



Title:	Self-reconfiguration of a robotic workcell for the recycling of electronic waste
Acronym:	ReconCycle
Type of Action:	Research and Innovation Action
Grant Agreement No.:	871352
Starting Date:	01-01-2020
Ending Date:	31-07-2024



Deliverable Number:	D3.2
Deliverable Title:	Skill adaptation and learning
Type:	Report
Dissemination Level:	Public
Authors:	Mihael Simonič, Kübra Karacan, Boris Kuster, Bojan Nemeč, Hamid Sadeghian, Sami Haddadin, and Aleš Ude
Contributing Partners:	JSI, TUM

Estimated Date of Delivery to the EC: 30-09-2023
 Actual Date of Delivery to the EC: 24-11-2023

Contents

Executive summary	3
1 Introduction	4
2 Adaptation and learning of control parameters	5
3 Learning of disassembly skills by exploiting physical task constraints	6
4 Relocation of computationally intensive learning algorithms to the cloud	7
5 Examples of adaptive operations in disassembly of electronic devices	8
5.1 Unscrewing in disassembly of smoke detectors	8
5.2 Levering of PCBs from heat cost allocators	9
5.3 Removing batteries from PCBs by a rocking motion	10
5.4 Opening a lid by levering	10
References	12
A Copies of scientific publications	13
A.1 “Passivity-Based Skill Motion Learning in Stiffness-Adaptive Unified Force-Impedance Control”	13
A.2 “The Inherent Representation of Tactile Manipulation Using Unified Force-Impedance Control”	22
A.3 “Hierarchical learning of robotic contact policies”	31
A.4 “Robust Execution of Assembly Policies Using Pose Invariant Task Representation”	44
A.5 “Tactile Exploration Using Unified Force-Impedance Control”	53
A.6 “A Passivity-based Approach on Relocating High-Frequency Robot Controller to the Edge Cloud”	60
A.7 “Adaptive Robotic Levering for Recycling Tasks”	68

Executive summary

In this deliverable, we primarily present the results of Task 3.4 “Adaptation and learning of control parameters” and Task 3.6 “Learning of disassembly movement primitives by exploiting physical task constraints”.

We review our contributions to the field of skill adaptation and learning in the context of robot-aided recycling of electronic waste, list our scientific publications and summarize how these contributions address the challenges of disassembling electronic devices.

1 Introduction

Conventional methods, such as the “crush-and-separate” technique for the recycling of electronic waste, exhibit limitations, particularly when the treated devices contain hazardous components such as batteries. Due to the fire hazard posed by batteries, they must be removed before further recycling steps can be initiated. However, removing batteries can only be accomplished if the electronic devices in question can be disassembled. The automation of this disassembly process for a wide range of electronic devices poses a significant challenge due to the diversity of electronic device and their varying physical condition when they are discarded. Consequently, devising a comprehensive automated solution for the battery removal has proven to be an exceptionally challenging task. As a result, current automation efforts have been directed toward specific device models to mitigate the complexity of this challenge. However, even within more focused applications, the array of different device models of a certain type remains substantial. Hence, there is need for efficient and adaptable solutions to improve the automation of the disassembly process.

Within the scope of the ReconCycle project, this issue is addressed through a tripartite process. Initially, we design an archetypical disassembly solution for one given exemplar of a device and specify the necessary steps/actions (such as levering, cutting, unscrewing, etc.) by employing modular and reconfigurable hardware and software architecture (see **Objective 1** in the Description of Action (DoA)). In parallel, the sensory information is aligned with the execution steps to create a semantic representation with variables that capture the action-relevant information (see **Objective 2** in DOA). This is necessary for the robot to autonomously determine which actions must be taken to disassemble. Lastly and of main interest in this document, the robot actions are subject to adaptation and need to be learned for each specific device model and the desired disassembly sequence (see **Objective 3** in DoA).

In previous reports, we outlined how the modularity and reconfigurability of the developed workcell enable us to alter the layout of the workcell quickly and efficiently (see deliverables **D1.1** and **D1.2**) Combined with re-configurable, flexible, and adaptable soft end-effectors (see deliverables **D4.1**, **D4.2** and **D4.3**), this helps to accommodate the handling of different device types in the same work cell. The reconfigurable hardware is accompanied by a modular and hierarchical software architecture, which can be roughly split into three levels: task-level programming (sequence of robotic skills), programming and acquisition of the robotic skills (e.g., levering, unscrewing, pushing, pulling), and the low-level control including skill adaptation. To achieve the necessary level of flexibility to accommodate the variability of devices within the same device family, we additionally rely on vision-based scene analysis and action prediction (see deliverables **D2.1** and **D2.2**), as well as skill learning adaptation of the control parameters (described in this report).

These capabilities have already been demonstrated in the use-case-related reports; in deliverable **D5.2**, we outlined an archetypical solution for the disassembly of a specific heat cost allocator (HCA) device that served as a basis for a more generalized pipeline for the disassembly of different models of HCAs, presented in **D5.4**. The proposed pipeline relies on workcell reconfiguration, action prediction, and skill adaptation. In deliverable **D5.5**, we demonstrated this process for another device type – smoke detectors.

In this report, we summarize our findings related to the aspects of skill adaptation and learning of robot control policies, providing an overview of the methodologies employed to address the challenges of disassembly of electronic devices. The report is organized as follows.

Section 2 discusses the methodologies for adapting and learning the control parameters for disassembly tasks. Section 3 focuses on how physical task constraints are utilized to learn disassembly movement primitives. Due to the computational intensity of certain learning and optimization algorithms, a discussion on cloud-based computational offloading is included in Section 4. Finally, in Section 5, we provide examples of how specific adaptive disassembly techniques are used to overcome use-case-specific challenges.

2 Adaptation and learning of control parameters

To disassemble various device models with a generic (high-level) disassembly protocol, an automatic adaptation of the control parameters of different (low-level) actions is required to successfully disassemble each specific device model. Each model comes with its own set of physical constraints and handling requirements, necessitating a system capable of integrating higher-level abstract knowledge, specific descriptions of task restrictions, and performance evaluation metrics.

Human experts in manufacturing have comprehensive insights into the task requirements [7]. This knowledge is object-specific and commonly comprises task constraints, such as the motion patterns, and the forces exerted on a specific object’s parts during manipulation. Two key elements are indispensable for effective transition from these requirements to their practical implementation: (1) a sophisticated adaptive control framework and (2) a task description formalism that enables programmatic translation of specific forces and motion profiles.

Our approach begins with predefined skills (such as levering or unscrewing) from the skill library. Whereas the skills are usually defined in the robot’s tool frame, the task requirements are usually defined in the object’s coordinate frame (e.g., apply force in x -direction to push the object). The control framework should, therefore, be able to determine a robot configuration to meet these requirements. Specific forces and torque profiles can also be met by modifying parameters such as the controller’s stiffness and additional force. To this end, we propose a **passivity-based skill motion learning scheme that utilizes a stiffness-adaptive unified force-impedance control framework** [5]. This framework extends adaptive impedance control with meta-parameter learning and compatible skill specifications.

Successfully integrating adaptive robot manipulation skills represents a highly complex problem that consists of desired force and form closures between the robotic end-effector and the objects to be manipulated. To address this complexity, we derive a **novel representation of tactile skills** to use within the unified force-impedance control framework, known as the *task phase plot* [4]. This formalism describes the entire cycle of a manipulation skill based on force and velocity information. Using unified force-impedance control, we develop a tactile manipulation strategy for robust contact initiation and flexible manipulation, even when environmental information is inaccurate. The manipulation approach dynamically employs impedance shaping to react to unforeseen contacts and force-shaping to initiate and shape desired contact conditions. The efficacy of this approach was empirically demonstrated in a peg-in-hole fitting experiment. The experiment yielded task phase plots that were evaluated against state-of-the-art impedance and force controllers regarding position accuracy, motion profile error, force tolerance, force profile error, and compliance.

The framework was presented at IROS 2022 [5]. The post-print version of the manuscript is provided in the Appendix. The representation of tactile skills will be presented at the upcoming CDC 2023 [4].

3 Learning of disassembly skills by exploiting physical task constraints

In practical applications, the pre-existing skill library may not contain all skills necessary for addressing specific use cases. In such cases, the robot should ideally be capable of learning new plans and skills autonomously. Given the inherent complexity of disassembling electronic devices, an advanced approach to autonomous skill acquisition is essential.

Disassembly tasks are intrinsically constrained by physical factors such as the device components' shape, size, and interconnections. Contact tasks are generally considered hard to learn, as the robot must learn a policy composed of poses and wrenches while interacting with an unknown and possibly changing environment. On the other hand, interacting with the environment can be advantageous to accelerate the learning process. Namely, learning physically constrained tasks is easier than learning tasks where a robot can move completely freely in space. The reason is that the environment constrains the admissible movement directions. Consequently, the number of parameters that need to be learned can be greatly reduced. To implement this type of learning, we need to use the natural robot motion along with the environmental constraints. Compliant robot control provides a suitable framework for implementing such a strategy.

Building upon these insights, we propose a novel **approach to learning contact-rich tasks based on a hierarchical learning scheme** [9]. One of the main advantages of our approach is its high learning speed, which exceeds the performance of reinforcement and deep reinforcement learning methods that do not exploit the constraints of contact tasks. A high learning speed is achieved by a meaningful decomposition of the task into decision processes at different abstraction levels and by exploiting environmental constraints while controlling the robot.

In addition, the framework proposes a powerful graph-based skill representation (example in Fig. 1), which can be used to represent the disassembly plan/sequence of disassembly primitives. The upper-level graph is essentially a precedence graph representing possible sequences of disassembly operations, whereas the lower-level graph represents the disassembly actions for the individual parts that need to be taken apart.

