute the contact forces and $\boldsymbol{q}_k = \boldsymbol{q}(t_k) \in \mathbb{R}^n$, $\boldsymbol{x}_k = \boldsymbol{x}(t_k) \in \mathbb{R}^6$.

The joint velocity $\dot{\boldsymbol{q}}_k = \dot{\boldsymbol{q}}(t_k) \in \mathbb{R}^n$ is approximated based on the measured joint configuration:

$$\dot{\boldsymbol{q}}_k \approx \frac{\boldsymbol{q}(t_k) - \boldsymbol{q}(t_k - \Delta T_{rob})}{\Delta T_{rob}}, \quad (2)$$

where ΔT_{rob} is the sample time of the robot measurements. Using the Jacobian matrix of the manipulator $\boldsymbol{J}_k = \boldsymbol{J}(\boldsymbol{q}_k) \in \mathbb{R}^{6\times n}$, the twist ${}_A\boldsymbol{t}$ with respect to the robot flange A is then given as [11]:

$${}_A\boldsymbol{t}_k = [{}_A\boldsymbol{v}^T, {}_A\boldsymbol{\omega}^T]^T = \boldsymbol{J}_k\dot{\boldsymbol{q}}_k. \quad (3)$$

The twist is composed of linear velocity ${}_A\boldsymbol{v} \in \mathbb{R}^3$ and angular velocity ${}_A\boldsymbol{\omega} \in \mathbb{R}^3$. Furthermore, we represent the relative twist space between MO and EO as a matrix $\boldsymbol{T} = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_m] \in \mathbb{R}^{(6\times m)}$ with the base twists $\boldsymbol{t}_j$ and dimension $m \leq 6$.

The robot controller provides an estimate of the external joint torques $\boldsymbol{\tau}_k = \boldsymbol{\tau}(t_k) \in \mathbb{R}^n$ based on measurements in the robot joints. The joint torques are induced by a contact wrench ${}_A\boldsymbol{w}^T = [{}_A\boldsymbol{F}^T, {}_A\boldsymbol{M}^T]$ at the MO, consisting of a force ${}_A\boldsymbol{F} \in \mathbb{R}^3$ and a torque ${}_A\boldsymbol{M} \in \mathbb{R}^3$, and the following well-known relation holds:

$$\boldsymbol{\tau}_k = \boldsymbol{J}_k^T {}_A\boldsymbol{w}_k. \quad (4)$$

We assume that the inertia properties are known and the geometries of the MO and EO are given as a 3D boundary representation model, e.g. from a CAD system. The DLR implementation [12] of the voxmap-pointshell algorithm (VPS, [13]) is then used as a reference contact model to check consistency of a sample of the unknown state $\boldsymbol{x}_k$ with the measured contact forces as will be described in the next section. The algorithm provides the contact distance $\tilde{d}_k = \tilde{d}(\boldsymbol{H}_k) \in \mathbb{R}$ and computes a contact wrench ${}_D\tilde{\boldsymbol{w}}_k^T = {}_D\tilde{\boldsymbol{w}}^T(\boldsymbol{H}_k)$ for a pose $\boldsymbol{H}_k$. The contact wrench from our reference model ${}_D\tilde{\boldsymbol{w}}_k$ is mapped according to (4) into the joint space [11]:

$$\tilde{\boldsymbol{\tau}}_k = \boldsymbol{J}_k^T {}_A\tilde{\boldsymbol{w}}_k = \boldsymbol{J}_k^T \boldsymbol{\Delta}\, \mathrm{Ad}(\boldsymbol{H}_{AD})\boldsymbol{\Delta}_D\tilde{\boldsymbol{w}}_k \quad (5)$$

with the adjoint operation

$$\mathrm{Ad}(\boldsymbol{H}_{AD}) = \begin{bmatrix} \boldsymbol{R}_{AD} & [{}_A\boldsymbol{r}_{AD}]_\times \, \boldsymbol{R}_{AD} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{R}_{AD} \end{bmatrix}, \tag{6}$$

the rotation matrix $\boldsymbol{R}_{AD} \in SO(3)$, the translation vector ${}_A\boldsymbol{r}_{AD} \in \mathbb{R}^3$ of $\boldsymbol{H}_{AD}$ written as skew-symmetric matrix $[\boldsymbol{r}]_\times$ and Δ is the so-called interchange operator [14] to change the coordinate order of the wrench:

$$[\boldsymbol{r}]_\times = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_x & r_x & 0 \end{bmatrix}, \boldsymbol{\Delta} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}. \tag{7}$$

B. Observation Model

The objective for our observation algorithm is to estimate the unknown state $\boldsymbol{x}_k$ given the set of observations $\boldsymbol{y}_{0:k} = \{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_k\}$ made from $t = 0$ to $t = t_k$:

$$p\left(\boldsymbol{x}_k | \boldsymbol{y}_{0:k}\right). \tag{8}$$

As in our previous work [7], a SMC approach will be applied to solve this Bayesian estimation problem. Within the SMC framework, hypothesis $\boldsymbol{x}_k^{(i)}$ with $i = 1, ..., N$ for the state value are at first sampled from an initial uncertainty distribution, then sequentially propagated and weighted according to their consistency with the made observations [2]. In our case, the observations at time k are $\boldsymbol{y}_k = (\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \boldsymbol{\tau}_k)$ as given in the previous section.

The observation cycle of the algorithm is depicted in Fig. 2. In the propagation step, the samples are moved according to an importance distribution $q(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{y}_k)$. In the existing approaches for contact state observation with a static EO, e.g. [3], this is often a pure random motion model of the samples in the uncertain space. Also a function to increase sample density in relevant contact regions, as shown in our previous work [7], has advantages in assembly tasks with narrow passages in the configuration space. In the next section, a new constraint-based propagation model will be presented that improves the tracking properties of the samples if the EO is not statically fixed.

After the propagation, the samples are weighted and afterwards resampled according the sequential importance sampling and resampling (SIS/R) scheme [2]. As in [7], we apply here a consistency based observation model for the joint torques:

$$p_{\boldsymbol{\tau}}(\boldsymbol{\tau}_k | \boldsymbol{x}_k^{(i)}) = \mathcal{N}(\boldsymbol{\tau}_k | \tilde{\boldsymbol{\tau}}_k^{(i)}, \Sigma_{\boldsymbol{\tau}}), \tag{9}$$

where $p_{\boldsymbol{\tau}}(\boldsymbol{\tau}_k | \boldsymbol{x}_k^{(i)})$ is the observation density for the measured joint torque evaluated for $\boldsymbol{x}_k^{(i)}$. We assume that $p_{\boldsymbol{\tau}}(\boldsymbol{\tau}_k | \boldsymbol{x}_k)$ is normally distributed around the reference contact torque $\tilde{\boldsymbol{\tau}}_k^{(i)}$ from (5) with the covariance matrix $\Sigma_{\boldsymbol{\tau}}$ referring to the error between virtual contact model and the real robot measurement. The contact distance $\tilde{d}_k^{(i)}$ for a sample i is used to penalize inadmissible violations of the configuration space between EO and MO [7]:

$$p_{\boldsymbol{q}}(\boldsymbol{q}_k | \boldsymbol{x}_k^{(i)}) = (\sigma_d \sqrt{2\pi})^{-1} \cdot \begin{cases} 1, & \tilde{d_k}^{(i)} \leq d_t \\ \exp(-\frac{(\tilde{d}_k - d_t)^2}{(2\sigma_d^2)}), & \tilde{d_k}^{(i)} > d_t. \end{cases} \tag{10}$$

Note that a small penetration is required for the VPS algorithm in order to compute the contact wrench. The standard deviation σ_d and $d_t > 0$ control the allowed penetration.

The total observation density is then used for assigning weights $W_k^{(i)}$ to the samples and is given under the assumption of independence as [7]:

$$p(\boldsymbol{y}_k | \boldsymbol{x}_k^{(i)}) = p_{\boldsymbol{q}}(\boldsymbol{q}_k | \boldsymbol{x}_k^{(i)}) \;\cdot\; p_{\boldsymbol{\tau}}(\boldsymbol{\tau}_k | \boldsymbol{x}_k^{(i)}). \tag{11}$$

The cycle continues after resampling the samples according to their weights. At every time instance, the expected value of $\langle \boldsymbol{x}_k^{(i)} \rangle$ or other properties of the sample distribution can be computed to monitor the task execution.

III. Velocity- and Constraint-based Propagation

In this section, a velocity- and our constraint-based propagation method will be described that improve the tracking performance by using the current measured velocity of the manipulator to place samples better as in a pure random diffusion model. The method is derived from the fundamental reciprocity property: when a body is in equilibrium, then an external wrench and a twist do not produce energy for a small displacement [15], the instantaneous work P vanishes:

$$P = 0. \tag{12}$$

Reciprocity was already used in the past to derive identification equations for uncertainties in compliant motions [16], but it was to the best of the authors knowledge not yet applied in the propagation model of SMC based algorithms in the context of assembly observation although it found heavy usage in the update step, e.g. [4].

The virtual instantaneous work $\tilde{P}^{(i)}$ produced by the current manipulator twist $\boldsymbol{t}_k$ at time k with the virtual wrench $\tilde{\boldsymbol{w}}_{k-1}^{(i)}$ for a sample $\boldsymbol{x}_{k-1}^{(i)}$ from the previous observation cycle is given as the product:

$$\tilde{P}^{(i)} = \boldsymbol{t}_k^T \tilde{\boldsymbol{w}}_{k-1}^{(i)}. \tag{13}$$

If $\boldsymbol{t}_k$ and $\tilde{\boldsymbol{w}}_{k-1}^{(i)}$ are a contrary screw pair [17], then they produce negative virtual energy, i.e. virtual contact constraints are violated by the current robot motion. In case the sample would not be propagated ($\boldsymbol{x}_k = \boldsymbol{x}_{k-1}$), virtual energy $\Delta\tilde{E}^{(i)}$ would be generated within one observation cycle of duration ΔT:

$$\Delta\tilde{E}^{(i)} \approx \tilde{P}^{(i)} \cdot \Delta T. \tag{14}$$

We apply now an approach inspired by time-domain passivity control [8], [18] to derive a compensation velocity of the sample such that the superfluous virtual energy $\Delta\tilde{E}^{(i)}$ vanishes. According to (12), it follows:

$$\Delta\tilde{E}^{(i)} + \boldsymbol{m}^{(i),T} \tilde{\boldsymbol{w}}_{k-1}^{(i)} \cdot \Delta T = 0, \tag{15}$$

where $\boldsymbol{m}^{(i)}$ is the compensation velocity and is chosen according to

$$\boldsymbol{m}^{(i)} = \begin{cases} \boldsymbol{A}^{(i)}{}_D \boldsymbol{t}_k, & \Delta\tilde{E}^{(i)} < 0 \\ 0, & \Delta\tilde{E}^{(i)} \geq 0. \end{cases} \tag{16}$$

Note that (15) is invariant against a change in coordinate reference frame (compare [11]) and hence, (15) and (16) can be formulated in the reference frame D of the MO, where ${}_D\boldsymbol{t}_k = \mathrm{Ad}(\boldsymbol{H}_{DA}){}_A\boldsymbol{t}_k$. The matrix $\boldsymbol{A}^{(i)}$ is a velocity mapping matrix that is composed of a scaling factor $a^{(i)} \in \mathbb{R}$ and a twist filter matrix $\tilde{\boldsymbol{C}}^{(i)} \in \mathbb{R}^{6\times 6}$ such that:

$$\boldsymbol{A}^{(i)} = a^{(i)} \cdot \tilde{\boldsymbol{C}}^{(i)}. \tag{17}$$

Insertion of (16) and (17) into (15) leads to

$$\Delta\tilde{E}^{(i)} + a^{(i)} \cdot (\tilde{\boldsymbol{C}}^{(i)}{}_D\boldsymbol{t}_k)^T {}_D\tilde{\boldsymbol{w}}^{(i)}_{k-1} \cdot \Delta T = 0 \tag{18}$$

which can be solved for $a^{(i)}$, i.e. the motion rate necessary for the energy compensation:

$$a^{(i)} = -\frac{\Delta\tilde{E}^{(i)}_k}{{}_D\boldsymbol{t}_k^T\tilde{\boldsymbol{C}}^{(i),T}{}_D\tilde{\boldsymbol{w}}^{(i)}_{k-1} \cdot \Delta T} \tag{19}$$

and we obtain a general description for the compensation motion to guarantee passivity:

$$\boldsymbol{m}^{(i)} = -\frac{\Delta\tilde{E}^{(i)}_k}{{}_D\boldsymbol{t}_k^T\tilde{\boldsymbol{C}}^{(i),T}{}_D\tilde{\boldsymbol{w}}^{(i)}_{k-1} \cdot \Delta T}\ \tilde{\boldsymbol{C}}^{(i)}{}_D\boldsymbol{t}_k. \tag{20}$$

A trivial solution for (20) is found for $\tilde{\boldsymbol{C}}^{(i)} = \boldsymbol{I}$, which corresponds to a reverse motion of the robot arm moving out of the contact:

$$\boldsymbol{m}^{(i)} = -{}_D\boldsymbol{t}_k. \tag{21}$$

Note that as the measured motion of the robot arm can not be changed, we have to change the inertial reference frame to obtain a compensation motion ${}_D\boldsymbol{t}^{(i)}$ for the sample:

$${}_D\boldsymbol{t}^{(i)} = -\boldsymbol{m}^{(i)} = {}_D\boldsymbol{t}_k \tag{22}$$

This trivial solution will be called velocity-based propagation in the following and can obviously be interpreted as a rigid attachment of the sample to the flange of the moving manipulator.

An alternative constraint-based solution can be obtained by taking the twist space $\boldsymbol{T}^{(i)}$ of the contact into account:

$$\tilde{\boldsymbol{C}}^{(i)} = \boldsymbol{I} - \boldsymbol{T}^{(i)}\boldsymbol{T}^{(i),\#}. \tag{23}$$

Then, $\tilde{\boldsymbol{C}}^{(i)}$ filters the robot motion such that only motions remain which violate the contact constraints and point in the directions where superfluous virtual energy is generated. The pseudo inverse matrix $\boldsymbol{T}^{\#}$ of the twist space is calculated as [19]:

$$\boldsymbol{T}^{\#} = (\boldsymbol{T}^T\boldsymbol{M}_v\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{M}_v, \tag{24}$$

using the weighting matrix $\boldsymbol{M}_v$, which is chosen here as the mass matrix of the EO with respect to D to minimize kinetic energy of the compensation motion. We obtain now with (20) and (23) the constraint-based sample motion:

$${}_D\boldsymbol{t}^{(i)} = \frac{{}_D\boldsymbol{t}_k^T{}_D\tilde{\boldsymbol{w}}^{(i)}_{k-1}\ \left(\boldsymbol{I} - \boldsymbol{T}^{(i)}\boldsymbol{T}^{(i),\#}\right)}{{}_D\boldsymbol{t}_k^T\left(\boldsymbol{I} - \boldsymbol{T}^{(i)}\boldsymbol{T}^{(i),\#}\right)^T{}_D\tilde{\boldsymbol{w}}^{(i)}_{k-1}}{}_D\boldsymbol{t}_k. \tag{25}$$

Both the velocity-based and the constraint-based propagation will be evaluated and compared in the next section. The compensation motions were deduced with respect to frame D in which we assume that the twist space of the contact is known. In order to convert it to a sample motion the velocity field of the twist has to be written relative to the sample pose $\boldsymbol{x}^{(i)}_{k-1}$:

$${}_C\boldsymbol{t}^{(i)} = \mathrm{Ad}(\boldsymbol{H}_{CD}(\boldsymbol{x}^{(i)}_{k-1}, \boldsymbol{q}_k))\ {}_D\boldsymbol{t}^{(i)}. \tag{26}$$

The proposed new pose $\bar{\boldsymbol{x}_k}^{(i)}$ of the sample at k is then obtained with the exponential map [11]:

$$\boldsymbol{H}_{\bar{C}C}(\bar{\boldsymbol{x}}^{(i)}_k) = \boldsymbol{H}_{\bar{C}C}(\boldsymbol{x}^{(i)}_{k-1})\exp([{}_C\boldsymbol{t}^{(i)}] \cdot \Delta T) \tag{27}$$

with the twist ${}_C\boldsymbol{t}^{(i)}$ written as

$$[{}_C\boldsymbol{t}^{(i)}] = \begin{bmatrix}[\boldsymbol{\omega}]_\times & \boldsymbol{v}\\ \boldsymbol{0}_{1\times 3} & 0\end{bmatrix} \in se(3). \tag{28}$$

IV. Evaluation

A. Experimental Setup

The presented method is designed for a single-arm robotic assembly scenario, where the workpiece potentially moves, e.g. on a table surface. The experiments are carried out on a dual-arm setup, where one robot actively performs the assembly and the other robot arm passively simulates the movement of the workpiece in the environment. This allows us to evaluate the presented methods with different spatial constraints and to use the sensor data from the second arm as a ground truth reference for the estimated pose of the workpiece in the environment.

We use two KUKA LBR iiwa robot arms with n=7 joints mounted on a workbench as shown in Figs. 1 and 4. The joints are equipped with torque sensors which allow for sensing of external contact forces and impedance control-based execution of assembly tasks [9]. The impedance controller of the passive arm is parameterized such that it has full motion freedom only in direction of a virtual table plane (depicted in transparent grey in Fig. 4).

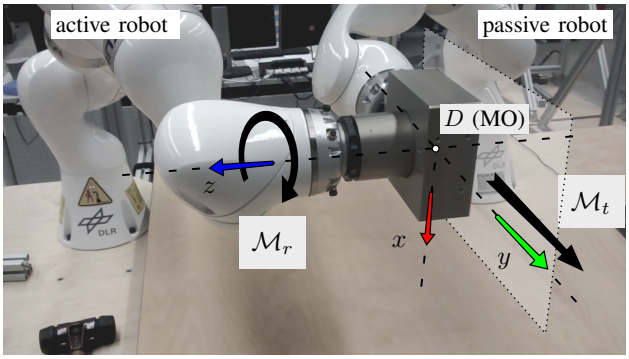


Fig. 4. Dual-arm setup. The active robot performs the assembly, the passive robot simulates the movement of the environment object. D (MO) is the coordinate system of the manipulated object. $\mathcal{M}_t$ and $\mathcal{M}_r$ are the carried out translational and rotational movements. The transparent plane depicts the virtual table plane.

In order to obtain clear results of the tracking performance, parts with known and primitive geometrical shapes are rigidly attached to the arms. In particular, there is a round- and a square-shaped peg and hole. The diameter of the round peg and the side length of the square peg are both 79 mm and the clearance with the hole in the assembled state is 1 mm. The peg is attached to the active robot, and the hole to the passive robot as depicted in Fig. 5.

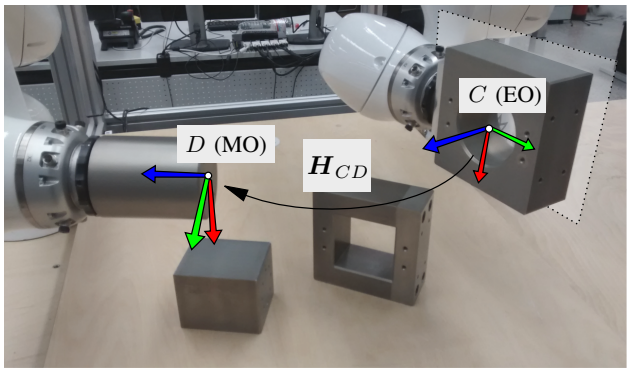


Fig. 5. We use two experimental setups: a round- and a square-shaped peg and hole. The coordinate systems of the objects are exemplary shown for the round-shaped peg (D) and hole (C).

The experimental assembly sequence can be divided in two parts: (i) the peg-in hole insertion, (ii) various movements of the active robot with the attached peg, the passive robot follows, i.e. simulates the slipping of the hole on the table. Compare our previous work [7] for a sample-based observation approach of (i). In this work, we primarily investigate (ii). In Part (ii), the observation based on the sensor data of the active robot is performed. The data is evaluated offline and in the current results, a high level control loop based on the observation is not yet closed, but will be presented in future work.

Part (ii) can consist of three movement types in our experiments: a pure translation $\mathcal{M}_t$ of 100 mm along the y-axis of C with a Cartesian velocity of 10 mm/s, a pure rotation $\mathcal{M}_r$ of 10 deg with respect to the hole axis (z-axis of C), or a combination of both motions $\mathcal{M}_c$, see Fig. 6. The induced motion of the passive hole, i.e. the sliding part in the environment, is depicted with dotted lines.

Clearly, in the assumed contact state, the square peg is unconstrained only into the translational direction of the hole axis, and the round peg is unconstrained in the translation and rotation with respect to the axis. The relative twistspace in contact with respect to the square-shaped peg is then represented as matrix ${}_D\boldsymbol{T} \in \mathbb{R}^{6\times 1}$ with the base twist $\boldsymbol{t}_{tz} \in \mathbb{R}^{6\times 1}$:

$$ {}_D\boldsymbol{T}^T = \begin{bmatrix}\boldsymbol{t}_{tz}^T\end{bmatrix} = \begin{bmatrix}0,0,1,0,0,0\end{bmatrix} \tag{29}$$

and for the round part ${}_D\boldsymbol{T} \in \mathbb{R}^{6\times 2}$ with the base twists $\boldsymbol{t}_{tz},\ \boldsymbol{t}_{rz} \in \mathbb{R}^{6\times 1}$:

$$ {}_D\boldsymbol{T}^T = \begin{bmatrix}\boldsymbol{t}_{tz}^T \\ \boldsymbol{t}_{rz}^T\end{bmatrix} = \begin{bmatrix}0,0,1,0,0,0 \\ 0,0,0,0,0,1\end{bmatrix}. \tag{30}$$

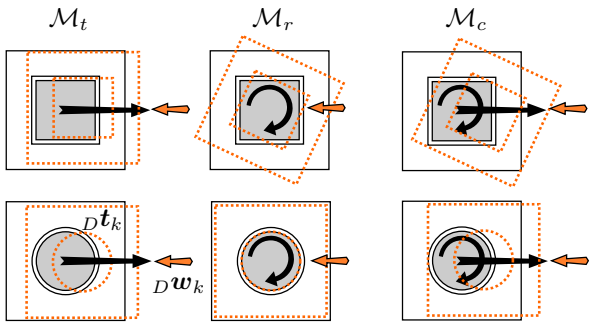


Fig. 6. Three movement types are investigated in the experiments: A pure translation $\mathcal{M}_t$, a pure rotation $\mathcal{M}_r$, and a combined motion $\mathcal{M}_c$. The dotted lines depict the motion of the sliding hole in the environment.

TABLE I
EXPERIMENTAL SETUP

workpieces:	square peg, round peg
movement types:	$\mathcal{M}_t$, $\mathcal{M}_r$, $\mathcal{M}_c$
propagation approaches:	velocity-based, constraint-based

The virtual stiffness of the passive part is set to zero in direction of the motion. The constraint directions are given a virtual mechanical impedance for the translation of 5000 N/m, and for the rotation 300 Nm/rad. An additional constant controller force of 10 N acts as a friction force against the motion of the active robot on the virtual table plane.

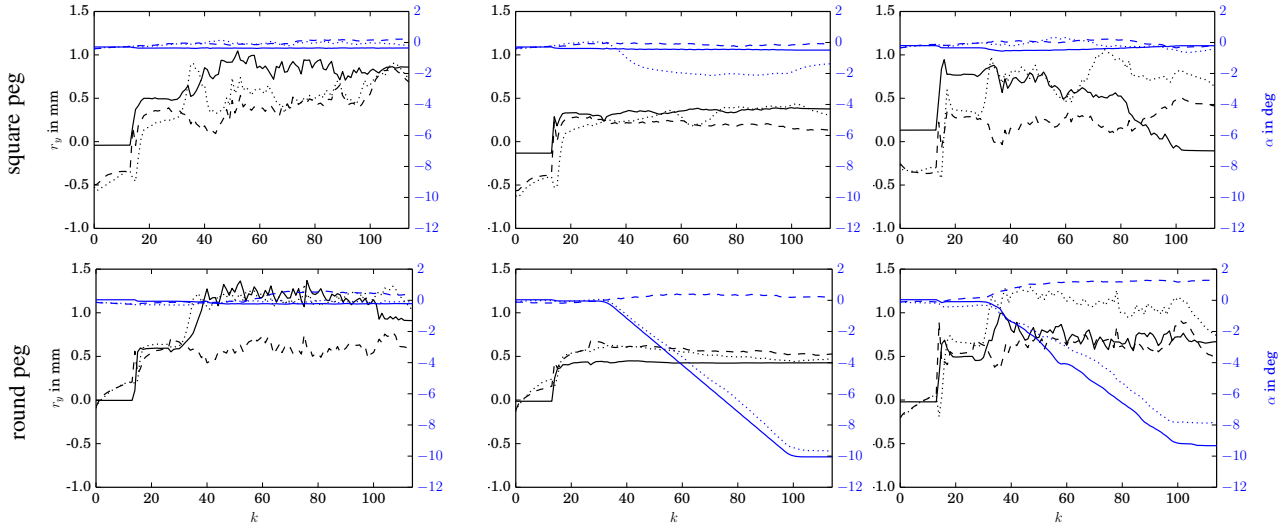
Evaluations are carried out for both presented propagation approaches. Table I summarizes the variables in the experimental setup with regard to workpieces, motions and propagation approaches.

In all runs of the evaluation, the same basic parameters for the probability distributions of the SMC algorithm are chosen. The initial samples $\{\boldsymbol{x}_0^{(i)}\}$ are drawn from a cubical uniform distribution around $\bar{C}$. The translational uncertainty in the hole plane (xy-plane of $\boldsymbol{H}_{B\bar{C}}$) is 2 mm and 20 mm along the z-axis. The rotational uncertainty is 1 deg in the parameters α, β and γ. We choose these parameters for the initial distribution for Part (ii), but they can also be retrieved from a perception system or from the preceding observation of the peg-in-hole insertion (i), from which a rough estimate would be available. The number of samples N is varied on a logarithmic scale $N \in \{10, 32, 100, 316, 1000, 3162\}$ and stays constant during the observation. Samples from the robotic execution are taken with $\Delta T_{rob} = 0.03$ s, whereas the cycle time of the observer is set to $\Delta T = 0.15$ s. The measurements are recorded and evaluated offline to compare the propagation methods on the same data sets.

In addition to the suggested propagation motion (27), an additional noise term against particle impoverishment is introduced in the propagation step:

$$ \boldsymbol{x}_k^{(i)} = \mathcal{N}(\bar{\boldsymbol{x}}_k^{(i)}, \Sigma_{\boldsymbol{x}}), \tag{31}$$

with a covariance matrix $\Sigma_{\boldsymbol{x}} = \mathrm{diag}(\sigma_t^2, \sigma_t^2, \sigma_t^2, \sigma_r^2, \sigma_r^2, \sigma_r^2)$ where $\sigma_t = 0.1$ mm and $\sigma_r = 0.1$ deg. Furthermore, we assume an independent and normal distributed error in the joint



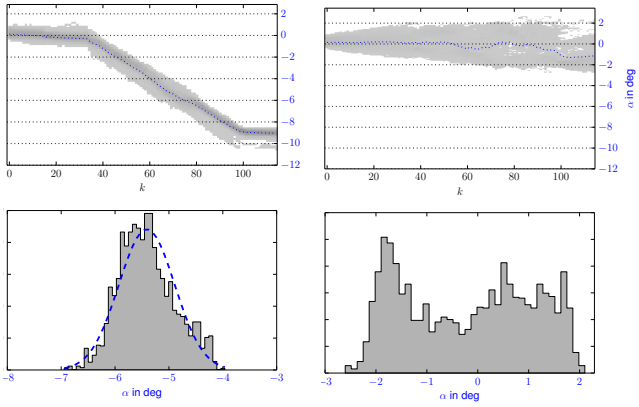
(a) $\mathcal{M}_t$ (b) $\mathcal{M}_r$ (c) $\mathcal{M}_c$

Fig. 7. Sequence of estimates for the relative transformation $\boldsymbol{H}_{CD}$ between EO and MO. Solid lines: ground truth; dashed lines: velocity-based propagation; dotted lines: constraint-based propagation. In black: translational movement r_y (mm); in blue: rotational movement α (deg).

torque measurements, thus, the consistency of the torque measurement (11) is weighted with $\Sigma_{\boldsymbol{\tau}} = \mathrm{diag}(\sigma_\tau^2, ..., \sigma_\tau^2)$ and σ_τ = 5000 Nmm. The virtual contact model is parameterized such that a penetration of d_t = 2.0 mm is feasible. Note that for the penalty based VPS, an intersection of MO and EO is required to compute the contact wrenches. The compliance with the configuration space is enforced with σ_d = 0.1 mm in (10).

B. Results

For the evaluation of the tracking performance, the ground truth $\boldsymbol{H}_{CD,ground,k}$ is computed from the kinematic chain between active and passive robot. Fig. 7 shows an exemplary sequence of estimates for the motions $\mathcal{M}_t$, $\mathcal{M}_r$ and $\mathcal{M}_c$ computed with $N = 1000$ samples. At each time step, an expected value $\langle \boldsymbol{H}_{CD,k} \rangle$ is computed for velocity- and constraint-based propagation. The ground truth value (solid lines) shows a step at $k \approx 14$ when the additional controller force is applied and backlash in the contact between MO and EO vanishes. The motion starts at $k \approx 28$ and the relative pose stays approximately constant. The estimates from the velocity-based (dashed) and constraint-based (dotted) propagation show in all cases a deviation from the ground truth < 0.5 mm in the relative position in y-direction (black lines). The error in the orientation around the hole axis (blue lines) is small in general (≈ 1 deg) but differs significantly for the evaluations with round pegs and rotational motions. In this case, the velocity-based method shows undesired behavior and is not able to track the ground truth. This is due to the fact that the samples always follow the robot motion no matter if the robot motion can actually induce a motion in the EO through the contact with the MO. This is obvious as the trivial solution (21) does not incorporate the constraints of the contact and the rotation about the axis of the round hole is not observable in the contact forces.