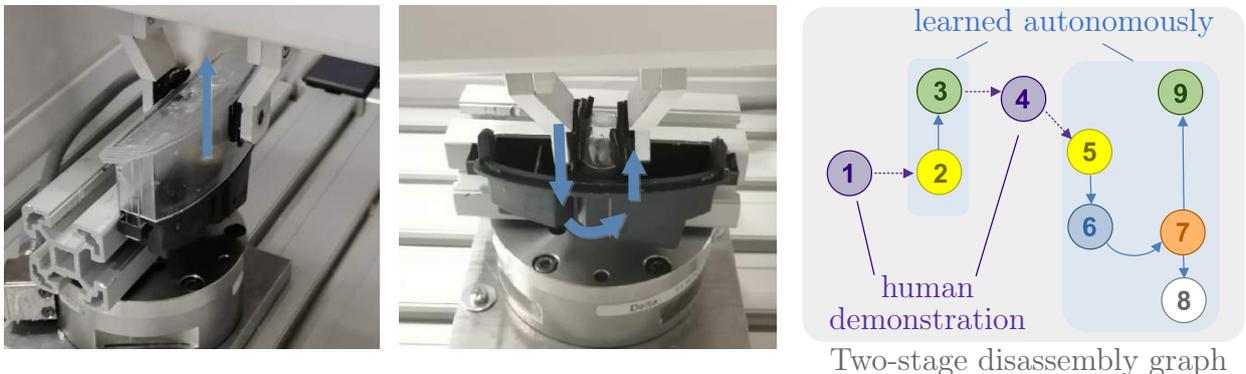


Figure 1: Robotic disassembly of a car number plate light. The robot first learns to remove the plastic cover and then continues by learning how to remove the bulb. A human demonstrates only where to grasp each part to start/proceed with the disassembly. The disassembly graph is shown on the right.

The learning task is to find optimal transitions from the start state to the goal state. Hierarchical reinforcement learning can be applied for this purpose. The upper RL level learns possible state sequences, while the lower level provides optimal sequences of movement primitives for the transition between the states. Both optimal transitions and optimal movement primitives can be learned simultaneously.

The framework has been experimentally verified on various tasks, including the disassembly of a car number plate light (shown in Fig. 1). A description and a video demonstration of this experiment are included in **D5.4**. The framework has been formally described and thoroughly evaluated in a paper published in Robotics and Computer-Integrated Manufacturing journal [9]. The paper is attached to this report in the Appendix.

Note that the disassembly graph can also be used to transfer an obtained skill to perform the reverse operation (assembly) or perform further disassemblies. To facilitate this, we developed an incremental and **pose invariant task representation that can be applied for more robust execution of contact tasks** presented at Ubiquitous Robotics 2023 [8]. The paper is included in the Appendix.

In addition to our framework for learning contact policies, we consider a complementary approach focusing on tactile skill adjustment in partially or fully unknown environments. This extension is rooted in the principles of tactile skill libraries, wherein a robot expert trains the robot with certain tactile action policies, including the motion and force profiles. Yet, when carrying out the desired task, the robot should be able to adapt those intended tactile action policies to an entirely or partially unknown environment based on estimations of environmental constraints. Even if the robot has an external sensing capability, such as a camera, it may still operate within tolerance, particularly under a fairly cluttered environment. Hence, the tactile skills should be developed to allow the robot to adjust the appropriate tactile skills to the surroundings with as little interference from outside as possible. Because the robot runs a tactile skill defined for certain conditions, changes in the geometry could be inferred from the local curvature [2]. Here, we use an exploration strategy to investigate the physical constraints of the environment, such as corners and edges, via local curvature observer. The exploration strategy utilizes the impedance controller's adaptive stiffness and robust contact via the force controller. Additionally, the recorded geometry information of the physical constraint might further be used to generate motion and force profiles for a new surface.

The detailed methodology, including the exploration strategy and the adaptive nature of the tactile skills, is comprehensively described in [2], which was presented at IFAC 2023. The paper is included in the Appendix.

4 Relocation of computationally intensive learning algorithms to the cloud

As robots become more and more intelligent, the complexity of the algorithms behind them is increasing. Specifically, most of the advanced (deep) learning algorithms based on vision require high computation power from the onboard robot controller. This increases the weight of the robot and energy consumption. A promising solution to tackle this issue is to **relocate the expensive computation to the cloud**. This part of the research investigates the possibility of relocating a state-of-the-art nonlinear control. This way, we can think about the recycling robot cell as a hardware station, with a centralized computation unit running all the software and

algorithms on the cloud centre. This is a promising solution for resolving intellectual properties and life-long learning algorithms that can be updated using data from all the stations. The results of this study have been presented at ICRA 2023 [1].

5 Examples of adaptive operations in disassembly of electronic devices

5.1 Unscrewing in disassembly of smoke detectors

As outlined in Section 2, any tactile process, such as levering or unscrewing, can be defined with specific boundary conditions in motion and force. Constraints restrict motion from a purely geometric standpoint, and the reaction force is zero along the free axis. In other words, during task execution, the tool should move along the free axis with a given velocity, while the contact forces occur along the other axes.

Ideally, the task phase plot demonstrates the entire power cycle that the object goes through, in which the force-velocity relation evolves such that the contact is established gradually. Continuity in the force-velocity task phase plot corresponds to the absence of abrupt power changes during the process, leading to success in the task. Therefore, we further developed the unified force-impedance control paradigm to command the object motion and force imposed by the task constraints [3]. We also developed the control shaping functions to maintain continuity in the task phase plot by stiffness variation and force adaptation. The undesired contacts cause deviations from the desired pose, creating either a pose error or external forces at the end-effector. This phenomenon can be exploited to react robustly to the undesired contacts and to re-configure the end-effector [5] by adapting the stiffness matrix.

The robot reacts to the environment if its behavior is compliant. Once the contact has been established, the robot recovers its maximum stiffness and resumes the desired motion from its present configuration. Additionally, we designed the force shaping function, which adapts the desired force needed to execute the task to account for errors in tool alignment and unintended contacts.

This approach can be employed for unscrewing tasks. While performing unscrewing using a readily available screwdriver (see Fig. 2), the controller automatically ensures continuous



(a) Contact established

(b) Maintaining contact

Figure 2: While performing the unscrewing using an off-the-shelf tool, the controller automatically maintains the contact.

contact with the target object. It allows the robot to handle potential tool alignment errors during the process. Furthermore, in situations where a significant alignment error occurs, leading to the loss of surface contact, the controller smoothly reduces the commanded force to zero. This transition enables the robot to seamlessly switch to an impedance-controlled mode, ensuring that it can continue following the desired motion without applying any force. This way we facilitate a safe and precise execution of the task.

5.2 Levering of PCBs from heat cost allocators

A common step in disassembly of electronic devices is levering, which allows the robot to apply greater forces when removing parts of the devices. In practical applications, the robot should be able to adapt a levering action to different device types without an operator specifically recording a trajectory for each device. A method to generalize the existing levering actions to new devices is thus needed. In [6], we presented a parameterized algorithm for performing robotic levering using feedback-based control to determine contact points and a sinusoidal pattern to realize adaptive levering motion.

Fig. 3 (a) shows a typical levering setup. The part to be levered (PCB) lies within the object (HCA casing). To increase mechanical advantage, the lever (fixed finger of the gripper) is positioned against the fulcrum (walls of the casing). We have observed that humans often use periodic movements when levering, especially when they do not know the force required to dislodge an object with the lever. In doing so, they slightly increase the force on the lever in each period. To mimic this behavior, we generate a single degree-of-freedom sinusoidal movement. In each repetition of the movement, we increase the amplitude. As the robot tries to move the lever even further, greater force is exerted on the PCB. The force is indirectly monitored based on the force-torque estimation performed internally by the robot control system using internal joint torque sensors. The levering is considered successful when a sudden drop is detected shortly after a peak in the estimated force, meaning that the locking mechanism has been broken and the PCB is levered out. We utilized the stiffness-adaptive unified force-impedance control framework described in Section 2 to control the robot during task execution.

The algorithm can deal with devices of different shapes, as shown in Fig. 3 (b) and (c). After the initial adaptation process, the subsequent executions of the learned levering action can be directly executed and accelerated to improve the performance.

Deliverable **D5.4** contains video demonstrations of the procedure for different HCAs. A paper presented at RAAD 2023 contains a more detailed description of the algorithm and performance evaluation [6].



Figure 3: Adaptive levering



Figure 4: Different types of battery mounting in the considered electronic devices.

5.3 Removing batteries from PCBs by a rocking motion

- cutting the contact wires,
- milling the contacts away,
- cutting the PCB into part with the battery and part without the battery, and
- loosening the contacts by a repetitive rocking motion.

Certain methods for battery removal from PCBs necessitate using specialized tools (such as a pneumatic cutter or CNC mill) or require dexterous manipulation (i.e., for holding the wires in place while cutting them). In addition, some approaches require precise sensing technology. Therefore, the procedure often needs to be manually adjusted for a new device and is less suitable for general-purpose disassembly pipelines. In contrast, human-inspired strategies like loosening the contacts by repetitive rocking motion offer greater adaptability.

In Fig. 5, we show how the battery can be detached from the rest of the smoke detector’s internals using the variable stiffness gripper (described in deliverable **D4.2**) with a levering tool by applying a rocking motion pattern.

We followed the task formalism described in Section 2, where the task is specified relative to the manipulated object. The battery needs to be pushed back and forth with gradually increasing force impulses (following the same underlying idea as with human-inspired levering described in the previous section). Likewise, the forces are monitored, and the motion is stopped when the object is freely moving (no opposing force).

5.4 Opening a lid by levering

The levering operation is one of the main steps in the disassembly pipeline. For instance, when removing the PCB from a heat-cost-allocator (HCA), levering lets us apply moments using the levering support at the edge of the HCA, as shown in Fig. 6. One approach to levering is to use periodic motions while maintaining contact perpendicular to the tooltip, essentially when the

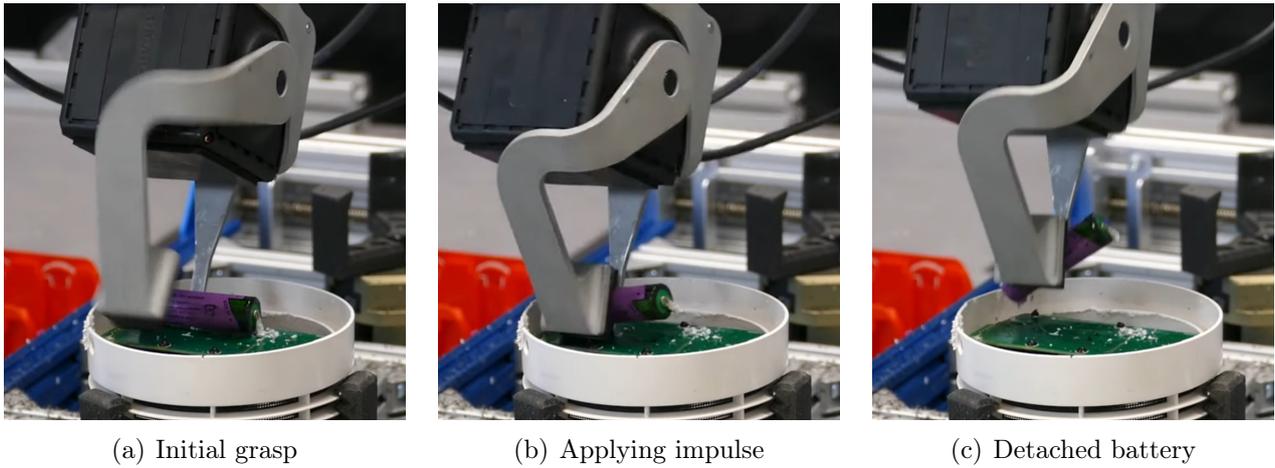


Figure 5: Battery removal by rocking motion.

desired force is complicated to define to lever an object [6]. Levering is likely successful when the locking mechanism is broken or fully opened. In other words, it is difficult to define a goal for a successful execution.