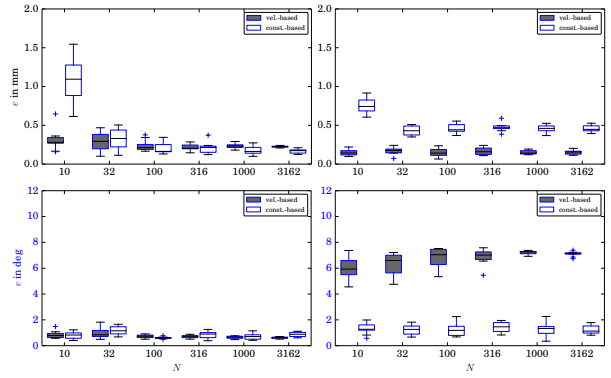


(a) square peg (b) round peg

Fig. 8. Sample evolution for the motion $\mathcal{M}_r$ with a constraint-based propagation. The grey values indicate the sample density. Note that the distribution is plotted relative to the assumed initial pose $\bar{C}$.

The principal effect of the knowledge about the twist space in the constraint-based solution (25) is clearly visible when we look at the sample distribution at a given instance of time. Fig. 8 shows the sample evolution for the motion $\mathcal{M}_r$ evaluated with the constraint-based propagation model. For the square peg, the rotation is observable and produces a unimodal and bound distribution of the samples at time k (the grey values indicate the sample density). On the other hand for the round peg, the rotation is not observable and consequently leads to a diffusion of the initial sample set in equal directions and a uniform shaped distribution. If we apply the velocity-based model instead, we would see both distributions appearing uniform.

Despite of this qualitative comparison, we also investigate



(a) square peg (b) round peg

Fig. 9. Reliability of the observation of motion $\mathcal{M}_c$ using different population sizes N. Each evaluation is repeated 10 times and the averaged error ϵ is shown (upper row: translational error (mm), lower row: rotational error (deg)). The filled grey boxes display the values for the velocity-based propagation and the empty boxes the constraint-based propagation.

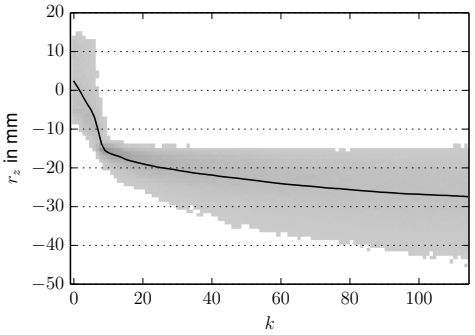


Fig. 10. Sample evolution for the motion $\mathcal{M}_t$ with a propagation model without information of the current robot motion and about the contact constraints (baseline propagation model). Grey color values describe the sample distribution, the black line gives the expected value of it.

the reliability of the observation in repeated evaluations of the same recorded data set. Fig. 9 visualizes the averaged error over time for multiple runs (each evaluation is repeated ten times). The average is computed from the expected values of $\{\boldsymbol{x}_k^{(i)}\}$ during the motion phase ($50 < k < 100$). In the case of the square workpieces, both methods show a high reliability even for a low number of particles and a high accuracy in the estimates. For the round workpieces, the mentioned effect of an undesired rotation of the samples is also revealed in the statistics of the velocity-based propagation (averaged error in rotation $\approx 6\,\text{deg}$). The averaged error in the position is constant for both methods but shows an offset of $\approx 0.3\,\text{mm}$, which is explained by the necessary intersection of the parts to compute the power of the violation that is more crucial in the case of the velocity-based propagation tracking round parts. Nevertheless, the method shows reliable results in the translational direction for a low number of particles.

Additionally, we examined the baseline propagation, i.e. a model in which the current robot motion is not incorporated and no direct knowledge about the constraints is available. We parameterize a pure random diffusion motion according to (31) with $\sigma_t = 1\,\text{mm}$ such that a sufficient amount of samples could follow the translational motion $\mathcal{M}_t$. Fig. 10 shows the state evolution in direction of the hole axis (z-axis of C). The samples evolve very quickly in the negative direction of the hole, and end in an unbound diffusion in free space, a further tracking of the motion in y-direction becomes impossible as in the virtual reference model, the peg and the hole are then separated. The weighting step does not prevent the samples to diffuse out of the hole as there is no force in this direction to penalize this behavior. Therefore, the diffusion would need to be adapted according to the contact constraints, i.e. make it small in direction of the hole and large in direction of the current robot motion. This insight supports again our proposal to incorporate the robot motion and the contact constraints within the propagation step.

C. Discussion

As clearly shown in the results, the information from the current robot motion is highly recommended to be used within the propagation step of a sample-based observation algorithm in the case that the objects in the environment have no fixed poses. Accurate tracking based on joint torque and positions measurements is achieved with an error in position considerably smaller than 1 mm and in rotation in the range of 1 deg. We show that the knowledge about the twist space in the contact improves the quality of the estimation when a relative motion between MO and EO in an unobservable degree-of-freedom is present. As summarized in Fig. 11, the constraint-based method has a clear advantage over the pure velocity-based method, because it does not induce motions of the EO in unobservable directions. The results are derived with simple and known geometrical shapes to get clear results on the differences. Nevertheless, it became also clear that a model of the twist space seems to be necessary to achieve this. While for the given geometry it is obvious to find the twist space analytically, a generalization is required to represent the twistspace for arbitrary shaped parts in real assembly applications. Existing approaches might be integrated here, e.g. from [4] in which twist- and wrenchspace are generated with a singular value decomposition of elemental contacts.

As it was already stated in [6], the performance of the SMC-based localization is dependent on the dimensionality of the uncertainty space, the number of required samples increases exponentially. Our suggested propagation model allows for an efficient tracking as the samples are placed in a narrowed search space dependent on the current robot motion. Currently an online observation is possible for tracking at a low observation rate $\approx 10\,\text{Hz}$. The cycle time is mainly dominated by the contact validation which takes $\approx 1\,\text{ms}$ for each sample. An advantage of the presented models is that they also achieve accurate tracking at this low rate.

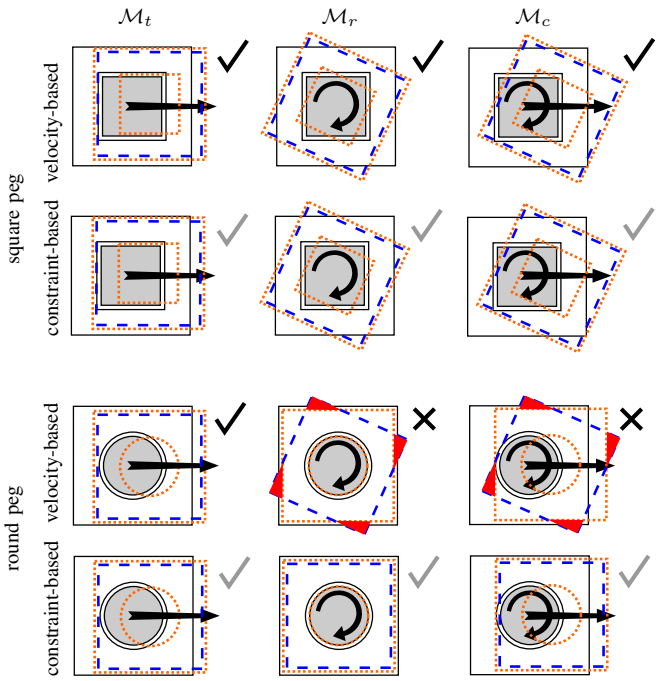


Fig. 11. Evaluation summary: constraint-based propagation is able to track the motion of the workpieces (real pose: orange dotted lines, estimated pose: blue dashed lines) without changing unobservable states, i.e. the rotation with respect to the hole axis. The velocity-based propagation leads to an unobservable motion of the EO which can be prevented if the twist space is modeled (constraint-based propagation).

V. Conclusion

This paper presents a new method to incorporate the motion of the robot arm and contact constraints in the propagation step of a sample-based observation of assembly tasks. We follow the idea of time domain passivity control: if virtual energy is generated by the samples, a compensation motion is computed which takes into account the measured robot velocity. A velocity-based and a constraint-based propagation model are derived. The approach is evaluated systematically with a dual-arm setup for two given peg-in hole scenarios where the environment object slides on a virtual table surface. Principal differences in the quality of the tracking were described according to the availability of the twist space in the contact. We showed that both methods for propagation could accurately track the observable motion of the workpieces.

In future work, methods to generate a representation of the twist space on-the-fly have to be integrated, especially when unobservable relative motions between the workpieces can occur. The presented work is also a basis for the fusion of torque sensor data with additional sensory information, e.g. vision will be used as a complement to increase the observable uncertainty space. Currently, we focus on the observation of the assembly process, but we plan to show the importance of the improved estimates again in a closed loop assembly application.

Acknowledgment

We thank Freek Stulp, Andreas Stemmer and Michael Kaßecker for providing helpful comments and suggestions for improving the paper and Mikel Sagardia for providing the VPS algorithm.

References

[1] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. De Gersem, "Active compliant motion: A survey," *Advanced Robotics*, vol. 19, no. 5, pp. 479–499, 2005.

[2] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: fifteen years later," *Handbook of Nonlinear Filtering*, vol. 12, pp. 656–704, 2009.

[3] S. Chhatpar and M. Branicky, "Localization for robotic assemblies using probing and particle filtering," in *Adv. Intelligent Mechatronics. Proc., 2005 IEEE/ASME Int. Conf. on*, July 2005, pp. 1379–1384.

[4] W. Meeussen, J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. De Schutter, "Contact-state segmentation using particle filters for programming by human demonstration in compliant-motion tasks," *IEEE Trans. on Robotics*, vol. 23, no. 2, pp. 218–231, 2007.

[5] U. Thomas, S. Molkenstruck, R. Iser, and F. Wahl, "Multi sensor fusion in robot assembly using particle filters," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, April 2007, pp. 3837–3843.

[6] Y. Taguchi, T. Marks, and H. Okuda, "Rao-blackwellized particle filtering for probing-based 6-dof localization in robotic assembly," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2010, pp. 2610–2617.

[7] K. Nottensteiner, M. Sagardia, A. Stemmer, and C. Borst, "Narrow passage sampling in the observation of robotic assembly tasks," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016, pp. 130–137.

[8] J.-H. Ryu, Y. S. Kim, and B. Hannaford, "Sampled- and continuous-time passivity and stability of virtual environments," *IEEE Trans. on Robotics*, vol. 20, no. 4, pp. 772–776, 2004.

[9] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, "The DLR lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: An Int. Journal*, vol. 34, no. 5, pp. 376–385, 2007.

[10] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decr, R. Smits, Aertbelin, K. Claes, and H. Bruyninckx, "Constrained-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The Int. Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[11] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[12] M. Sagardia, T. Stouraitis, and J. a. Lopes e Silva, "A new fast and robust collision detection and force computation algorithm applied to the physics engine bullet," in *Conf. and Exhibition of the European Association of Virtual and Augmented Reality (EuroVR)*, 2014.

[13] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *ACM SIGGRAPH 2005 Courses*, ser. SIGGRAPH '05. New York, USA: ACM, 2005.

[14] K. H. Hunt, "Review: Don't cross-thread the screw," *Journal of Robotic Systems*, vol. 20, no. 7, pp. 317–339, 2003.

[15] R. S. Ball, *A Treatise on the Theory of Screws*. Cambridge University Press, 1900.

[16] H. Bruyninckx, J. De Schutter, and S. Dutr, "The 'reciprocity' and 'consistency' based approaches to uncertainty identification for compliant motions," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1993, pp. 349 – 354.

[17] M. S. Ohwovoriole and B. Roth, "An extension of screw theory," *Journal of Mechanical Design*, vol. 103, pp. 725–735, 1981.

[18] K. Hertkorn, T. Hulin, P. Kremer, C. Preusche, and G. Hirzinger, "Time domain passivity control for multi-degree of freedom haptic devices with time delay," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 1313–1319.

[19] K. L. Doty, C. Melchiorri, and C. Bonivento, "A theory of generalized inverses applied to robotics," *The Int. Journal of Robotics Research*, vol. 12, no. 1, 1993.

Publication 4

Nottensteiner, K., A. Sachtler, and A. Albu-Schäffer (2021): "*Towards Autonomous Robotic Assembly: Using Combined Visual and Tactile Sensing for Adaptive Task Execution*". Springer Journal of Intelligent & Robotic Systems (JINT). Springer, 2021.

Version Note

The following attached version corresponds to the final published version of the publication.

The final published version is also available under:

- https://link.springer.com/article/10.1007/s10846-020-01303-z

Please refer to the final published version for citation:

```
@Article{Nottensteiner2021,
  author =      {Korbinian Nottensteiner and Arne Sachtler and Alin
                 Albu-Schäffer},
  title =       {Towards Autonomous Robotic Assembly: Using Combined
                 Visual and Tactile Sensing for Adaptive Task
                 Execution},
  journaltitle = {Springer Journal of Intelligent & Robotic Systems
                 (JINT)},
  year =        2021,
  month =       3,
  volume =      101,
  issue =       3,
  number =      49,
  doi =         {10.1007/s10846-020-01303-z}}
```

Open Access

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

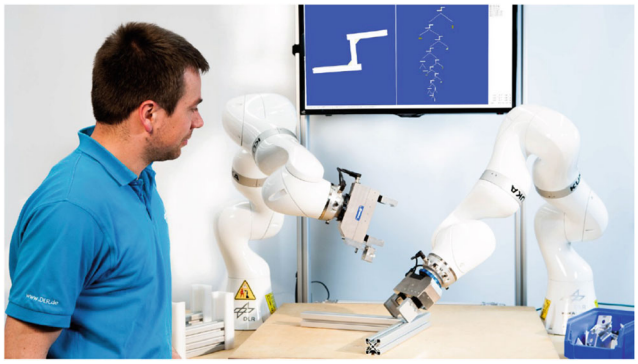


Fig. 1 Autonomous assembly of aluminum profile structures with a dual-arm robotic system without specialized holders

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 $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \propto p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$.

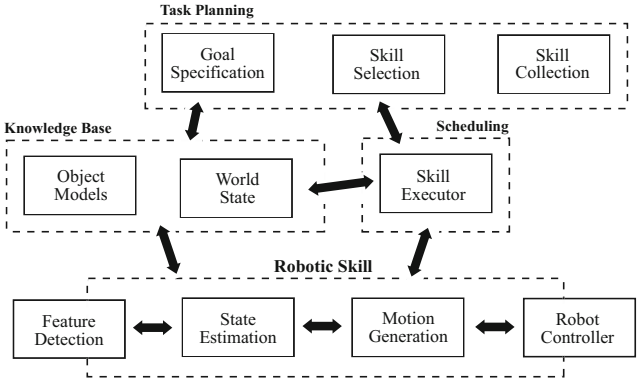


Fig. 2 Components of a robotic system for autonomous planning and adaptive execution of assembly tasks

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

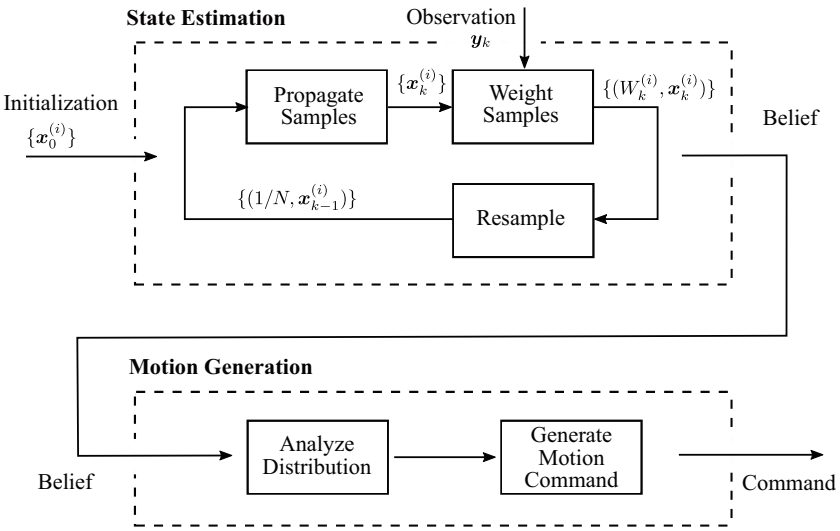


Fig. 3 State estimation and motion generation components of the assembly skill. Recursive Bayesian state estimation provides the belief state, which is then analyzed for generating motion commands

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.