We designed an experimental setup to enable reproducible comparisons by choosing a car outlet socket as our exemplary object and manufacturing an aluminium counterpart to fix it firmly [3]. The lid of a car socket outlet is levered by using the peg. The experiment starts with no contact, and the algorithm is defined such that the robot should start with force control to establish contact. The expected behaviour is that if no contact is sensed, the robot should stop force control and restart when contact is sensed.

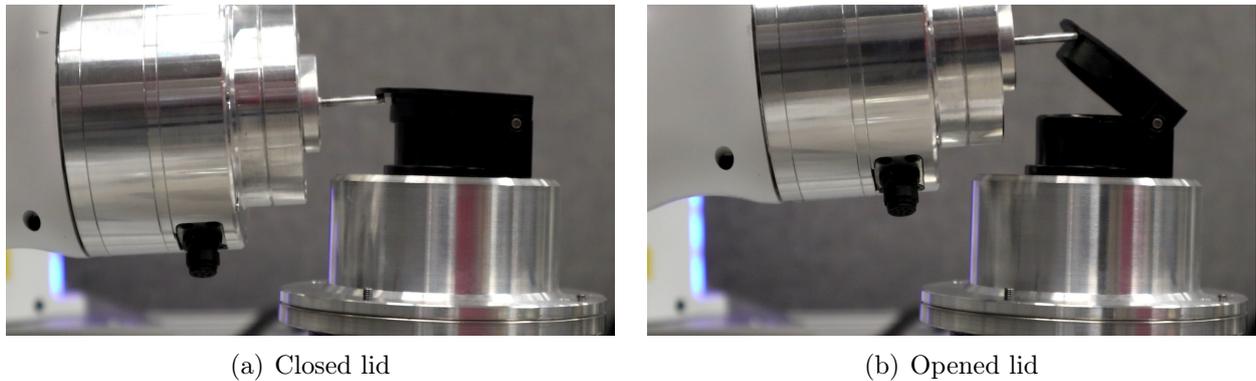


Figure 6: The levering operation can also be used to open lids. Some HCAs come with a plastic sleeve using the same locking mechanism.

References

- [1] X. Chen, H. Sadeghian, L. Chen, M. Tröbinger, A. Swirkir, A. Naceri, and S. Haddadin. “A Passivity-based Approach on Relocating High-Frequency Robot Controller to the Edge Cloud”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 5242–5248.
- [2] K. Karacan, D. Grover, H. Sadeghian, F. Wu, and S. Haddadin. “Tactile Exploration Using Unified Force-Impedance Control”. In: *IFAC Papers OnLine, 22nd IFAC World Congress* 56.2 (2023), pp. 5015–5020.
- [3] K. Karacan, R. Kirschner, H. Sadeghian, F. Wu, and S. Haddadin. “Tactile Robot Programming: Transferring Task Constraints into Constraint-Based Unified Force-Impedance Control”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Under review. 2024.
- [4] K. Karacan, R. J. Kirschner, H. Sadeghian, F. Wu, and S. Haddadin. “The Inherent Representation of Tactile Manipulation Using Unified Force-Impedance Control”. In: *IEEE 62nd Conference on Decision and Control (CDC)*. 2023.
- [5] K. Karacan, H. Sadeghian, R. Kirschner, and S. Haddadin. “Passivity-Based Skill Motion Learning in Stiffness-Adaptive Unified Force-Impedance Control”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 9604–9611.
- [6] B. Kuster, M. Simonič, and A. Ude. “Adaptive Robotic Levering for Recycling Tasks”. In: *Advances in Service and Industrial Robotics: Proceedings of the 32nd International Conference on Robotics in Alpe-Adria Danube Region (RAAD 2023)*. Springer, 2023, pp. 417–425.
- [7] A. Lambert and S. Gupta. *Disassembly Modeling for Assembly, Maintenance, Reuse and Recycling*. 1st ed. CRC Press, 2004.
- [8] B. Nemeč, M. M. Hrovat, M. Simonič, S. Shetty, S. Calinon, and A. Ude. “Robust Execution of Assembly Policies Using Pose Invariant Task Representation”. In: *20th International Conference on Ubiquitous Robots (UR)*. 2023, pp. 779–786.
- [9] M. Simonič, A. Ude, and B. Nemeč. “Hierarchical learning of robotic contact policies”. In: *Robotics and Computer-Integrated Manufacturing* 86 (2024), 102657, 12 pages.

A Copies of scientific publications

A.1 “Passivity-Based Skill Motion Learning in Stiffness-Adaptive Unified Force-Impedance Control”

This is the post-print version (author accepted manuscript) of the following publication: K. Karacan, H. Sadeghian, R. Kirschner, and S. Haddadin. “Passivity-Based Skill Motion Learning in Stiffness-Adaptive Unified Force-Impedance Control”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 9604–9611.

Passivity-Based Skill Motion Learning in Stiffness-Adaptive Unified Force-Impedance Control

Kübra Karacan, Hamid Sadeghian, Robin Kirschner and Sami Haddadin*

Abstract—Tactile robots shall be deployed for dynamic task execution in production lines with small batch sizes. Therefore, these robots should have the ability to respond to changing conditions and be easy to (re-)program. Operating under uncertain environments requires unifying subsystems such as robot motion and force policy into one framework, referred to as tactile skills. In this paper, we propose the enhancement of these skills for passivity-based skill motion learning in *stiffness-adaptive unified force-impedance control*. To achieve the increased level of adaptability, we represent all tactile skills by three basic primitives: contact initiation, manipulation, and contact termination. To ensure passivity and stability, we develop an energy-based approach for unified force-impedance control that allows humans to teach the robot motion through physical interaction during the execution of a tactile task. We incorporate our proposed framework into a tactile robot to experimentally validate the motion adaptation by interaction performance and stability of the control. While the polishing task is presented as our use case through the paper, the experiments can also be carried out with various tactile skills. Finally, the results show the novel controller’s stability and passivity to contact-loss and stiffness adaptation, leading to successful programming by interaction.

I. INTRODUCTION

With the demand for customized production, flexible production lines for small batch sizes are required [1]. These production lines ask for flexible adaptation to varying conditions [2]. Traditional position-controlled robots do not provide these capabilities. Compliant interaction for robots only become feasible, e.g., when using admittance control [3], impedance control [4], or force control [5]. Depending on the type of the robot and the controller, the tactile performance may vary significantly [6].

Highly tactile robots enable human-like intelligent recognition of touch and can perform sensitive tasks such as polishing, screwing, levering, or cutting [7]. Ideally, with those robots, the programmer only needs to command the robot what task to perform and where to perform it to (re-)program the robot task [8]. For example, when polishing cars, the human would tell the robot to polish in a certain spot and then quickly show it the next soiled spot while the robot is still in motion. Just as one human would show another where the polishing task is most required.

However, with the current state of the art in tactile robot programming, fairly accurate positioning is required before

The authors are with the Chair of Robotics and Systems Intelligence, MIRMI - Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, Germany * and also with the Centre for Tactile Internet with Human-in-the-Loop (CeTI). H. Sadeghian also has an affiliation with University of Isfahan, 8174673441 Isfahan.

Corresponding Author: K. Karacan kuebra.karacan@tum.de

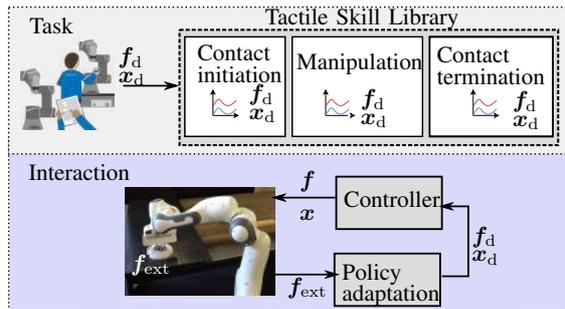


Fig. 1: **Adaptive tactile skills.** The tactile skill library provides the basic primitives to perform tactile robot tasks. In reality, however, actual situations differ from those defined in the library. Therefore, methods are required to extend the library and allow robots to adapt these well-defined skills to a new environment.

the tactile task can be successfully performed [9]. This requires expert knowledge and inefficient robot downtime to (re-)program the robot action [10]. Therefore, to enable flexible adaptation of tactile robot capabilities, new concepts are required to simplify the (re-)programming of robot tasks.

In this paper, a *tactile skill motion adaptation through unified force-impedance control* is proposed, as schematically depicted by Fig. 1. Three tactile skill primitives are defined, which are valid for any tactile robot-environment interaction; namely contact initiation, manipulation, and contact termination. To enable robot (re-)programming by interaction, we consider the manipulation primitive and design an adaptive unified force-impedance controller. This controller couples a stiffness-adaptive impedance control with force control to passively adapt the manipulation primitive’s periodic motion and direction of action. The result is a novel method applicable to any tactile robot skill, as it is illustrated in the accompanying video. For the sake of clarity in the paper we focus our explanation on the polishing task which is demonstrated using a Franka Emika (FE) robot arm.