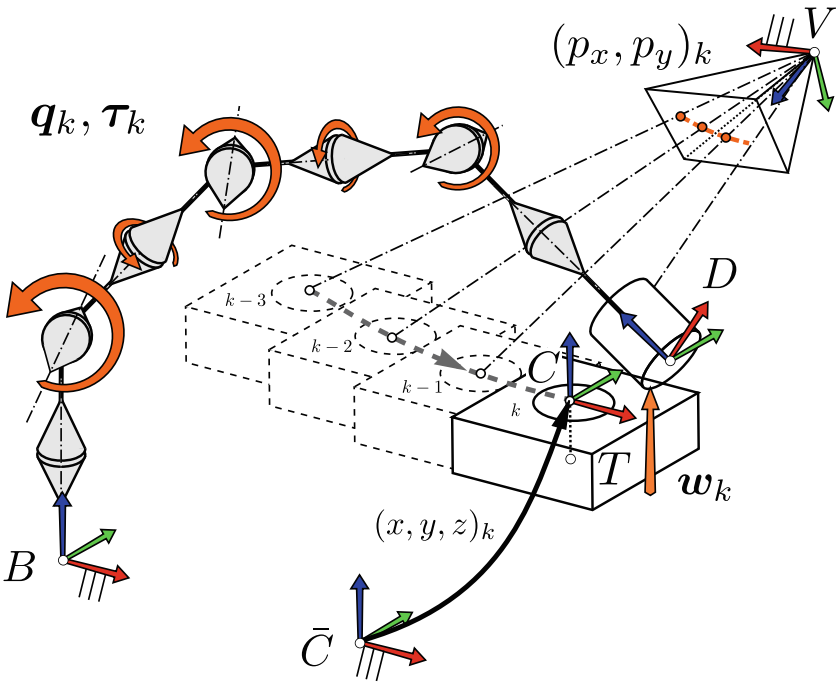


Fig. 4 Definition of frames and variables in the considered scenario. A peg with reference frame D is rigidly attached to a manipulator with base frame B. The position $(x, y, z)_k$ of a moving hole C at time $t = t_k$ is uncertain with respect to a known reference frame $\bar{C}$. The task is to transfer 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

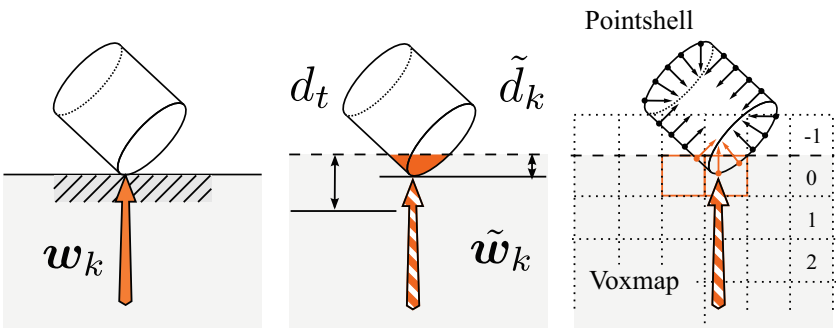


Fig. 5 Left: contact situation with contact wrench $\boldsymbol{w}_k$. Center: penalty-based contact model. Right: implementation of contact model with a voxelmap and pointshell representation of the objects. $\tilde{d}_k$ denotes the contact distance and d_t a threshold on the maximal feasible virtual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

Algorithm 1 Propagation model.

function PROPAGATESAMPLE($\boldsymbol{x}_{k-1}{}^{(\mathrm{i})}, \boldsymbol{q}_k$)
  $\boldsymbol{x}_I \leftarrow$ CONSTANTVELOCITY($\boldsymbol{x}_{k-1}{}^{(\mathrm{i})}$) ▷ (3)
  for j := 1 to L_{max} **do**
    draw $\boldsymbol{p}_{II} \sim \mathcal{N}(\boldsymbol{p}_I, \boldsymbol{\Sigma}_{p,b})$.
    $\mathcal{C}_{II} \leftarrow$ 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 $\boldsymbol{p}_{IV} \sim \mathcal{N}(\boldsymbol{p}_I, \boldsymbol{\Sigma}_{p,p})$.
  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 $\boldsymbol{\Sigma}_{p,b}$ 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 $\boldsymbol{\Sigma}_{p,p}$ 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 $\boldsymbol{H}_{BV} = \text{const.} \in SE(3)$.

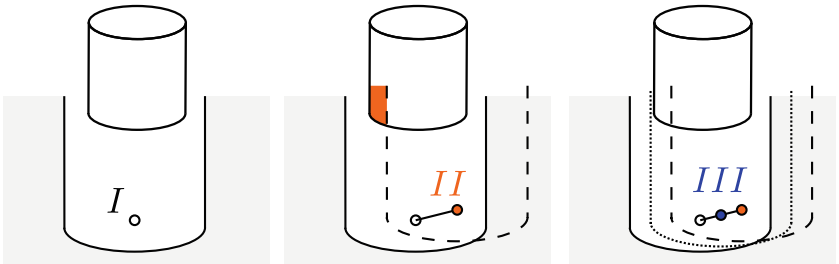


Fig. 6 The bridge test policy in three steps. An auxiliary sample with frame II is drawn in the neighborhood of the original sample frame I in order to find so-called bridge points with frame III in the configuration space, which is located at half-distance between I and II

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.

1: **function** WEIGHTSAMPLE($\boldsymbol{y}_k^{s_v}$, $\boldsymbol{y}_k^{s_t}$, $\boldsymbol{x}_k^{(i)}$)
2: $a \leftarrow$ TACTILELIKELIHOOD($\boldsymbol{y}_k^{s_t}$, $\boldsymbol{x}_k^{(i)}$)
3: $b \leftarrow$ VISUALLIKELIHOOD($\boldsymbol{y}_k^{s_v}$, $\boldsymbol{x}_k^{(i)}$)
4: weight $\leftarrow \ln a + \ln b$ ▷ update particle weight
5: **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

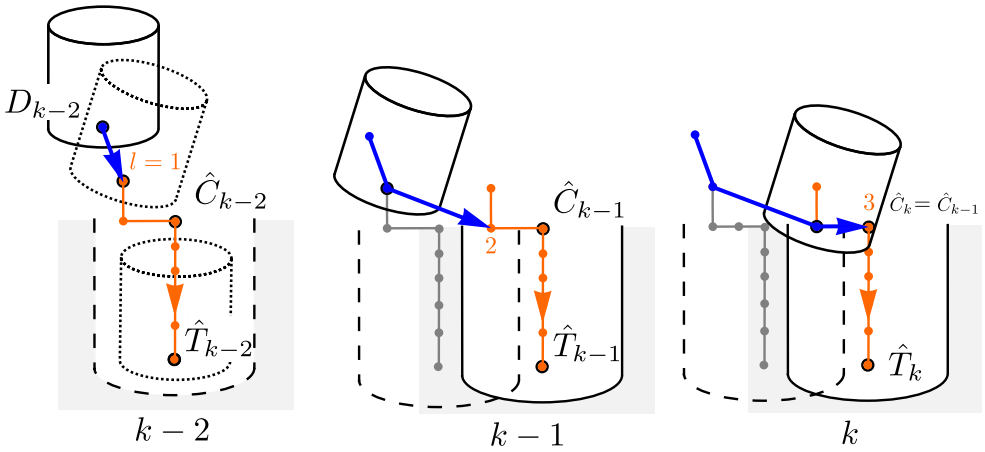


Fig. 7 Adaptive execution of an object-centric path (orange line) considering the currently estimated frame of the hole $\hat{C}_k$. The hole moves to the right between time step $k-2$ (left) and $k-1$ (center). The motion commands (blue lines) follow the estimated poses

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.

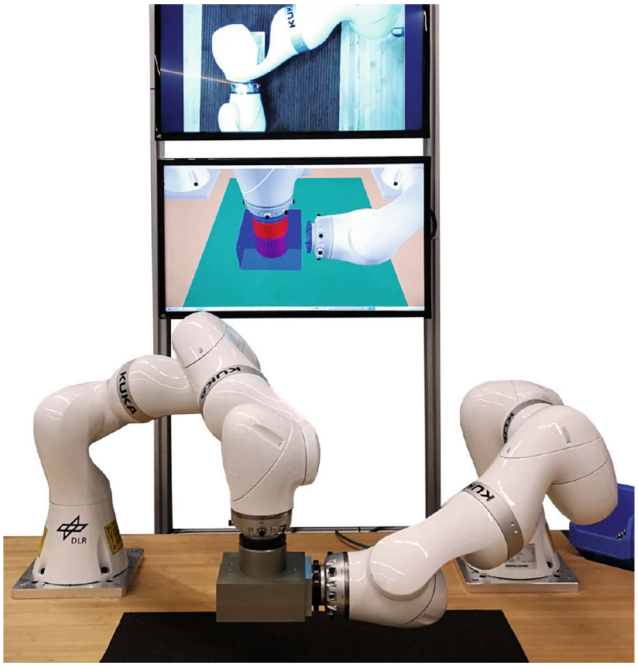


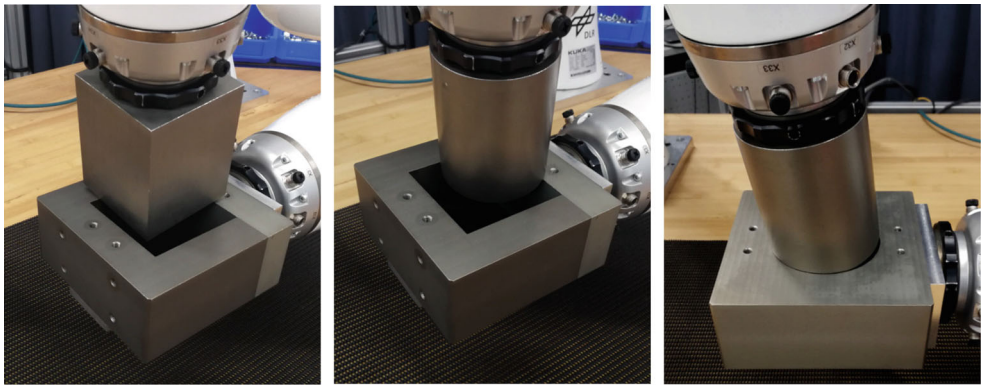
Fig. 8 Setup for the peg-in-hole experiments. The left arm executes the assembly skill. The right arm is used as a ground truth measurement device and simulates the hole motions. The camera image is visible in the upper screen, the lower screen shows the live view of the world model

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	
$\boldsymbol{\Sigma}_{\boldsymbol{x}}$	diag [0, 0.1, 0, 0.1, 0, 0]	mm, mms^{-1}
$\boldsymbol{\Sigma}_{\boldsymbol{p},b}$	diag [6, 6, 3]	mm
$\boldsymbol{\Sigma}_{\boldsymbol{p},p}$	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
$\boldsymbol{\Sigma}_{\boldsymbol{\tau}}$	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

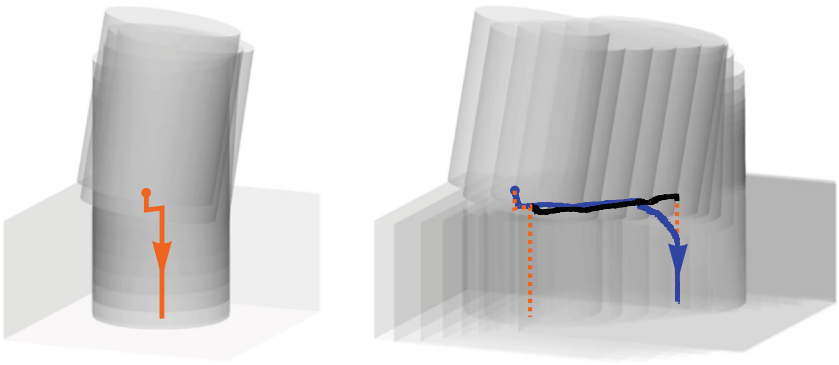


Fig. 10 Nominal object-centric peg motion following a tilt-and-align strategy (left) and finally executed peg motion (right). The nominal path is drawn in orange, the blue line represents the executed path of the peg reference frame, the black line the online estimated pose of the hole to which the motion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