The paper is structured as follows. In Sec. II, we summarize the state of the art. Section III introduces our proposed framework for stiffness-adaptive tactile skills. The experimental procedure as well as the results are presented in Sec. IV and Sec. V. Finally, Sec. VI concludes the paper.

II. STATE OF THE ART

Robotic tactile skills such as polishing, screwing, unscrewing, and levering require precise control of interaction at the end-effector. In other words, tactile skills should contain a motion generation unit with force policy. There are different perspectives in the literature for combining

motion generation and force control in robotic manipulators. For instance, Zielenski et al. [11] studied a method that classifies the manipulator behavior under three different phases: free-motion where the force is of no significance, exerting generalized forces, and lastly the transitions between the latter two behaviors. Robot manipulators need to develop complex perceptuomotor skills, as they further challenge real-world problems [12]. Therefore, defining tactile skills with a position control strategy alone is an impractical solution, and other control methods should be considered instead [4]. Furthermore, force control has been considered in many works, where the proposed controllers are validated based on constant force values or as thresholds/constraints [13]–[17]. However, the robots need to be robust enough to autonomously operate in unstructured environments and perform desired tasks in the presence of perception uncertainties [18].

Impedance control is a well-known approach that imposes a dynamic behavior between the external interaction and the desired motion, rather than tracking motion or force trajectory independently [4]. These dynamics can be realized in the joint space, operational space, or even in the redundant space of a robot manipulator [19]. The impedance inertia, damping, and stiffness parameters are usually selected constant in each direction based on the assigned tasks. The impedance dynamics represent a passive mapping between the external force input and the motion of the robot as an output. This ensures that the system does not generate additional energy when interacting with the passive environment and can therefore be considered stable [20]. However, adaptive change of the impedance parameters seems beneficial as well in many applications to impose a more wise and human-like behavior for the external interaction on the EEF [21]. Nevertheless, the variation of impedance parameters compromises the passivity of the control loop and therefore leads to instability. To this end, the concept of the energy tank is exploited to alter the dynamics of the closed-loop system and ensure passivity [22]. To include compliant force tracking on the environment with unknown geometry, the Wavelet Neural Network (WNN) has effectively been used in [23]. The variable admittance approaches can also be used to coordinate the human and robot motion in applications with shared autonomy [24]. However, still, with the state of the art in tactile robot programming, rather accurate positioning needs to be ensured before the task is performed.

In this work, we demonstrate that the required skill motion learning can be performed by adapting stiffness within unified force-impedance control, leveraging the energy transfer between robot and environment. We extend the unified force-impedance control paradigm by featuring control shaping function dynamics to allow humans to adapt any tactile skill by physical interaction.

III. METHODOLOGY

Our framework is based on adaptive force-impedance control that can be used for dynamic programming of tactile skill motions by interaction. To this end, we use stiffness

adaptation to enable programming by interaction and define tactile skill primitives for motion learning.

The taxonomy for tactile skills is first applied at the primitive level, and all the tactile skills are defined by three basic primitives. Unified force-impedance control is then extended to introduce the stiffness-adaptive dynamics to the control shaping function. This allows humans to adapt each tactile skill based on the energy transfer within the system. The proposed framework is illustrated in Fig 2 and explained in detail in the following.

A. Tactile primitives

To ensure compatibility with high-level planning and low-level control algorithms, any tactile skill is decomposed into three primitives. Each primitive \mathcal{P}_i , contributed to the desired skill, has its own policy $(\mathbf{x}_d, \mathbf{f}_d)$ and, therefore, force-motion generation unit:

- \mathcal{P}_1 – Contact initiation between tool and the work piece,
- \mathcal{P}_2 – Manipulation defined by a periodic motion and corresponding force policy,
- \mathcal{P}_3 – Contact termination to stop the interaction and decide if the desired skill is achieved.

The transition parameters between these primitives can be derived automatically from the perception of the scene through exteroceptive sensing and external user interaction, or manually through predefined scenarios. A blending strategy is then exploited for seamless integration of the primitives to achieve the desired skills.

In this framework, \mathcal{P}_2 is the core of any skill. It is defined by a periodic Dynamic Movement Primitive (DMP) as motion generator. The DMP is composed of the frequency and shape of the motion, and the required force policy is designed to be constant in the direction of motion. For adaptation of the motion, appropriate shaping functions are modelled to allow human interaction and safe loss-of-contact behavior, respectively, as described detailed in the following.

B. Stiffness-adaptive unified force-impedance control

The proposed control law for adaptive tactile skills is extended from the unified force-impedance controller introduced in [25] and comprises four main components:

- I) tracking the desired motion \mathbf{x}_d ,
- II) initializing and controlling the desired force \mathbf{f}_d ,
- III) deciding on the robot stiffness level ρ_{imp} , and
- IV) gravity compensation $\boldsymbol{\tau}_g$.

The robot dynamics equation in Cartesian space can be expressed as

$$\mathbf{M}_C(\mathbf{q})\ddot{\mathbf{x}} + \mathbf{C}_C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}} + \mathbf{g}_C(\mathbf{q}) = \mathbf{f}_{\text{in}} + \mathbf{f}_{\text{ext}}, \quad (1)$$

$$\boldsymbol{\tau}_{\text{in}} = \mathbf{J}^T(\mathbf{q})\mathbf{f}_{\text{in}}, \quad (2)$$

$$\boldsymbol{\tau}_g = \mathbf{J}^T(\mathbf{q})\mathbf{g}_C(\mathbf{q}), \quad (3)$$

where the robot mass matrix, Coriolis/centrifugal matrix, Jacobian matrix, and gravity vector in Cartesian space are denoted as $\mathbf{M}_C(\mathbf{q}), \mathbf{C}_C(\mathbf{q}, \dot{\mathbf{q}}), \mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times 6}$ and $\mathbf{g}_C(\mathbf{q})$ respectively. Furthermore, $\boldsymbol{\tau}_{\text{in}} \in \mathbb{R}^n$ is the control input

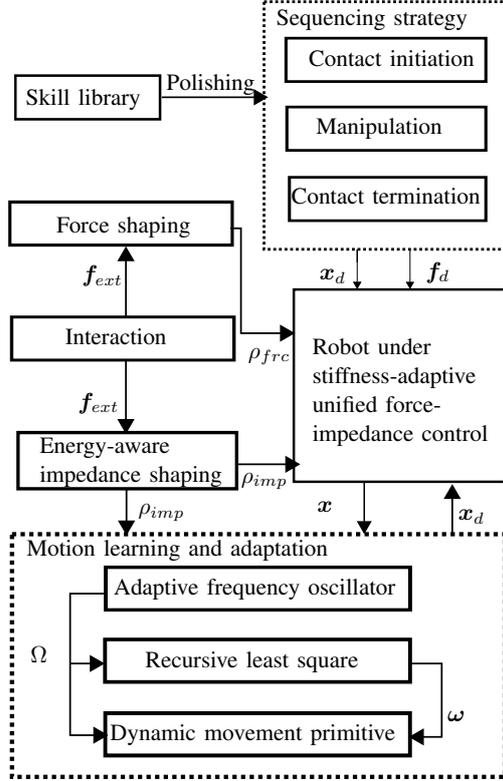


Fig. 2: **Block diagram for stiffness-adaptive tactile skill architecture.** Tactile skills are achieved by combining three basic primitives. The core of any skill is the manipulation primitive defined by a periodic motion and a desired force in the direction of action. The controller shaping functions passively regulate the interaction with the environment, such that ρ_{imp} allows motion adaptation by interaction, and ρ_{frc} is responsible for maintaining the contact force.

torque and $\mathbf{f}_{ext} \in \mathbb{R}^6$ is the external wrench acting on the robot.

The input torque, τ_{in} , is:

$$\tau_{in} = \tau_{imp} + \tau_{frc} + \tau_g, \quad (4)$$

where τ_{imp} , τ_{frc} , and $\tau_g \in \mathbb{R}^n$ are the input torques for the impedance, force, and gravity respectively.

1) *Force controller:* To initiate and control a desired contact force \mathbf{f}_d at the end-effector according to the external force, \mathbf{f}_{ext} , the force control is defined as

$$\tau_{frc} = \rho_{frc} \mathbf{J}^T(\mathbf{q}) \mathbf{f}_{frc}, \quad (5)$$

$$\mathbf{f}_{frc} = \mathbf{f}_d + k_p \tilde{\mathbf{f}}_{ext} + k_i \int \tilde{\mathbf{f}}_{ext} dt + k_d \dot{\tilde{\mathbf{f}}}_{ext}, \quad (6)$$

$$\tilde{\mathbf{f}}_{ext} = \mathbf{f}_d - \mathbf{f}_{ext}, \quad (7)$$

where $\mathbf{f}_{frc} \in \mathbb{R}^6$ is the feedback and feed forward force term, and k_p, k_i, k_d are PID gains. The force shaping function ρ_{frc} in (5) deactivates the force controller, when the robot deviates from the set-point of the end-effector. The idea behind the shaping function is to avoid undesired motions, especially when contact loss occurs. Therefore, ρ_{frc} is designed with

three cases based on the deviation from the set-point

$$\tilde{\mathbf{x}} = \mathbf{x}_d - \mathbf{x}. \quad (8)$$

The first case resembles the condition before contact where $\tilde{x}_z > 0$ and, thus, the energetic potential $\mathbf{f}_d^T \tilde{\mathbf{x}} \geq 0$. The second condition is $\tilde{x}_z < 0$ where the desired position is violated and thus the energetic potential turns $\mathbf{f}_d^T \tilde{\mathbf{x}} < 0$. To ensure a transition between both situations and prevent errors due to a mismatch of force and motion direction, a third case is included, which allows smooth reduction of the control torque within a certain position range up to a threshold $\delta_{frc} > 0$. This happens for example when the robot loses the contact with a surface during polishing and falls in z -direction. Finally, ρ_{frc} becomes

$$\rho_{frc} = \begin{cases} 1, & \tilde{x}_z \geq 0 \\ 0.5(1 + \cos(\pi(\frac{|\tilde{x}_z|}{\delta_{frc}}))), & 0 < |\tilde{x}_z| \leq \delta_{frc}, \\ 0 & else \end{cases} \quad (9)$$