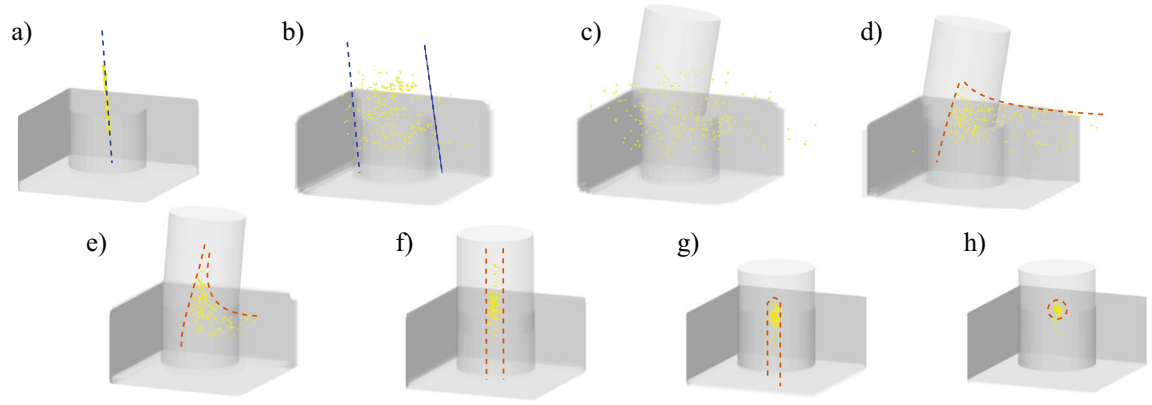
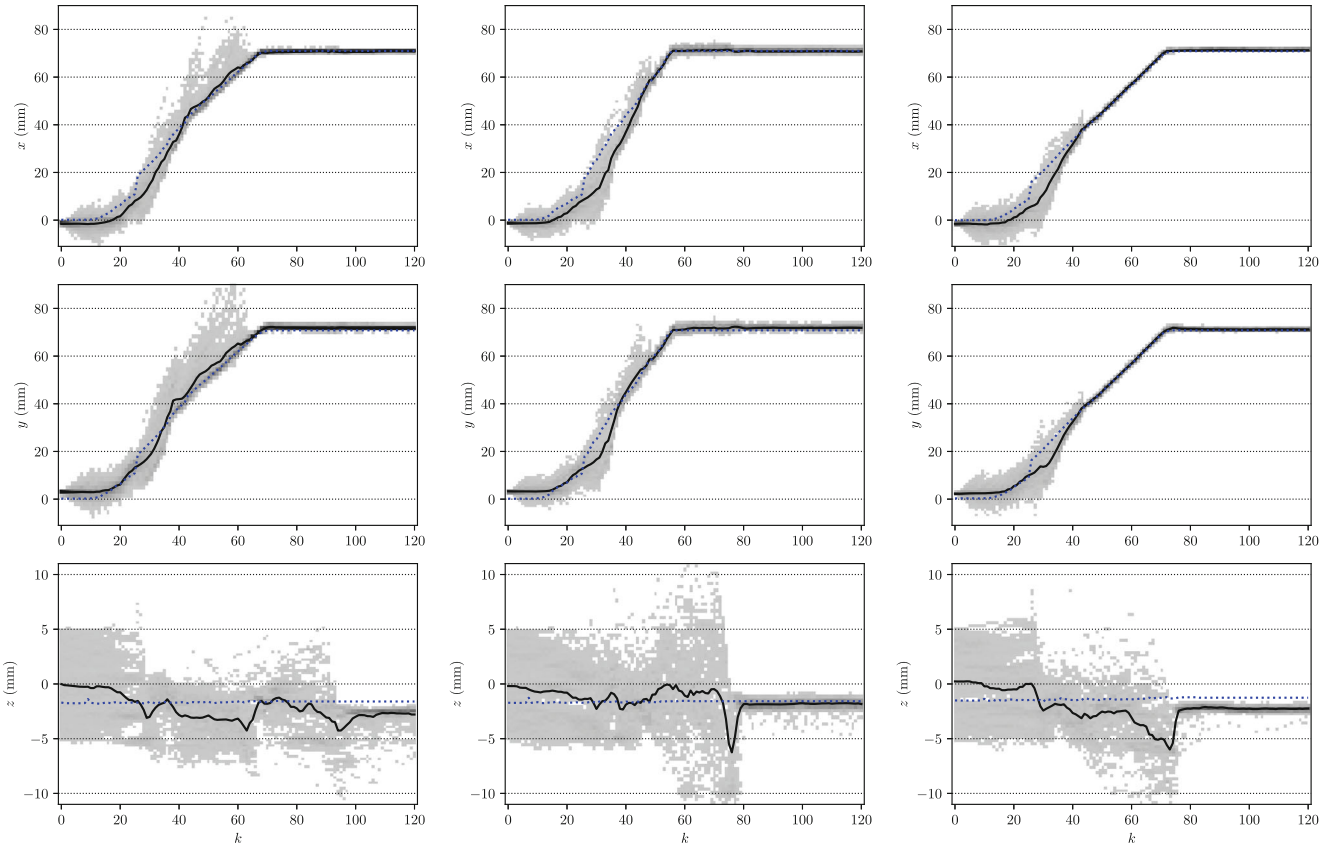


Fig. 11 Evolution of the sample distribution for the cylindrical peg-in-hole ($\mathcal{P}_\circ$). Yellow dots represent the origins of possible hole frames. For each dot the hole geometry is additionally rendered. The peg is displayed with its measured pose. **a** At first, the samples are initialized 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

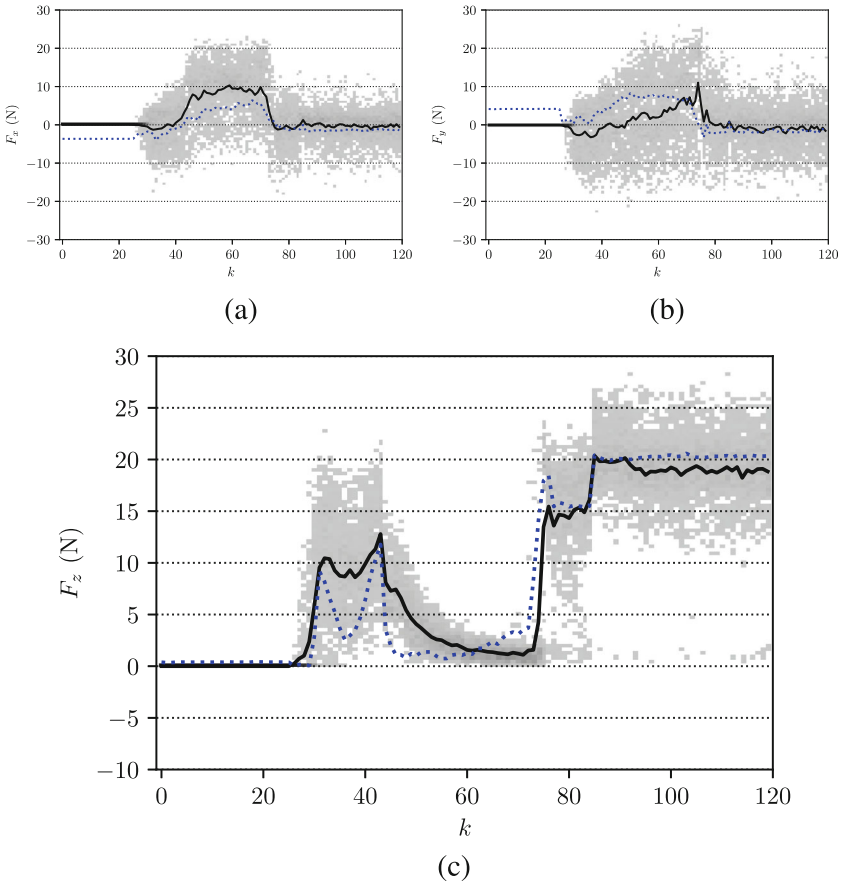


Fig. 13 Measured force (dashed blue line approximated from the joint torque measures using a pseudo inverse of the Jacobian) and force distribution represented by the samples for a run of $\mathcal{P}_\circ$. For each sample the virtual contact force is computed with respect to the peg frame D. The density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

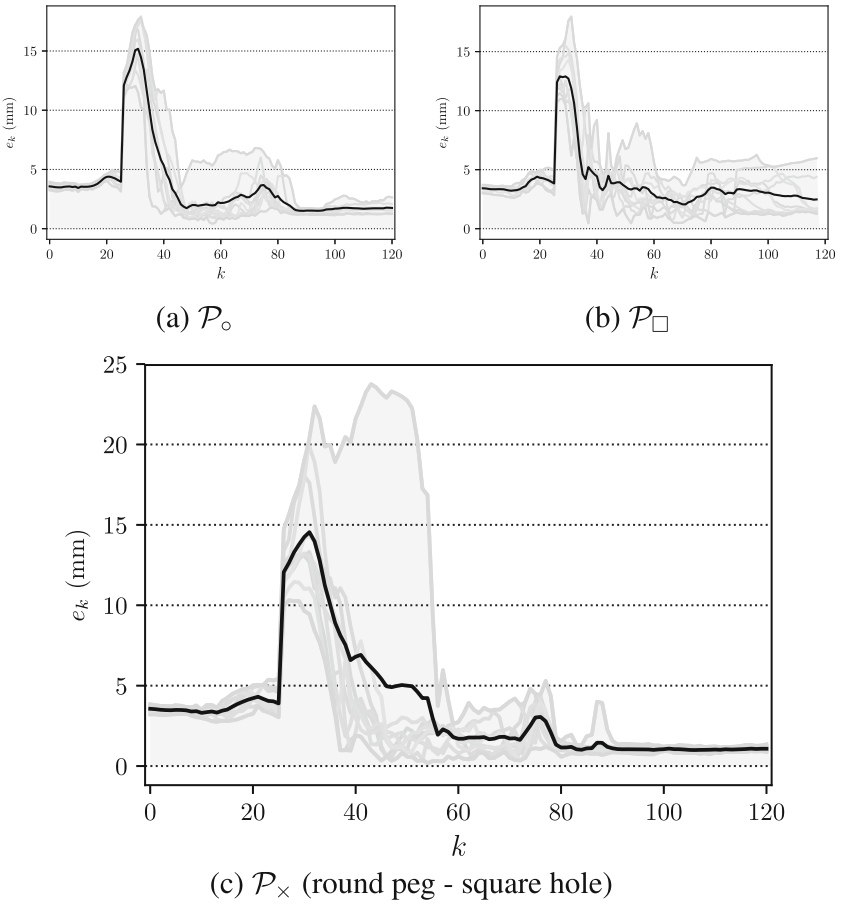


Fig. 14 Pose estimation error over time, computed from the ground truth measurement and the expected value of the sample distribution. Each light gray line represents a single run. The black line is the average of the error over all 10 runs for each case

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

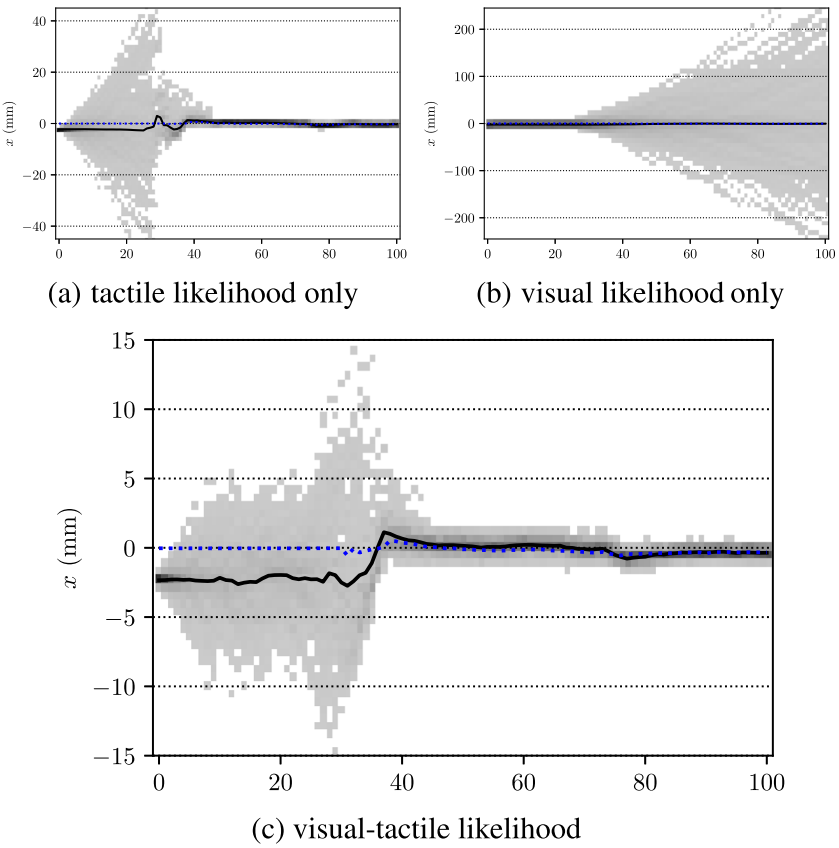


Fig. 15 Sample evolution of the x-component of the state in the case of a cylindrical peg-in-hole ($\mathcal{P}_\circ$) with static hole using three variants of likelihood functions. The gray-value indicates the sample density, the black line corresponds to the expected value and the blue dotted line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

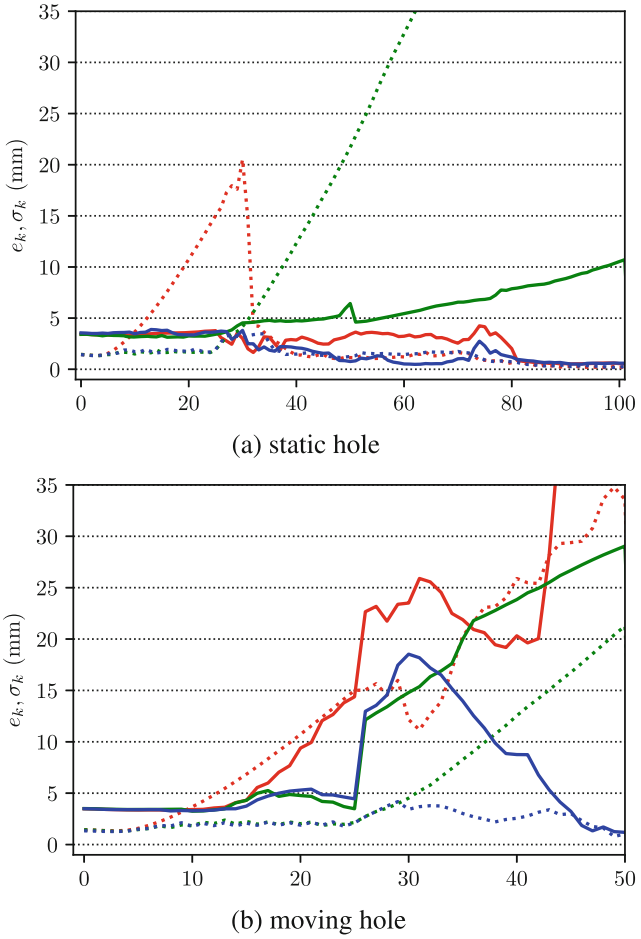


Fig. 16 Pose estimation error over time (solid) and spread of the current sample distribution (dotted line) given as standard deviation for the case of using a tactile (red), visual (green) or visual-tactile likelihood (blue). The values are plotted for $\mathcal{P}_\circ$ and for the case of a static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

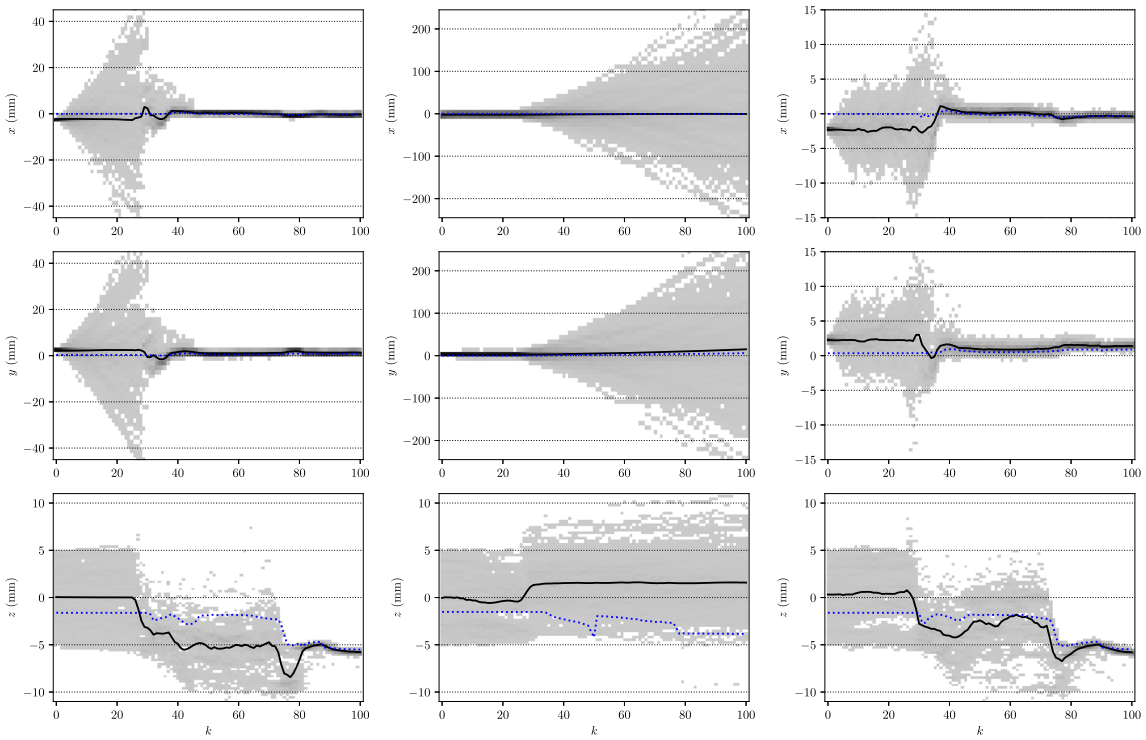
Although the filter step is computationally more expensive than in alternative approaches, an advantage is that the image of the local configuration space can be approximated by the sample distribution, and it is geometrically interpretable. A possible future extension of the presented work is to adapt 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.

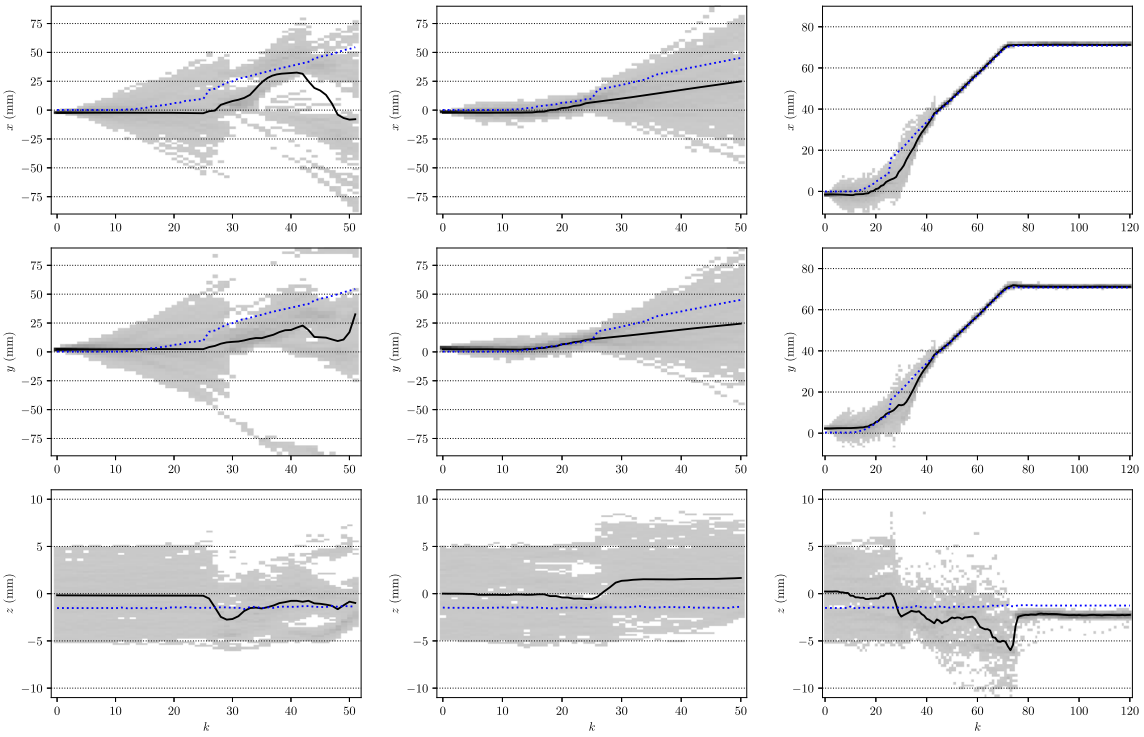
Appendix

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.

Publication 5

K. Nottensteiner, F. Stulp, and A. Albu-Schäffer (2020): "*Robust Locally Guided Peg-in-hole with Impedance Controlled Robots*". IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020.

Version Note

The following attached version corresponds to the accepted manuscript of the publication.

The final published version is available under:

- https://ieeexplore.ieee.org/document/9196986

Please refer to the final published version for citation:

```
@InProceedings{Nottensteiner2020,
  author =      {Korbinian Nottensteiner and Freek Stulp and Alin
                 Albu-Schäffer},
  title =       {Robust Locally Guided Peg-in-hole with Impedance
                 Controlled Robots},
  booktitle =   {IEEE International Conference on Robotics and
                 Automation (ICRA)},
  year =        2020,
  venue =       {Paris, France (online)},
  eventdate =   {2020-05-31/2020-08-31},
  publisher =   {IEEE},
  pages =       {5771-5777},
  isbn =        {978-1728173955},
  eid =         {2-s2.0-85092733172},
  doi =         {10.1109/ICRA40945.2020.9196986}}
```

Copyright Note

Robust, Locally Guided Peg-in-Hole using Impedance-Controlled Robots

Korbinian Nottensteiner[1], Freek Stulp[1], Alin Albu-Schäffer[1,2]

***Abstract*— We present an approach for the autonomous, robust execution of peg-in-hole assembly tasks. We build on a sampling-based state estimation framework, in which samples are weighted according to their consistency with the position and joint torque measurements. The key idea is to reuse these samples in a motion generation step, where they are assigned a second task-specific weight. The algorithm thereby guides the peg towards the goal along the configuration space. An advantage of the approach is that the user only needs to provide: the geometry of the objects as mesh data, as well as a rough estimate of the object poses in the workspace, and a desired goal state. Another advantage is that the local, online nature of our algorithm leads to robust behavior under uncertainty. The approach is validated in the case of our robotic setup and under varying uncertainties for the classical peg-in-hole problem subject to two different geometries.**

I. Introduction

Increasing the autonomy and robustness of assembly tasks is important for reducing implementation and set-up times in future manufacturing scenarios. Some challenging aspects of peg-in-hole assembly include the transitions from free to highly constrained motion, and jamming effects that arise due to alignment errors exacerbated by pose uncertainties. In this paper, we present an approach that does not require predefined strategies, and incorporates the currently observed relative configuration space to guide the peg to a goal pose. The user provides the geometry of the objects as mesh data, as well as a rough estimate of the hole pose in the base frame, and a desired goal state at the end of execution, e.g. pose of peg relative to the hole. As these three are usually known, this substantially lowers the implementation effort for novel assembly tasks. The method is online and local, which means that it can react to the current execution state.

Our approach is a natural extension of the sample-based observation algorithm presented in our previous work [1]. For state estimation, samples of possible hole poses are weighted according to their consistency with the current measurements of the robot. In this work, our main novel contribution is to extend the state estimation step with a motion generation step, where the same samples are reused to determine the motion (see Fig. 1). This is done by weighting the samples with a task-specific cost function, which is related to the distance between the current and goal state. After presenting related work in the next section, we describe the two main components of the framework: sample-based state estimation (Section III) and sample-based motion generation with local guidance (Section IV). The results of the experiments are presented and discussed in Section V.

[1] German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchner Str. 20, 82234 Weßling, Germany

[2] Technical University Munich (TUM), Chair of Sensor Based Robotic Systems and Intelligent Assistance Systems, 85748 Garching, Germany

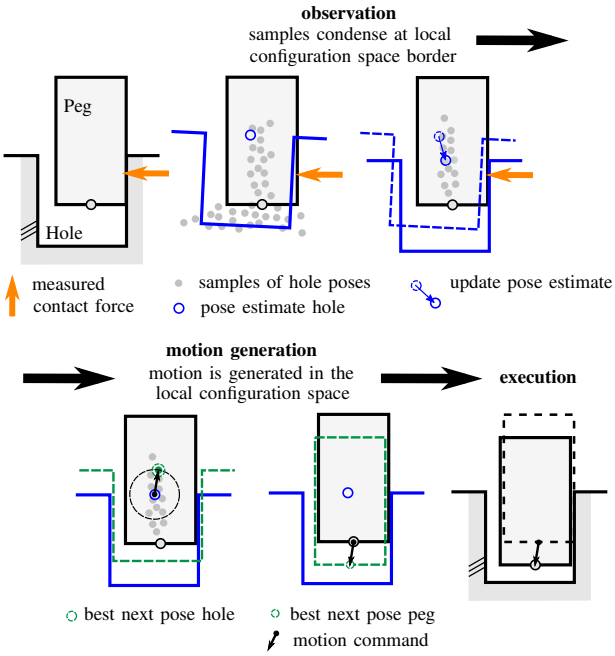


Fig. 1. Combined sample-based state estimation and motion generation model. The estimation algorithm sequentially propagates and weights samples (gray dots) according to their consistency with the sensory information of the robot and the geometry of the parts. At a second stage, a motion command is generated that can be executed with impedance control.

II. Related Work

Kinestatic models of peg-in-hole. Prior knowledge about object geometry allows us to formulate exact analytical models for peg-in-hole. Bruyninckx et al. [2] derive a model-based solution with a kinematic description of occurring contacts. A given sequence of actions is then executed to search for the hole, align the peg with the hole axis and insert it. The strategy can deal with large uncertainties, but requires the contact states to be modeled geometrically in advance. A detailed algebraic description of kinestatic models is provided by Staffetti et al. [3]. Later, Tang et al. [4] reuse the model of [2] and combine it with an estimator in order to achieve a robust alignment of peg and hole in the case of 3-point contact. While establishing theoretical foundations and working practically, the methods are not easy to generalize to more complex part geometries.

Strategies planned offline. A robust strategy for assembling complex-shaped planar parts is presented by Stemmer et al. [5]. The method analyses the geometry of the parts, and automatically generates a tilt-and-align mating action that is robust towards pose uncertainties, and which is

executed with an impedance controller. It was shown that it performs very well, but misses explicit feedback during the execution, which is necessary to react on unforeseen events. Wirnshofer et al. [6] use an advanced belief space planner to find optimal trajectories for compliant assembly under the presence of uncertainties before execution. So called "blind-search" strategies can be seen as a special form of offline planning. A motion pattern is selected in advance to deal with the expected uncertainties, and the time required for the search is often seen as a disadvantage [7]. The drawback of all offline planned strategies is the limited possibility of reasoning about the current contact during the execution.

Online detection of contact states. Various methods to make the robotic system aware of its current contact situation were developed in the field of contact state detection, e.g. [8], [9]. In particular, Meeussen et al. [9] present an approach that reliably detects contact formations (regions on the manifold of the configuration space obstacle) of polyhedral objects, using particle filtering and contact state graphs. The knowledge of the contact state can be used for the generation and execution of compliant motions [10], [11]. Unfortunately, it is not clear how the approach scales up to more complex geometrical objects where a high number of possible discrete contact states exist. A more specific controller, but following a similar concept, is presented by [12] for folding assembly tasks. In this approach, the controller can switch between two contact situations according to a simultaneous estimation of the contact point with a Kalman filter.

Uncertainty sampling methods. The estimation of uncertainties without modeled or discrete contact states is often performed with particle filtering techniques [1], [13]–[15]. The drawback of these is the substantial computational effort required to propagate and weight the large number of samples. Nevertheless, the approaches based on particle filters were further applied to actively reduce uncertainties as a result of motion selection. This applies, for example, to touch-based, online object registration by Taguchi et al. [16], or offline during the planning stage [17]. Andre et al. [18] demonstrated an active compensation of position errors using a particle filter. At the end of each filter step, a motion towards the sample with the best weight according to the measurements is commanded. While this approach can localize and compensate motion errors, it lacks a task-specific weighting of the samples as suggested in our work, which allows a goal to be specified for the task independently from the estimators' objective to undertake localization. Recently, Wirnshofer et al.[19] successfully extended their offline planning method using online Bayesian reasoning, and evaluated it in multiple scenarios with non-prehensile manipulation and peg-in-hole. In the provided real-world experiment, simple-shaped part geometries are used and uncertainties are reduced largely before the actual insertion starts. Nevertheless, they demonstrate successfully the real-time capabilities of this class of approaches.

Online parameter optimization and learning. Another class of approaches for peg-in-hole focuses on the parameter optimization of a given or demonstrated strategy. For example, [20] present a controller-tuning method that uses constrained Bayesian optimization to improve the performance of establishing and maintaining contact in manipulation tasks. Furthermore, [21] analyze different methods for parameter learning in peg-in-hole tasks and show how their approach can outperform humans in execution time. Nevertheless, they use a predefined strategy, a so-called manipulation primitive for peg-in-hole comparable to that of [2], and start the learning phase with an accurately known hole pose in which the execution is made robust through slightly varying poses. In contrast, our work does not assume a predefined motion strategy, and thus is applicable for various complex manipulation tasks.

Deep learning and self-supervised learning. Deep learning has been applied to peg-in-hole tasks [22]–[25]. While showing promising results, it is not yet clear how well those methods generalize to varying tasks and larger uncertainties. Lee et al. [25] demonstrate robustness in various situations and different shapes with a self-supervised learning approach, but their approach relies on visual feedback to complete insertion at a high success rate. To date, the performance of learning approaches has been strongly dependent on the available datasets from (extensive) training phases which can not always be afforded in real applications. In contrast, our aim in the presented work is to achieve a successful run with the first attempt, using the local feedback from the observation and known geometry information.

III. Sample-based State Estimation

In this work, we combine state estimation and motion generation in an integrated sample-based framework. The algorithm reuses samples of the configuration space from the state estimator to compute the next motion command. The following section introduces the robot and uncertainty model, and briefly describes the state estimation approach from [1].

A. Robot and Uncertainty Model

We consider robots with $n \geq 6$ degrees of freedom, which are equipped with joint torque sensors and support impedance control [26]. At each time step k, the robot provides measurements of the joint position $\boldsymbol{q}_k \in \mathbb{R}^n$ and the external joint torque $\boldsymbol{\tau}_k \in \mathbb{R}^n$. The forward kinematics of the arm is given by a homogeneous transformation $\boldsymbol{H}_{BA,k} = \boldsymbol{H}_{BA}(\boldsymbol{q}_k) \in SE(3)$, where B denotes the frame of the robot base and A the end effector. A rigid object is attached to the flange with a known transformation $\boldsymbol{H}_{AD} = \text{const.} \in SE(3)$ with the object frame D. We assume a quasi-static robot model and rigid links with known dimensions. The given task is to transfer the peg from a start frame to a desired target frame T specified with respect to a hole in the environment with reference frame C. We define D to be located at the bottom of the peg, and T at the bottom of the hole. Both the geometries of the peg and the hole are known and given as mesh data. The pose of the hole relative to the robot given as $\boldsymbol{H}_{BC} \in SE(3)$ is assumed to be constant, but subject to uncertainty. We denote the frame of the initially assumed pose of the hole as $\bar{C}$. For example, this pose can be obtained by a teach-in process

or by visual object detection. The geometric uncertainty is represented by a hidden state $\boldsymbol{x}$. The real pose of the hole can then be written as $\boldsymbol{H}_{BC}(\boldsymbol{x}) = \boldsymbol{H}_{B\bar{C}}\boldsymbol{H}_{\bar{C}C}(\boldsymbol{x})$. We choose translational components and Z-Y-X Euler angles [27, p. 43] as parameterization for $\boldsymbol{H}_{\bar{C}C}$ and consequently as variables for a minimal state representation $\boldsymbol{x} = (x, y, z, \alpha, \beta, \gamma)$. This corresponds to a possible parameterization of the configuration space of a solid object [28, p. 682]. Small orientational uncertainties $\ll 90\text{deg}$ are considered to avoid singularities and other problems connected to Euler angles, and to allow for convergence of the local policy with respect to the orientational part.