2) *Energy-aware impedance controller:* To introduce an impedance behavior to the end-effector and allow human-based skill adaptation, the following control law is applied,

$$\tau_{imp} = \mathbf{J}^T(\mathbf{q})(\mathbf{K}_C \tilde{\mathbf{x}} + \mathbf{D}_C \dot{\tilde{\mathbf{x}}} + \mathbf{M}_C(\mathbf{q}) \ddot{\tilde{\mathbf{x}}} + \mathbf{C}_C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\tilde{\mathbf{x}}}). \quad (10)$$

The stiffness and damping matrices are denoted by \mathbf{K}_C and $\mathbf{D}_C \in \mathbb{R}^{6 \times 6}$, respectively. The energy storage in the closed loop system is operated by

$$S = \frac{1}{2} \dot{\tilde{\mathbf{x}}}^T \mathbf{M}_C(\mathbf{q}) \dot{\tilde{\mathbf{x}}} + \frac{1}{2} \tilde{\mathbf{x}}^T \mathbf{K}_C \tilde{\mathbf{x}}. \quad (11)$$

Considering (1), (5), and (10), the time derivative of (11) is

$$\dot{S} = \dot{\tilde{\mathbf{x}}}^T (-\mathbf{C}_C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\tilde{\mathbf{x}}} - \mathbf{D}_C \dot{\tilde{\mathbf{x}}} - \mathbf{K}_C \tilde{\mathbf{x}} - \mathbf{f}_{frc} - \mathbf{f}_{ext}) + \frac{1}{2} \dot{\tilde{\mathbf{x}}}^T \dot{\mathbf{M}}_C(\mathbf{q}) \dot{\tilde{\mathbf{x}}} + \dot{\tilde{\mathbf{x}}}^T \mathbf{K}_C \tilde{\mathbf{x}} + \frac{1}{2} \tilde{\mathbf{x}}^T \dot{\mathbf{K}}_C \tilde{\mathbf{x}}, \quad (12)$$

where the closed loop equation,

$$\mathbf{M}_C(\mathbf{q}) \ddot{\tilde{\mathbf{x}}} = -\mathbf{C}_C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\tilde{\mathbf{x}}} - \mathbf{D}_C \dot{\tilde{\mathbf{x}}} - \mathbf{K}_C \tilde{\mathbf{x}} - \mathbf{f}_{frc} - \mathbf{f}_{ext}. \quad (13)$$

has been used. Note that the skew-symmetry of the matrix $(\dot{\mathbf{M}}_C(\mathbf{q}) - 2\mathbf{C}_C(\mathbf{q}, \dot{\mathbf{q}}))$ is also applied. Considering (12) and the positive-definiteness of the damping matrix \mathbf{D}_C , it can be deduced that

$$\dot{S} \leq \frac{1}{2} \dot{\tilde{\mathbf{x}}}^T \dot{\mathbf{K}}_C \tilde{\mathbf{x}} - \dot{\tilde{\mathbf{x}}}^T \mathbf{f}_{frc} - \dot{\tilde{\mathbf{x}}}^T \mathbf{f}_{ext} + \dot{\tilde{\mathbf{x}}}^T \mathbf{f}_{ext}. \quad (14)$$

$$P_{acv,imp} = -\dot{\tilde{\mathbf{x}}}^T \mathbf{f}_{ext}, \quad (15)$$

$$P_{acv,K} = \frac{1}{2} \dot{\tilde{\mathbf{x}}}^T \dot{\mathbf{K}}_C \tilde{\mathbf{x}}, \quad (16)$$

$$P_{acv,frc} = -\dot{\tilde{\mathbf{x}}}^T \mathbf{f}_{frc}, \quad (17)$$

$$P_{ext} = \dot{\tilde{\mathbf{x}}}^T \mathbf{f}_{ext}, \quad (18)$$

shows the passivity with respect to the summation of the non-passive (i.e., active) control power P_{acv} as well as the external power P_{ext} .

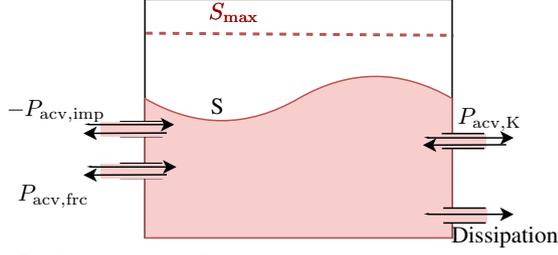


Fig. 3: **Energy transfer scheme between robot and the environment.** The energy is bounded by S_{\max} , such that the stability of the system during adaptation by interaction is preserved.

Here, we propose applying the derived energy models to bound the energy transferred from the robot to the human and to modulate the reaction of the robot to the human interruption (see Fig. 3). In other words, the stiffness of the robot changes with the shaping function, $\rho_{\text{imp}} \in [0, 1]$, where the stiffness adaptation rate, $\dot{\rho}_{\text{imp}}$, is formed based on the energy storage function of the robot, S , and the power inputs to the system $\langle -\dot{x}_d, \mathbf{f}_{\text{ext}} \rangle$, $\langle \dot{x}, \mathbf{K}_C \rangle$ by the force-motion policy generator. Inspired by [26], the stiffness adaptation is defined as

$$\mathbf{K}_C = \rho_{\text{imp}}(t) \mathbf{K}_{\max}, \quad (19)$$

and the adaptation parameter ρ_{imp} is obtained by

$$\dot{\rho}_{\text{imp}} = \begin{cases} \min\{\rho, 0\}, & \rho_{\text{imp}} = 1 \\ \rho, & 0 < \rho_{\text{imp}} < 1, \rho_{\text{imp}}(0) = 0, \\ \max\{\rho, 0\}, & \rho_{\text{imp}} = 0 \end{cases} \quad (20)$$

where, ρ is given by

$$\rho = h\rho_{\text{imp}} + \rho_{\text{min}}, \quad (21)$$

and,

$$h = \frac{S_{\max} - S}{S_{\max}}. \quad (22)$$

Note that, in order to have an initial increment for the case $\rho_{\text{imp}} = 0$, a small positive constant ρ_{min} has been introduced into the shaping function dynamics.

The above dynamic ensures that ρ_{imp} acts as a decreasing or an increasing exponential function, for $S > S_{\max}$ and $S < S_{\max}$, respectively. Overall, this leads to a stiffness behavior as follows.

$$\mathbf{K}_C = \begin{cases} \text{fully autonomous,} & \rho_{\text{imp}} = 1 \wedge S < S_{\max} \\ \text{fully loose,} & S > S_{\max} \\ \text{shared autonomy,} & \rho_{\text{imp}} = 0 \wedge S < S_{\max} \end{cases} \quad (23)$$

After S reaches the maximum allowable energy threshold S_{\max} , energy shaping starts¹.

When the robot operates without any human interruption, i.e., $\rho_{\text{imp}} = 1$, the stiffness equals its maximum value \mathbf{K}_{\max} . During human intervention, i.e., $P_{\text{acv,imp}} < 0$, the threshold

¹For now, S_{\max} is set heuristically by the user. Its autonomous adaptation is left for future studies.

is exceeded since the human inserts energy to the system. As the robot adapts its motion trajectory based on the human guidance, $\dot{\rho}_{\text{imp}}$ increases in (20), resulting in progressive energy dissipation.

C. Force-motion policy

As mentioned before, the primitive \mathcal{P}_2 is defined as the core of any tactile skill which is constituted by a periodic motion and a matching force policy. A periodic motion is comprised by its frequency Ω and shape ω . Moreover, the motion generator is coupled to the controller via a shaping function ρ_{imp} to allow autonomous adaptation and teaching of new skill motions by the human. However, for sake of clarity, we assume the desired force to be constant during contact, so we concentrate on the developed energy-awareness within our system on motion and stiffness adaptation level.

1) *Adaptive frequency oscillators*: In order to extract the frequency from a motion shown by the human operator, we employ the online learning and modulation method for periodic movements introduced in [27], which is shortly reviewed in the following. Generally, the core of the algorithm is formed by the dynamics [26]

$$\dot{\boldsymbol{\theta}} = (1 - \rho_{\text{imp}})(\boldsymbol{\phi} - a\mathbf{E}\sin(\boldsymbol{\theta})), \quad (24)$$

$$\dot{\boldsymbol{\phi}} = -(1 - \rho_{\text{imp}})(a\mathbf{E}\sin(\boldsymbol{\theta})), \quad (25)$$

$$\mathbf{E} = \text{diag}(\mathbf{x} - \hat{\mathbf{x}}), \quad (26)$$

where $a \in \mathbb{R}$ is the coupling constant and error matrix $\mathbf{E} \in \mathbb{R}^{m \times m}$ refers between the current position \mathbf{x} and predicted motion $\hat{\mathbf{x}}$, $(1 - \rho_{\text{imp}})$ introduces the progressive learning and adaptation behavior to the system, such that when the robot is fully autonomous $\rho_{\text{imp}} = 1$, the adaptation stops. The frequency vector $\boldsymbol{\phi}$ and the corresponding phase vector $\boldsymbol{\theta}$ belong to \mathbb{R}^m , where m is the number of task space coordinates to be adapted. The predicted motion is computed by

$$\hat{\mathbf{x}}_i = \sum_{c=0}^M (\alpha_{i,c} \cos(c\theta_i) + \beta_{i,c} \sin(c\theta_i)), \quad i = 1, 2, \dots, m, \quad (27)$$

where M denotes the number of Fourier components. The amplitudes $\alpha_{i,c}$, $\beta_{i,c}$ are updated as follows.

$$\begin{bmatrix} \dot{\alpha}_{i,c} \\ \dot{\beta}_{i,c} \end{bmatrix} = \begin{bmatrix} (1 - \rho_{\text{imp}}) \mu \cos(c\theta_i) e_i \\ (1 - \rho_{\text{imp}}) \mu \sin(c\theta_i) e_i \end{bmatrix}. \quad (28)$$

The error e_i is the i^{th} diagonal element of \mathbf{E} , and μ is the learning constant. Finally, the frequency Ω of the motion generator is set to the minimum element of $\boldsymbol{\phi}$ [28].

In this work, M , μ , and α are chosen to be 1, 1, and 50, respectively. The initial value for the phase vector is $\boldsymbol{\theta}_0 = 2\pi \mathbf{I}_{m \times 1}$.

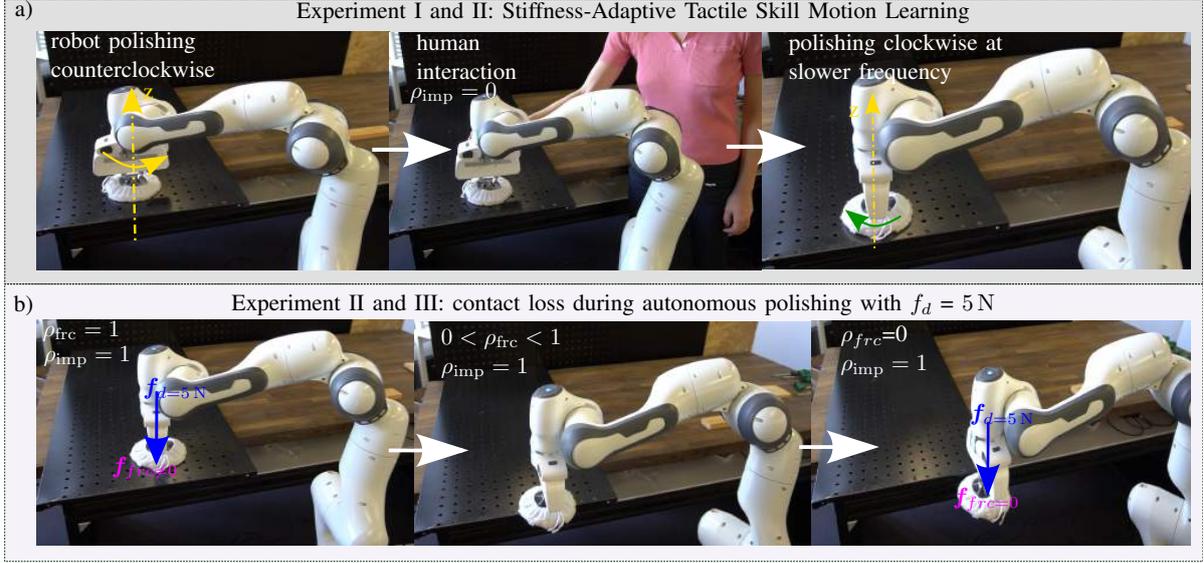


Fig. 4: **Tactile robot polishing a surface.** Image series for the experimental procedure a) Experiment I and II for passivity-based stiffness-adaptation by human interaction: human adapts the speed and direction of the free and contact motion b) Experiment III shows the performance of force and impedance shaping functions together: the robot autonomously and smoothly stops the force controller after contact loss.

2) *Motion generator:* DMPs have been designed to represent a smooth robot motion [29]. A periodic DMP controlling the motion of one degree of freedom $m = 1$, x_d is given by a second order differential equation system

$$\dot{s} = \Omega, \quad (29)$$

$$\dot{z} = \Omega(\alpha_z(\beta_z(x_g - x_d) - z) + \gamma(s)), \quad (30)$$

$$\dot{x} = \Omega z, \quad (31)$$

where x_g the attractor point and $s \in R$ is the phase variable which is used to make the system time independent. Furthermore, the constant parameters $\alpha_z, \beta_z > 0$ are selected such that the system has a unique attractor x_g . The nonlinear forcing term $\gamma(s)$ is defined as

$$\gamma(s) = \frac{\sum_{i=1}^N \omega_i \psi_i(s)}{\sum_{i=1}^N \psi_i(s)}, \quad (32)$$

$$\psi_i(s) = \exp(h(\cos(s - c_i))). \quad (33)$$

This allows us to freely design periodic trajectories from an initial position x_0 to the goal x_g by linear combination of N radial-basis functions $\psi_i(s)$. c_i is the center of the functions and h determines their width. In this work, we choose $N = 30$, $\alpha_z = 20$, $\beta_z = 5$, $h = 2.5$, and $c_i = \frac{2\pi}{N}(i - 1)$. The frequency Ω is chosen based on adaptive frequency oscillators. The generated trajectory with x_d, \dot{x}_d , and \ddot{x}_d is finally obtained by Euler integration of (31) with the initial values $s = 0$, $x = x_0$, and $\dot{x} = \Omega z = 0$. To extract the shape of the motion namely the weight matrix $\omega \in \mathbb{R}^{m \times N}$, each weight vector $\omega_i \in \mathbb{R}^m$ where $m = 3$ is

updated by the recursive least squares as follows [27]

$$\omega_i(t+1) = \omega_i(t) + \psi_i \text{diag}(\mathbf{P}_i(t+1)) e_{r_i}(t), \quad (34)$$

$$\mathbf{P}_{j,i}(t+1) = \frac{1}{\lambda} \left(\mathbf{P}_{j,i}(t) - \frac{\mathbf{P}_{j,i}(t)^2}{\frac{\lambda}{\psi_i} + \mathbf{P}_{j,i}(t)} \right), \quad (35)$$

$$e_{r_i}(t) = (1 - \rho_{\text{imp}})(\gamma_d(t) - \omega_i), \quad (36)$$

$$\mathbf{P}_{j,i}(0) = 1, \quad (37)$$

where γ_d the desired trajectory shape demonstrated by the human [29]. The inverse covariance vector $\mathbf{P}_i \in \mathbb{R}^m$ ($i = 1, 2, \dots, N; j = 1, 2, \dots, m$) corresponding to the weights ω_i is computed by using the forgetting factor $\lambda = 0.9995$.

When the impedance shaping function ρ_{imp} , reaches 1, the adaptation of the shape of the motion in (37) stops and is accepted as successfully approximated to the desired shape.

3) *Force policy:* the force policy is especially useful to mitigate the chattering/jamming behavior of the robot while trying to maintain the contact force and the pose error is high due to friction, etc. In this work, the force policy is chosen to be constant at the direction of action and coupled to the controller shaping function, ρ_{frc} .

IV. EXPERIMENTAL PROCEDURE

To demonstrate the real-life performance of our framework, the experiments are conducted using a Franka Emika robot for a polishing use case (see Fig. 4). The experiments were conducted considering three different cases:

- I) passivity-based learning by interaction during free motion in adaptive unified force-impedance control,
- II) adaptation by interaction during contact to validate the overall adaptive tactile skill definition,

III) performance of force and impedance shaping simultaneously.

To achieve robust and flexible behavior and to avoid long run times, the primitives are connected using a sequencing strategy [30], resulting in seamless and smooth integration of the primitives into the desired skill. Additionally, the skill definition enables straightforward and partly autonomous programming of complex skills. Algorithm 1 illustrates the procedure for an example scenario of polishing skill.

Algorithm 1 Skill definition for polishing

```

1: procedure POLISHING
2:   parameter server  $\leftarrow x_0, x_g, f_d, \omega, \Omega, \rho_{\text{frc}}, \rho_{\text{imp}}$ 
3:   contact initiation  $\mathcal{P}_1 \leftarrow x_g$ 
4:   manipulation  $\mathcal{P}_2 \leftarrow x_0, x_g, \Omega, \omega, f_d, \rho_{\text{frc}}, \rho_{\text{imp}}$ 
5:   contact termination  $\mathcal{P}_3 \leftarrow x_g$ 
6:  $\mathcal{P}_1$ :
7:   if  $x_g = x$  then
8:     go to  $\mathcal{P}_2$ 
9:   end if
10:  $\mathcal{P}_2$ :
11:   if  $\rho_{\text{frc}} < 1$  then
12:     go to  $\mathcal{P}_1$ 
13:   end if
14:   if  $\rho_{\text{imp}} < 1$  then
15:     adapt  $\Omega, \omega$ 
16:   end if
17:   if  $x = x_g$  then
18:     go to  $\mathcal{P}_3$ 
19:   end if
20:  $\mathcal{P}_3$ :
21:   if  $x = x_g$  then
22:     return success
23:   end if
24: end procedure

```

Any tactile skill such as polishing is defined by combining the above primitives (contact initiation, manipulation, and contact termination). Each of the primitives $\mathcal{P}_i(x_d, f_d)$ comprises a DMP and force policy. Furthermore, the DMP equation in (31) has the initial position x_0 and the goal x_g , which are varied between primitive skills and depends on the sequencing strategy. To solve the dependency problems, initial position x_0 and goal x_g are updated and combined according to Algorithm 1. The idea behind this operation is that when one primitive executes, this primitive policy is updated to generate the motion x_d and force trajectory f_d .

A. Experiment I

In the first experiment, the robot starts to perform a circular motion autonomously $\rho_{\text{imp}} = 1$ and a human operator interacts with the robot to vary the motion speed. Impedance shaping ρ_{imp} activates the adaptation of the motion based on the energy transfer between the robot and its environment, as bounded by the maximum allowable energy $S_{\text{max}} = 14$ J. The adaptation of the motion frequency Ω and x_d vs x

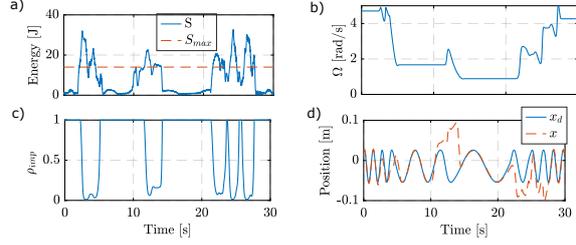


Fig. 5: Experiment I: Evolution of S , Ω and ρ_{imp} in free motion during adaptation of x_d a) and c) ρ_{imp} activates the adaptation of the motion based on the energy transfer between robot and the environment bounded by $S_{\text{max}} = 14$ J. b) The adaptation of Ω d) The adaptation of motion and comparison between the current state x_d and x based on the energy transfer.

are evaluated based on the energy transfer and impedance shaping.

B. Experiment II

In the second set of experiments, the robot polishes a surface and autonomously performs the entire stiffness-adaptive tactile skill definition, starting with the contact initiation primitive. Then, the manipulation primitive is activated to perform the polishing process by simultaneous force application and motion. The robot applies the desired force $f_d = 5$ N to the surface and executes the desired trajectory. During the primitive manipulation of the polishing process, the human operator interacts with the robot to adjust the speed and shape of the executed motion. To allow the human operator to interact with the robot during the contact skill, the maximum energy bound is set to $S_{\text{max}} = 2$ J.

C. Experiment III

In the last set of experiments, the performance of the force shaping function ρ_{frc} during the contact-loss is validated. The robot autonomously executes the entire framework, robustly stopping and restoring contact via ρ_{frc} . The applied force in the direction of action $f_{\text{frc},z}$ is expected to change uniformly from 5 N to 0 N after contact loss.