B. Torque Consistent Contact Observation

The state estimator implements a sequential Monte Carlo (SMC) algorithm [29] that approximates the distribution of a hidden state $\boldsymbol{x}$ through sampled density functions, taking into account past measurements $\boldsymbol{y}_{0:k}$. An important advantage of SMC is that nonlinear distributions and models can be handled without the explicit knowledge of analytical model functions or the need for linearizations. The initial uncertainty at time $t = 0$ is represented by a set of N samples $\{\boldsymbol{x}_0{}^{(0)}, \boldsymbol{x}_0{}^{(1)}, \ldots, \boldsymbol{x}_0{}^{(\mathrm{N})}\}$ drawn from the initial probability density $p(\boldsymbol{x}_0)$. At each time step, the samples are propagated with the importance density $q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}{}^{(\mathrm{i})}, \boldsymbol{y}_k)$, updated according to the consistency with the new measurements $\boldsymbol{y}_k$ at time $t = t_k$, and finally resampled. The observation density function $p_{\boldsymbol{y}}(\boldsymbol{y}_k|\boldsymbol{x}_k{}^{(\mathrm{i})})$ assigns weights $W_k{}^{(\mathrm{i})}$ for the approximation of $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$ and the resampling.

The configuration space between peg and hole is approximated locally by the samples generated by the estimator. An estimated pose $\hat{C}$ can be obtained by computing the expected value $\hat{\boldsymbol{x}}_k$ of the samples. In this work, we apply the importance and observation density functions presented in [1]. The samples are propagated with the *bridge test policy*, which pulls samples into narrow passages of the configuration space. The update is performed using a comparison of the current joint position and external torque measurements $\boldsymbol{y_k} = (\boldsymbol{q}_k, \boldsymbol{\tau}_k)$ of the robot with a virtual contact model. We use a fast and accurate penalty-based collision detection algorithm for the contact force and distance computation, and assume Gaussian errors in the residuals. The object geometries are efficiently represented with voxelmaps and pointshells as known from haptic rendering [30]. A similar approach is also taken in the work of [18]. Our contact model is a rigorous simplification of the real physical effects; the contact stiffness is chosen such that the contact wrenches are reproduceable in magnitude and friction is neglected. Nevertheless, the model provides enough directional information to distinguish certain contact states.

IV. Sample-based Motion Generation

A. Task-Specific Sample Weighting

In our previous work [1], we pointed out that the samples $\boldsymbol{x}_k{}^{(\mathrm{i})}$ condense at the border of the relative configuration space between the objects in contact. Under the assumption that the motion of the robot is only constrained by the contact, then the distribution of the samples $\boldsymbol{x}_k{}^{(\mathrm{i})}$ captures

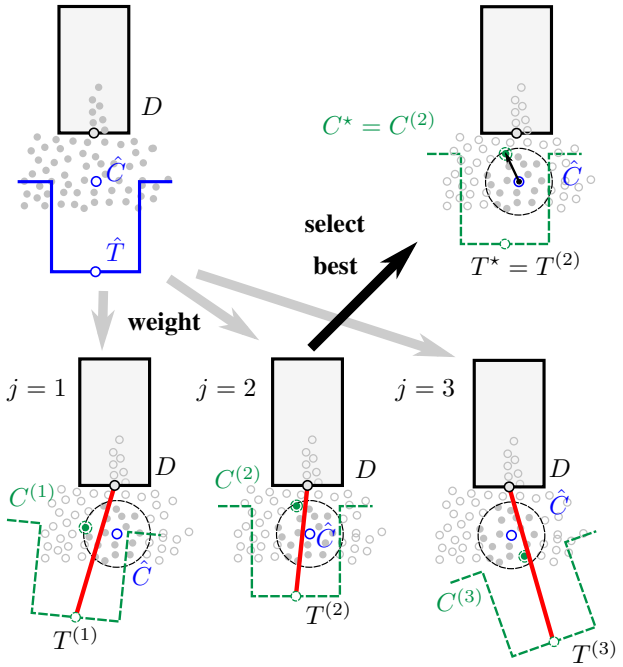


Fig. 2. The samples are evaluated in a neighborhood (dashed circle) of the estimated frame $\hat{C}$. For each sample, the distance of D_k to the goal $T^{(i)}$ is computed (red lines). The sample with the lowest value provides the next best frame $C^\star$ of the hole, which will be converted to a motion of the peg.

the motion space between peg and hole. The idea behind our approach is to define a new task-specific weighting function over the sample set, which brings the peg closer to the (estimated) task goal at each iteration (Fig. 2).

We define T as the goal frame at the bottom of the hole with a known transformation $\boldsymbol{H}_{CT} = \text{const.} \in SE(3)$. The task is to achieve a match of frame D and T at the terminal time of the execution, i.e. $\boldsymbol{H}_{DT} = I_4$, which can be estimated by the expected value computed from the samples $\boldsymbol{H}_{D\hat{T}}$. The motion policy is derived locally in a neighborhood around the estimated hole frame $\hat{C}$:

$$B_m(\hat{C}) = \{\boldsymbol{x}_k{}^{(\mathrm{j})} | \operatorname{dist}(\boldsymbol{H}_{BC}{}^{(\mathrm{j})}, \boldsymbol{H}_{B\hat{C}}) < r_m\}. \qquad (1)$$

$\operatorname{dist}(\boldsymbol{H}_1, \boldsymbol{H}_2)$ denotes a distance metric between two transformations $\boldsymbol{H}_1, \boldsymbol{H}_2 \in SE(3)$. In Fig. 2, the neighborhood is visualized as a circle around $\hat{C}$. The choice of $r_m \in \mathbb{R}$ allows us to set the scope of the analysis and limit the possible robot motion per time step. In our current implementation, we only analyze if the samples are within an Euclidean distance of the position.

Then, for each sample j within $B_m(\hat{C})$ a task-specific weight $L_k{}^{(\mathrm{j})}$ is computed according to:

$$L_k{}^{(\mathrm{j})} = L(\boldsymbol{x}_k{}^{(\mathrm{j})}) = \operatorname{dist}(\boldsymbol{H}_{BT}{}^{(\mathrm{j})}, \boldsymbol{H}_{BD,k}), \qquad (2)$$

with the currently (measured) pose of the peg $\boldsymbol{H}_{BD,k} = \boldsymbol{H}_{BA,k}\boldsymbol{H}_{AD}$ and the pose of the goal frame attached to the hole evaluated for a single sample with $\boldsymbol{H}_{BT}{}^{(\mathrm{j})} = \boldsymbol{H}_{BC}{}^{(\mathrm{j})}\boldsymbol{H}_{CT}$. This weight describes how large the distance is between a hypothesis for the hole pose j and the currently measured pose of the peg. It is visualized as

a red line in Fig. 2. In our case, we compute the distance measure of (2) with:

$$\text{dist}() = d(\boldsymbol{H}_{BT}{}^{(\text{j})}, \boldsymbol{H}_{BD,k}) + f \cdot d_g(\boldsymbol{R}_{BT}{}^{(\text{j})}, \boldsymbol{R}_{BD,k}), \quad (3)$$

where $d()$ denotes the Euclidean distance of the translation between two frames, $d_g()$ is the geodesic distance on $SO(3)$ between orientations $\boldsymbol{R}_{BT}{}^{(\text{j})}$ and $\boldsymbol{R}_{BD,k}$, and $f \in \mathbb{R}$ is a constant scalar factor. At the end of the task-specific weighting step, we obtain the following set of samples:

$$P_L = \{(\boldsymbol{x}_k{}^{(\text{j})}, L_k{}^{(\text{j})}) \mid \boldsymbol{x}_k{}^{(\text{j})} \in B_m(\hat{C})\}. \quad (4)$$

Note that there might be alternative formulations for task-specific weights, e.g. potential energy with respect to the goal. Nevertheless, also in those formulations a balancing between errors in orientation and translation is typically necessary (at least implicitly).

B. Impedance-Controlled Guidance Strategy

Based on the samples P_L, we apply a policy to minimize the task-related weights. The policy selects the hypothetical best frame $C^\star$ to get closer to the task goal, which can be formulated as a minimization problem:

$$\boldsymbol{H}_{BC}{}^\star = \underset{\boldsymbol{H}_{BC} \in B_m}{\arg\min} (L_k{}^{(\text{i})}) \quad (5)$$

As depicted in Fig. 2 each sample in B_m is connected to a hypothesis of the goal $T^{(\text{j})}$. The weights $L_k{}^{(\text{j})}$ thus represent the estimated cost distribution, and the update step is a $(1, \lambda)$-ES algorithm [31].

As we assume that the pose of the real hole is fixed, we need to convert $C^\star$ to a reachable next best pose of the peg denoted as $D^\star$. This step is necessary, as the uncertainties were formulated with respect to the hole pose, but we only can command peg motions. The conversion is visualized in Fig. 3. First, the transformation from the best (hypothetical) hole frame $C^\star$ to the (measured) peg frame D can be formulated with the help of the (estimated) hole frame $\hat{C}$:

$$\boldsymbol{H}_{C^\star D} = \boldsymbol{H}_{C^\star \hat{C}} \boldsymbol{H}_{\hat{C} D} \quad (6)$$

In a similar way, we find the transformation from $\hat{C}$ to the best next peg frame $D^\star$ (to be determined) as:

$$\boldsymbol{H}_{\hat{C} D^\star} = \boldsymbol{H}_{\hat{C} D} \boldsymbol{H}_{D D^\star}. \quad (7)$$

Second, it is required that the (best) transformation between peg and hole in both formulations equals $\boldsymbol{H}_{\hat{C} D^\star} \equiv \boldsymbol{H}_{C^\star D}$. By setting (6)=(7) it follows:

$$\boldsymbol{H}_{D D^\star} = \boldsymbol{H}_{D \hat{C}} \boldsymbol{H}_{C^\star \hat{C}} \boldsymbol{H}_{\hat{C} D}. \quad (8)$$

The next best pose in the task space is then given as:

$$\boldsymbol{H}_{BD,k+1} = \boldsymbol{H}_{BD,k} \boldsymbol{H}_{D D^\star}. \quad (9)$$

Note that in the conversion of (8), an estimate of the current relative transformation $\boldsymbol{H}_{\hat{C} D}$ is required. The same would be necessary if the conversion were in the other direction from an uncertain peg to a known hole pose.

Finally, a linear motion to frame $D^\star$ is executed in the task space with a Cartesian impedance controller. We assume that the generated motions are reachable in joint space and that the robot is not in a singular configuration. The underlying low-level impedance controller guarantees the passivity of the robot in contact, and compensates pose errors that occur when the estimate is not yet accurate. Note that no adaption of the impedance is performed, and that the robot uses the same controller parameters throughout the entire insertion. A possible extension of the presented work is to adapt the controller parameters according to the current sample distribution.

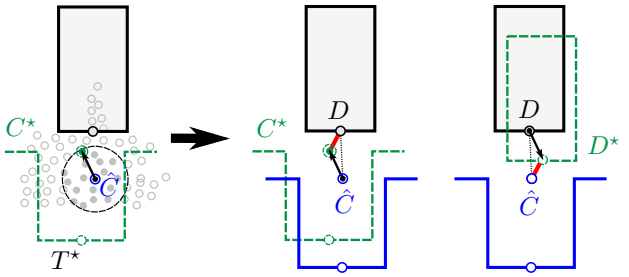


Fig. 3. Conversion of the hypothetical best frame $C^\star$ to a executable motion of the peg from D to $D^\star$. The best relative transformation $\boldsymbol{H}_{\hat{C} D^\star}$ needs to equal $\boldsymbol{H}_{C^\star D}$ (red lines).

TABLE I
PARAMETERS OF EXPERIMENTS

		Peg Round	Peg Profile	
Dimensions	Width	Ø 79	40 × 40	mm
	Depth	60	100	mm
	Clearance	1	0.1	mm
Uncertainty	Initial	Uniform	Gaussian	
	Position	40 × 40 × 10	$\sigma = 2$	mm
	Orientation	≈ 0.5	≈ 1	deg

V. RESULTS

A. Experimental Setup

The setup consists of a KUKA LBR iiwa robot with 7-DOF and joint torque sensors for the measurement of external forces and impedance control. We investigate the approach in two scenarios: a round peg-in-hole and the insertion of an aluminum profile into a fixture (Fig. 4). Geometric properties and initial uncertainties are summarized in Table I.

The algorithm is implemented in a partially paralleled manner, with 16 threads for collision checking, so that a command rate of up to $\approx$5 Hz can be realized. By default, we use a set of $N = 320$ samples. Further parameters of the observation algorithm are chosen according to [1]. In our current implementation, the KUKA RoboticsAPI is used to send the commands at the rate of our observation algorithm. The low level impedance controller of the robot runs at a higher controller rate >1 kHz. We use a standard Cartesian impedance controller with the peg frame D as a reference. An additional Lissajous overlay for oscillations in force and torque is added in the round peg experiment. This helped to make the insertion more robust when the peg got stuck at the upper rim of the hole. The values of r_m and f are currently not determined automatically, but tuned in the course of an initial test run.

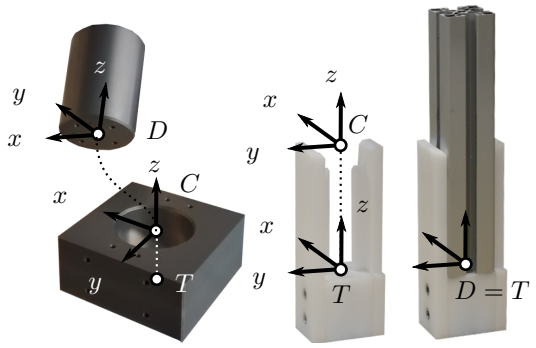
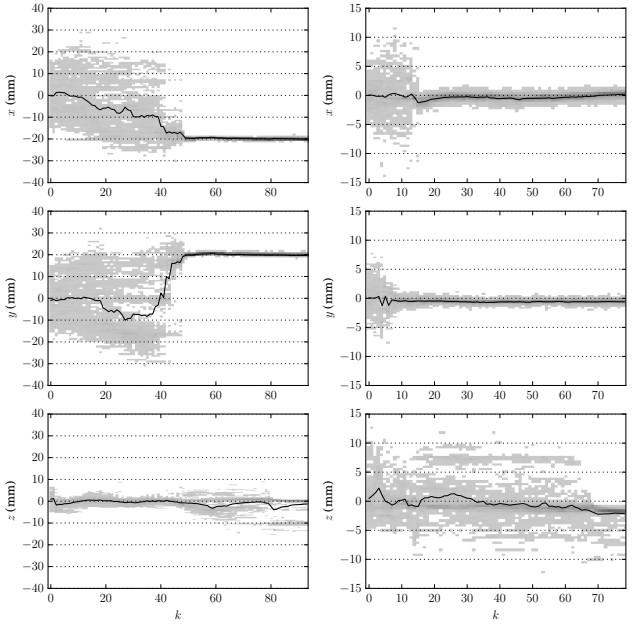


Fig. 4. The round peg (left) and aluminum profile with fixture (right) that are used for the experimental evaluation.