V. RESULTS AND DISCUSSION

The results for Experiment I in Fig. 5.a) and Fig. 5.c) show that the impedance shaping function ρ_{imp} is activated whenever the energy transfer between humans and robots is above the predefined maximum energy level ($S_{\text{max}} = 14$ J). More specifically, right after the interaction ρ_{imp} decreases to 0 from 1 such that the stiffness is also zero ($K_C = 0$), allowing humans to easily interact. Finally, the frequency of the motion is adapted during the interaction with respect to the guided motion. Additionally, the comparison between the desired trajectory x_d computed by the motion generator and the current state of the robot x is depicted in Fig. 5.b) and Fig. 5.d). Based on the results of the motion comparison, the human operator is able to slow down the robot during intervention and then speed it up again.

In Experiment II, it can be seen that in Fig. 6.a) and Fig. 6.b), where the S is greater than S_{max} , the human operator could change the shape and the frequency of the

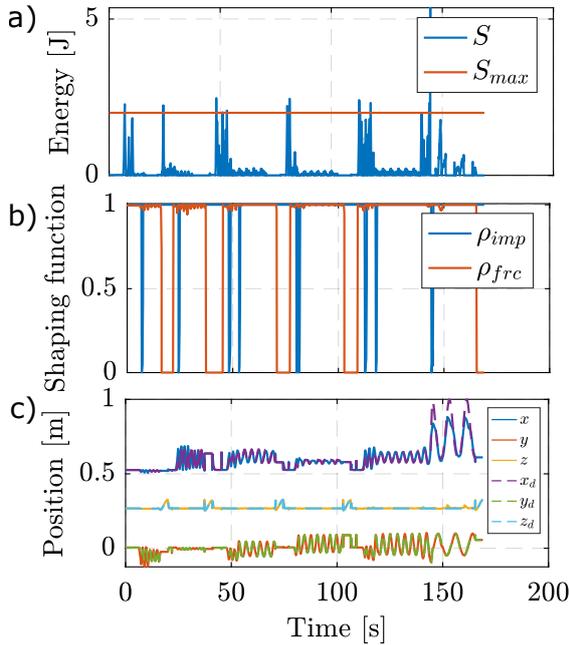


Fig. 6: **Experiment II: Evolution of S and ρ_{imp} , adaptation of x_d and y_d in contact with the environment.** S_{max} is set to 2 J to enable the human to interact with the robot that polishes by applying $f_a = 5$ N on the surface.

motion. As expected, after interacting with the human, the robot autonomously performs the rest of the adapted skill using the contact initiation and termination primitives.

Finally, in Experiment III, the robot robustly and safely stops the force controller by smoothly changing ρ_{frc} from one to zero as shown in Fig. 7.a). Consequently, the applied force to the surface, f_z , decrease to zero from 5 N as depicted in Fig. 7.b) and Fig. 7.d). Note that the robot is fully autonomous during the force regulation, as can be seen from the fact that $\rho_{imp} = 1$. Additionally, once contact with the surface is restored, ρ_{frc} is equal to 1 again and the manipulation primitive continues to apply force and executes the motion trajectory in parallel until another contact loss occurs.

VI. CONCLUSION

In this study, we developed a framework for skill motion learning in stiffness-adaptive unified force-impedance control based on energy transfer between robot and environment. The tactile capabilities are partitioned into three basic primitives, contact initiation, manipulation, and contact termination. The passivity-based dynamics of impedance shaping function is then introduced into unified force-impedance control, which results in motion adaptation capabilities by human interaction. We demonstrated the performance and stability of our system in real-world experiments, such as polishing a surface, unscrewing a knob, and levering a lid. In this paper, the polishing task is presented, while in the attached video the transition to other tasks is demonstrated

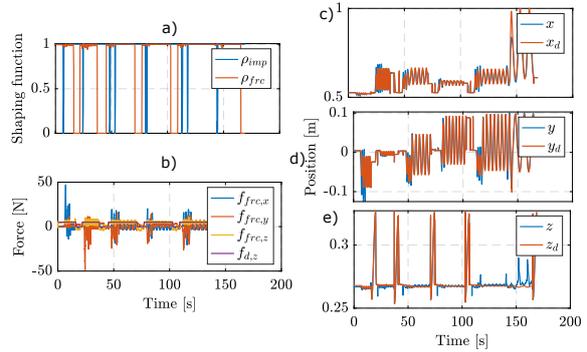


Fig. 7: **Experiment III: Evolution of ρ_{imp} and ρ_{frc} during polishing and adaptation of $f_{frc,z}$ and motion x, y, z .** a,b) The robot autonomously executes $\rho_{imp} = 1$, robustly stops and re-establishes contact via ρ_{frc} . $f_{frc,z}$ smoothly changes from 5 N to 0 N during contact loss, thereafter increases to 5 N with re-establishing contact c-e) set point for contact x_d, y_d, z_d vs. current position of the robot x, y, z during contact, contact loss and re-establishing contact

as well. In future work, we will investigate force learning through stiffness adaptation and the decision-making process for controlling the maximum allowable energy transfer between the robot and its environment during free and contact motions.

ACKNOWLEDGEMENT

We gratefully acknowledge the funding by the European Union's Horizon 2020 research and innovation program as part of the project ReconCycle under grant no. 871352 and project IAM under Grant agreement ID: 871899. This work was supported by LongLeif GaPa gGmbH (Geriatrics Project Y). The authors would like to thank the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for supporting the Lighthouse Initiative KI.FABRIK, (Phase 1: Infrastructure as well as the research and development program under, grant no. DIK0249), and the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy EXC 2050/1 Project ID 390696704 Cluster of Excellence Centre for Tactile Internet with Human-in-the-Loop (CeTI) of Technische Universität Dresden.

The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany (BMBF) in the programme of "Souverän. Digital. Vernetzt." Joint project 6G-life, project identification number 16KISK002.

Please note that S. Haddadin has a potential conflict of interest as a shareholder of Franka Emika GmbH

REFERENCES

- [1] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada, "The evolution of robotics research," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 90–103, 2007.
- [2] K.-D. Thoben, S. Wiesner, and T. Wuest, "'Industrie 4.0" and smart manufacturing – a review of research issues and application examples," *International Journal of Automation Technology*, vol. 11, pp. 4–19, 01 2017.

- [3] E. Shahriari, A. Kramberger, A. Gams, A. Ude, and S. Haddadin, "Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 136–142.
- [4] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American Control Conference*, 1984, pp. 304–313.
- [5] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robotics Autom.*, vol. 3, pp. 43–53, 1987.
- [6] R. J. Kirschner, A. Kurdas, K. Karacan, P. Junge, S. Birjandi, N. Mansfeld, S. Abdolshah, and S. Haddadin, "Towards a reference framework for tactile robot performance and safety benchmarking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4290–4297.
- [7] S. Haddadin, L. Johansmeier, and F. Díaz Ledezma, "Tactile robots as a central embodiment of the tactile internet," in *Proceedings of the IEEE*, vol. 107, no. 2, Feb. 2019, pp. 471–487.
- [8] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. V. Brussel, "Reconfigurable Manufacturing Systems," *CIRP Annals*, vol. 48, no. 2, pp. 527–540, 1999. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0007850607632326>
- [9] G. Hirzinger, A. O. Albu-Schäffer, M. Hähle, I. Schäfer, and N. Sporer, "On a new generation of torque controlled light-weight robots," *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 4, pp. 3356–3363 vol.4, 2001.
- [10] A. Kramberger, A. Gams, B. Nemeč, C. Schou, D. Chrysostomou, O. Madsen, and A. Ude, "Transfer of contact skills to new environmental conditions," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 668–675.
- [11] C. Zieliński and T. Winiarski, "Motion Generation in the MRROC++ Robot Programming Framework," *The International Journal of Robotics Research*, vol. 29, no. 4, pp. 386–413, 2010, eprint: <https://doi.org/10.1177/0278364909348761>. [Online]. Available: <https://doi.org/10.1177/0278364909348761>
- [12] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, "Skill learning and task outcome prediction for manipulation," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3828–3834.
- [13] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse, "Collaborative manufacturing with physical human–robot interaction," *Robotics and Computer-Integrated Manufacturing*, vol. 40, pp. 1–13, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0736584515301769>
- [14] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human–robot physical interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, Aug 2015.
- [15] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 620–629, March 2016.
- [16] F. Kulakov, G. V. Alferov, P. Efimova, S. Chernakova, and D. Shymanchuk, "Modeling and control of robot manipulators with the constraints at the moving objects," in *2015 International Conference "Stability and Control Processes" in Memory of V.I. Zubov (SCP)*, Oct 2015, pp. 102–105.
- [17] C. Ott, A. Dietrich, and A. Albu-Schäffer, "Prioritized multi-task compliance control of redundant manipulators," *Automatica*, vol. 53, pp. 416–423, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109815000163>
- [18] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, "Towards associative skill memories," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Nov 2012, pp. 309–315.
- [19] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 493–506, 2013.
- [20] C. Ott, *Cartesian impedance control of redundant and flexible-joint robots*. Springer, 2008.
- [21] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE transactions on robotics*, vol. 27, no. 5, pp. 918–930, 2011.
- [22] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4948–4953.
- [23] M. H. Hamedani, H. Sadeghian, M. Zekri, F. Sheikholeslam, and M. Keshmiri, "Intelligent Impedance Control using Wavelet Neural Network for dynamic contact force tracking in unknown varying environments," *Control Engineering Practice*, vol. 113, p. 104840, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0967066121001179>
- [24] F. Ferraguti, C. T. Landi, L. Sabattini, M. Bonfè, C. Fantuzzi, and C. Secchi, "A variable admittance control strategy for stable physical human-robot interaction," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 747–765, 2019. [Online]. Available: <https://doi.org/10.1177/0278364919840415>
- [25] C. Schindlbeck and S. Haddadin, "Unified Passivity-Based Cartesian Force / Impedance Control for Rigid and Flexible Joint Robots via Task-Energy Tanks," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 440–447, 2015.
- [26] F. Dimeas, T. Kastritsi, D. Papageorgiou, and Z. Doulgeri, "Progressive Automation of Periodic Movements," in *Human-Friendly Robotics 2019*, F. Ferraguti, V. Villani, L. Sabattini, and M. Bonfè, Eds. Cham: Springer International Publishing, 2020, pp. 58–72.
- [27] A. Gams, A. Ijspeert, S. Schaal, and J. Lenarcic, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Autonomous Robots*, vol. 27, pp. 3–23, 07 2009.
- [28] T. Petrič, A. Gams, A. J. Ijspeert, and L. Žlajpah, "On-line frequency adaptation and movement imitation for rhythmic robotic tasks," *The International Journal of Robotics Research*, vol. 30, no. 14, pp. 1775–1788, 2011. [Online]. Available: <https://doi.org/10.1177/0278364911421511>
- [29] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models formotor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [30] L. Johansmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 5844–5850, 2019.