TABLE II
RESULTS OF EXPERIMENTS

		Peg Round	Peg Profile	
Success	# Runs	25	20	
	# Successful	21	16	
		84 %	80 %	
Speed	Execution Time			
	min./max.	15 / 59	69 / 258	s
	average	29.47	131.81	s
Estimation Accuracy	Position			
	min./max.	0.614 / 1.542	1.117 / 2.850	mm
	average	1.100	1.996	mm
	Orientation			
	min./max.	1.034 / 9.165	0.407 / 2.164	deg
	average	2.513	0.907	deg



(a) Peg Round (b) Peg Profile

Fig. 5. Sample evolution using the *bridge test policy* (N = 320) for the round peg (a) and the profile (b). The gray values encode the sample density at each time step k. The spread of the samples represents the uncertainty. The samples condense when the pegs enter the narrow passages of the configuration space at $k \approx 40$ and $k \approx 7$, respectively. It is visible in (b) that the distribution shrinks first in the y-, then in the x- and finally in the z-direction, when it touches constraint surfaces in corresponding directions of the hole. The black lines correspond to expected values.

B. Evaluation

In Fig. 5, an example for the sample evolution is shown in a histogram plot over time, with values given in dark gray corresponding to a high particle density. The initial distribution of the samples corresponds to uncertainties in the position of the hole. The uncertainties in the orientation are not plotted, as they are considered to be relatively small and do not change significantly over time. In Fig. 5a it can be seen that the distribution in x and y widens at the beginning of the process, where the force feedback does not provide information on the translation uncertainty, i.e. the peg is touching the upper plane of the hole. Once the estimator has feedback from the contact with the upper edge of the hole at $k \approx 40$, the samples condense and the peg follows a clear direction. At $k \approx 50$ the peg reaches the desired task goal at the bottom of the hole. In Fig. 5a the actual hole pose deviates from the initially assumed pose by $20\,\mathrm{mm}$ in x- and y-direction. At the end, the pose of the hole is estimated with an accuracy of $\approx 1\,\mathrm{mm}$. See Table II for a summary of the results. Also for the aluminum profile in Fig. 5b it can be seen that the distribution shrinks every time the configuration space narrows due to new constraint surfaces. At the beginning of the insertion, there is only a planar constraint in the y-direction; then the slot of the profile enters a passage with a geometrical guide that constrains in the x-direction before it touches the bottom.

The commands sent to the robot controller are plotted in Fig. 6 for the round peg, together with the estimate of the goal pose. At the beginning, the peg moves down from a height of $65\,\mathrm{mm}$, before contacting. Then the search for the entrance into the narrow passage of the configuration space starts implicitly. This is a challenging problem as the probability of placing samples into this passage is very small, but the *bridge test sampling* technique helps to keep the overall number of samples low. As soon as the algorithm manages to place enough samples in the narrow passage, directed motion commands guide the peg to the goal. Note that there is no controller switch involved in the transition from search to insertion. In Fig. 6 it is visible how the goal estimate pulls the peg to the goal. At the end, the current pose of the peg and the estimate of the goal pose converge in the same region at the actual hole bottom.

Fig. 7 provides the measured paths of the round peg for 25 runs with varying uncertainty. All runs start at the center of the assumed pose of the hole (with a small offset in z-direction which is not visible in the plot). The pose of the hole is varied in the xy-plane with a grid spacing of $10\,\mathrm{mm}$. 21 runs are successful, 4 runs fail mostly because the peg gets stuck in a stable peg-on-hole configuration (3-point contact). The experiment is canceled when the samples do not find a way to insert the peg in a reasonable amount of time. In Fig. 8, the translational distance to the goal is plotted for all runs with the round peg. The overall execution time is dependent on the size of the initial uncertainty, as indicated by the gray value of the lines. The run in which

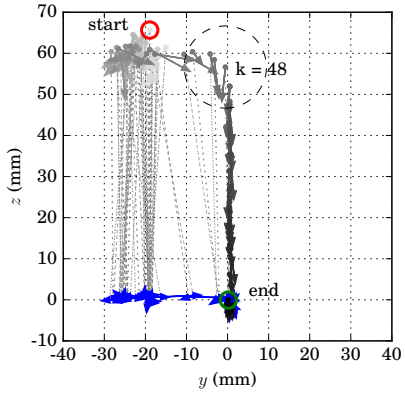


Fig. 6. Motion commands sent to the robot controller. Gray arrows represent the commanded motion of the peg (darker for increasing time). Each motion command is connected (light dashed lines) to an estimate of the goal (blue arrows indicate the update of the pose estimate). After a search phase at the beginning (start) the commands become more directed, as the force feedback provides a better estimate of the real pose of the goal. At the end, the estimate of the goal and peg converge in the same region.

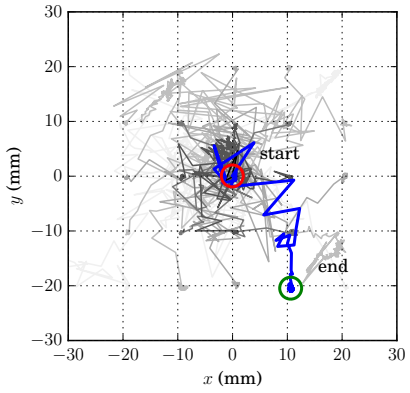


Fig. 7. Paths of the peg in the xy-plane for 25 runs. The pose of the hole is varied in 10 mm steps. The executions start above the assumed pose $\hat{C}$ (center) and end at the real pose of the goal T. One path is plotted in blue as an example. At the beginning, it appears to have the character of a search motion, but changes to a directed motion towards the end.

the uncertainty equals zero is the fastest and takes around 15 s. During the insertion phase, most runs show a decrease in distance. The execution is stopped when the estimated distance of D_k to the estimated goal $\hat{T}$ is below a threshold. For the round peg, the overall uncertainty reduction is significantly large; for the profile it is smaller and depends on the spatial direction. Note that the orientation error of the peg is comparably large as the rotation around the z-axis is unobservable for the estimator [32].

C. Discussion

In our previous work, we implemented the insertion of the profile manually [33] and achieved execution times of ≈ 6 s. Because of the tight tolerances between profile and fixture, and the depth of 100 mm, a well-chosen sequence of motion commands was necessary, which is not straightforward to program. The time for task completion in our current work is high compared to those existing approaches that use a manually programmed or predefined strategy for the complete insertion. Nevertheless, those approaches typically use more prior knowledge than in our approach or/and a parameter optimization phase either offline or online to speed up. Our approach can be combined with a second-stage parameter optimization approach after the first successful autonomous run. [21] provides a good overview of execution times for peg-in-hole, which lie in the range of 2 - 40 s. In our case, we estimate that second-stage optimization could accelerate the execution by a factor of one magnitude. Furthermore, the performance can be improved by even faster collision checks (currently in the worst case: 50 ms per sample) which limits the possible rate for motion generation to the greatest extent. It is worth noting that we have already achieved a successful insertion with a low command rate and fixed impedance controller parameters.

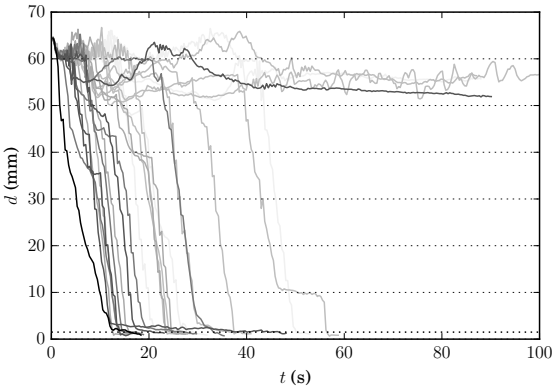


Fig. 8. (Estimated) distance between peg and goal for 25 runs over time. Dark lines indicate less initial uncertainty. The black line corresponds to zero uncertainty.

Further reasoning about the shape of the object and the configuration space could be integrated in order to increase the overall performance and make it more robust in specific contact situations, in the case of large uncertainties, and when faced with local minima. To date, we can not guarantee a successful execution for arbitrary geometries. Nevertheless, failures can be detected by a shape analysis of the sample distribution. Ideally, the estimated goal $\hat{T}$ converges towards the actual goal T, but whether this is possible or not depends also on the geometry of the object and the sensed contacts. In order to avoid local minima, a combination involving offline and experience-based planning methods should be investigated in future work.

VI. Conclusion

We provide a concept for combined sample-based observation and motion generation. It is shown how the algorithm successfully generates motion commands that guide a peg to a task goal specified relative to a hole. One benefit is the reduction of the prior knowledge required to execute an assembly task. A future extension of the framework is planned which maps the samples of the current configuration space to torque commands, and combines it with an online parameter optimization approach to speed up execution.

REFERENCES

[1] K. Nottensteiner, M. Sagardia, A. Stemmer, and C. Borst, "Narrow passage sampling in the observation of robotic assembly tasks," in *2016 IEEE Int. Conf. on Robotics and Automation (ICRA)*, 5 2016, pp. 130–137.

[2] H. Bruyninckx, S. Dutré, and J. De Schutter, "Peg-on-hole: A model based solution to peg and hole alignment," in *Proceedings of the 1995 IEEE Int. Conf. on Robotics and Automation*. IEEE, 1995, pp. 1919–1924.

[3] E. Staffetti, "Analysis of rigid body interactions for compliant motion tasks using the grassmann-cayley algebra," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 80–93, 2009.

[4] T. Tang, H. Lin, and M. Tomizuka, "Autonomous alignment of peg and hole by force/torque measurement for robotic assembly," in *2016 IEEE Int. Conf. on Automation Science and Engineering (CASE)*, Aug 2016, pp. 162–167.

[5] A. Stemmer, A. Albu-Schäffer, and G. Hirzinger, "An analytical method for the planning of robust assembly tasks of complex shaped planar parts," in *Proceedings of the 2007 IEEE Int. Conf. on Robotics and Automation*. IEEE, 2007, pp. 317–323.

[6] F. Wirnshofer, P. S. Schmitt, W. Feiten, G. v. Wichert, and W. Burgard, "Robust, compliant assembly via optimal belief space planning," in *2018 IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2018, pp. 1–5.

[7] M. Jokesch, J. Suchý, A. Winkler, A. Fross, and U. Thomas, "Generic algorithm for peg-in-hole assembly tasks for pin alignments with impedance controlled robots," in *Robot 2015: Second Iberian Robotics Conf.* Cham: Springer Int. Publishing, 2016, pp. 105–117.

[8] T. J. Debus, P. E. Dupont, and R. D. Howe, "Contact State Estimation Using Multiple Model Estimation and Hidden Markov Models," *The Int. Journal of Robotics Research*, vol. 23, no. 4-5, pp. 399–413, 2004.

[9] W. Meeussen, J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. De Schutter, "Contact-state segmentation using particle filters for programming by human demonstration in compliant-motion tasks," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 218–231, 2007.

[10] W. Meeussen, E. Staffetti, H. Bruyninckx, J. Xiao, and J. De Schutter, "Integration of planning and execution in force controlled compliant motion," *Robotics and Autonomous Systems*, vol. 56, no. 5, pp. 437–450, 2008.

[11] W. Meeussen, "Compliant robot motion: From path planning or human demonstration to force controlled task execution," Ph.D. dissertation, Katholieke Universiteit Leuven, 2006.

[12] D. Almeida, F. E. Viña, and Y. Karayiannidis, "Bimanual folding assembly: Switched control and contact point estimation," in *2016 IEEE-RAS 16th Int. Conf. on Humanoid Robots (Humanoids)*, Nov 2016, pp. 210–216.

[13] S. R. Chhatpar and M. S. Branicky, "Localization for robotic assemblies using probing and particle filtering," in *Proceedings of the IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*. IEEE, 2005, pp. 1379–1384.

[14] U. Thomas, S. Molkenstruck, R. Iser, and F. Wahl, "Multi sensor fusion in robot assembly using particle filters," in *Robotics and Automation, 2007 IEEE Int. Conf. on*, April 2007, pp. 3837–3843.

[15] Y. Taguchi, T. K. Marks, and H. Okuda, "Rao-blackwellized particle filtering for probing-based 6-dof localization in robotic assembly," in *Proceedings of the 2010 IEEE Int. Conf. on Robotics and Automation*. IEEE, 2010, pp. 2610–2617.

[16] Y. Taguchi, T. K. Marks, and J. R. Hershey, "Entropy-based motion selection for touch-based registration using rao-blackwellized particle filtering," in *Proceedings on the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systemss*. IEEE, 2011, pp. 4690–4697.

[17] S. Kim and M. Likhachev, "Parts assembly planning under uncertainty with simulation-aided physical reasoning," in *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2017, pp. 4074–4081.

[18] R. Andre, M. Jokesch, and U. Thomas, "Reliable robot assembly using haptic rendering models in combination with particle filters," in *2016 IEEE Int. Conf. on Automation Science and Engineering (CASE)*, Aug 2016, pp. 1134–1139.

[19] F. Wirnshofer, P. S. Schmitt, P. Meister, G. v. Wichert, and W. Burgard, "State estimation in contact-rich manipulation," in *2019 Int. Conf. on Robotics and Automation (ICRA)*, May 2019, pp. 3790–3796.

[20] D. Drieß, P. Englert, and M. Toussaint, "Constrained bayesian optimization of combined interaction force/task space controllers for manipulations," in *2017 IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2017, pp. 902–907.

[21] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," in *2019 Int. Conf. on Robotics and Automation (ICRA)*, May 2019, pp. 5844–5850.

[22] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana, "Deep reinforcement learning for high precision assembly tasks," in *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 819–825.

[23] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino, "Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects," in *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 2062–2069.

[24] T. Ren, Y. Dong, D. Wu, and K. Chen, "Learning-based variable compliance control for robotic assembly," *Journal of Mechanisms and Robotics*, vol. 10, no. 6, 2018.

[25] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *2019 Int. Conf. on Robotics and Automation (ICRA)*, May 2019, pp. 8943–8950.

[26] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots." *The Int. Journal of Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.

[27] J. J. Craig, *Introduction to Robotics, Mechanics and Control*, 3rd ed. Pearson, 2009.

[28] T. Lozano-Perez, "Automatic planning of manipulator transfer movements," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 10, pp. 681–698, Oct 1981.

[29] O. Cappé, S. J. Godsill, and E. Moulines, "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo," *Proc. of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[30] M. Sagardia and T. Hulin, "Evaluation of a penalty and a constraint-based haptic rendering algorithm with different haptic interfaces and stiffness values," in *2017 IEEE Virtual Reality (VR)*, March 2017, pp. 64–73.

[31] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies – a comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, Mar 2002.

[32] K. Nottensteiner and K. Hertkorn, "Constraint-based sample propagation for improved state estimation in robotic assembly," in *2017 IEEE Int. Conf. on Robotics and Automation (ICRA*, 5 2017, pp. 549–556.

[33] K. Nottensteiner, T. Bodenmüller, M. Kaßecker, A. M. Roa, A. Stemmer, T. Stouraitis, D. Seidel, and U. Thomas, "A complete automated chain for flexible assembly using recognition, planning and sensor-based execution," in *Proceedings of ISR 2016: 47st Int. Symposium on Robotics*. VDE, 6 2016.