A.2 “The Inherent Representation of Tactile Manipulation Using Unified Force-Impedance Control”

This is an accepted manuscript: K. Karacan, R. J. Kirschner, H. Sadeghian, F. Wu, and S. Haddadin. “The Inherent Representation of Tactile Manipulation Using Unified Force-Impedance Control”. In: *IEEE 62nd Conference on Decision and Control (CDC)*. 2023.

The Inherent Representation of Tactile Manipulation Using Unified Force-Impedance Control

Kübra Karacan, Robin Jeanne Kirschner, Hamid Sadeghian, Fan Wu, and Sami Haddadin

Abstract—Different robotic manipulation tasks require different execution and planning strategies. Nevertheless, the versatility of tasks in assembly and disassembly demands flexible control strategies. Fundamental to achieving such adaptive control methods is understanding and generalizing the interactions between tools, the manipulated object, and the environment required to perform a manipulation. This paper addresses the problem of generating adaptive manipulation by introducing the force-velocity task phase plot that represents the inherent nature of tactile manipulation skills. This representation enables us to identify the primary phases of the interaction in the force-velocity domain. Using unified force-impedance control, we establish a tactile manipulation strategy to robustly conduct versatile manipulation tasks even in case of disturbances or imprecise task information. The proposed control scheme features a dynamic process for impedance shaping based on the external force applied to the robot and the skill motion error for collision response, as well as a force-shaping function that enables both a smooth transition from free motion to contact and force regulation. We implement and compare the control strategy to previously proposed strategies using peg-in-hole reference experiments that include force disturbance and positioning inaccuracies and show the respective task phase plots. As a result, we observe high controller robustness and conclude that using the task phase plot as the inherent representation of tactile manipulation via unified force-impedance control enables successful adaptive controller design and creates a quantifiable basis for robotic skill solution comparison.

I. INTRODUCTION

Robotic manipulation plays a vital role in the digital transformation of traditional factories, enabling the automation of assembly and disassembly operations, e.g., for electronic waste recycling [1]. Robotic solutions to such processes require complex robot capabilities and robustness to varying conditions [2]–[4]. For example, the dismantling procedure for a battery from a heat cost allocator (see Fig. 1) includes

The authors are with the Chair of Robotics and Systems Intelligence, MIRMI-Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, Germany, and also with the Centre for Tactile Internet with Human-in-the-Loop (CeTI). H. Sadeghian also has an affiliation with the University of Isfahan. We gratefully acknowledge the funding by the European Union’s Horizon 2020 research and innovation program as part of the project ReconCycle under grant no. 871352, the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for the Lighthouse Initiative KI.FABRIK, (Phase 1: Infrastructure as well as the research and development program under, grant no. DIK0249), LongLeif GaPa gGmbH (Geriatronics Project Y) and German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany’s Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence “Centre for Tactile Internet with Human-in-the-Loop” (CeTI) of Technische Universität Dresden. The authors would like to thank the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for financial support as part of the project SafeRoBAY (grant number: DIK0203/01). Please note that S. Haddadin has a potential conflict of interest as a shareholder of Franka Emika GmbH.

Corresponding Author: K. Karacan kuebra.karacan@tum.de

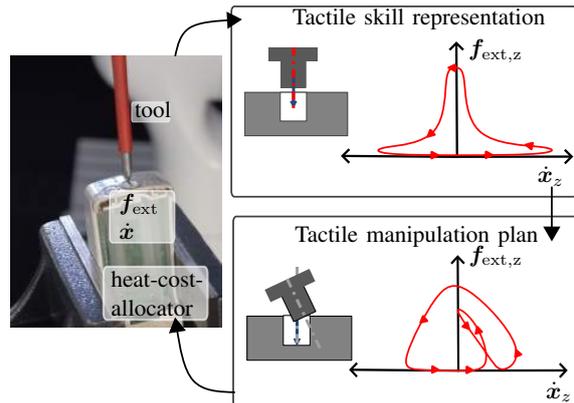


Fig. 1: Pipeline from real-life application to tactile skill. Understanding the underlying system of a tactile skill by simplified representations enables the derivation of manipulation plans that can cope with environmental uncertainties for successful task execution.

(i) placing the tool in contact with the gap (pre-contact and contact initiation); (ii) pushing the pin; (iii) levering the lid and PCB; and (iv) separating the battery [5]. To obtain a flexible automation solution to this versatile process, the location, and dimension of the region of interest, like the screw hole, can be obtained, e.g., by a camera. Even though such external sensing suffers from uncertainties and may provide imprecise positioning information, the robot should robustly align the tool with the contact to execute the desired tactile manipulation skills, such as pushing the pin, levering the lid, and cutting the battery. Thus, besides precise motion, a sense of touch has become crucial, which leads to the definition of tactile manipulation skills [6]–[8].

The introduction of torque-controlled tactile robots capable of perceiving touch enabled robotic skills that require force-motion commands and high compliance [9], [10]. To allow these skills, multiple strategies are available, e.g., admittance control [11], impedance control [12], force control [13], and even unified control [14]. Numerous studies consider force control as a solution to adaptive robotic skills, validating the suggested controllers using constant force values, thresholds, or limitations [15]–[19]. However, these strategies cannot cope with environmental uncertainties and may fail when faced with perception imprecision [20], [21]. Impedance control is a well-known technique that enforces dynamic behavior for interacting with the environment and the desired motion [12]. These dynamics can be achieved in the joint space, operational space, or even the null space of a robot manipulator [22]. Adaptive adjustment of impedance

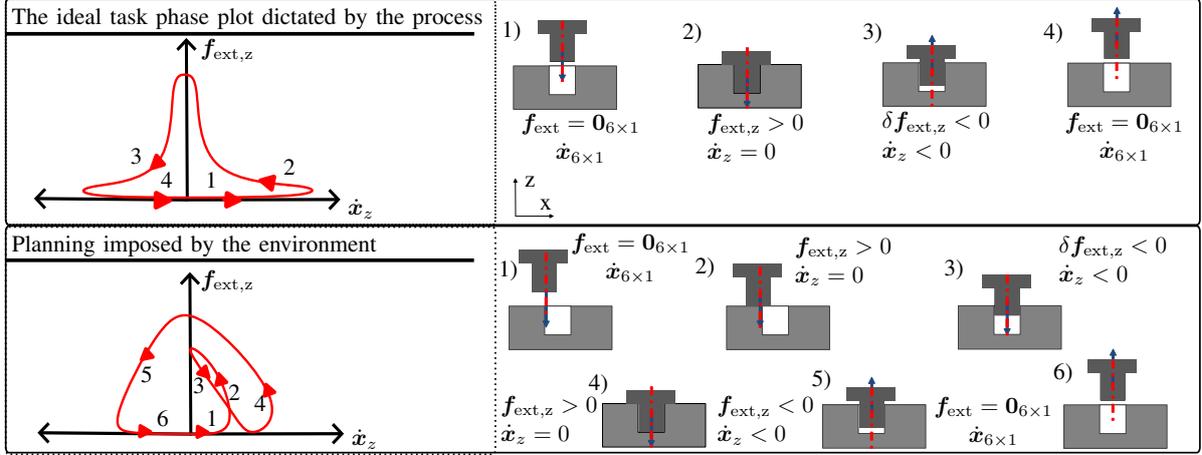


Fig. 2: **Tactile skill representation.** The task phase plot (f_{ext}, \dot{x}) is constrained by the desired state of the object, while the current environmental conditions form its shape. A manipulation plan should be devised to enable smooth transitions between the phases.

parameters is helpful in many applications [23], [24]. In applications with shared autonomy, variable impedance techniques are also utilized to coordinate the motions of humans and robots and update the desired skill motion policy [6]. Although the robotic community has been investigating adaptive manipulation with perception uncertainties so far, this is yet to be solved in principle and has not found its way into the industrial application domain [25]–[28]. Consequently, the field of electronic waste recycling is still primarily dominated by manual labor.

Successfully integrating adaptive robot manipulation skills represents a highly complex problem that consists of desired force and form closures between the robotic end-effector and the objects to be manipulated. Despite multiple attempts to solve adaptive manipulation, to the best of the authors’ knowledge, a methodological approach to skill development starting with a formalism to represent the desired tactile manipulation strategy is yet missing. In this study, we derive a first representation of tactile skills based on unified force-impedance control, namely the *task phase plot*. It describes the entire cycle of a manipulation skill based on force and velocity information. Using unified force-impedance control, we use this formalism to develop a tactile manipulation strategy for robust contact initiation and flexible manipulation assuming inaccurate environmental information. The manipulation approach is designed dynamically, using impedance shaping to react to unforeseen contact and force-shaping to initiate and shape desired contact conditions. Using a peg-in-hole fitting experiment, we demonstrate the manipulation method and derive the task phase plots for comparison with state-of-the-art impedance and force controllers.

Additionally, we introduce the soft displacement metric that tests the robustness of the introduced manipulation strategy to position inaccuracies. We observe good positioning robustness and increased success of the introduced manipulation strategy compared to other solutions. Lastly, we present a possible extension of our controller for human-robot interaction

The paper is organized as follows. Section II introduces the inherent representation of tactile skills for robust contact initiation and flexible manipulation under environmental and positioning uncertainties, using unified force-impedance control. The experimental procedure and results are presented consecutively in Sec. III and Sec. IV. Finally, Sec. V concludes the paper.

II. METHODOLOGY

Physical interaction between two bodies requires robustly establishing and maintaining the desired contact by either force or form closure. Adapting to undesired contacts is one enabling factor in maintaining contact forces or achieving form fit. In this study, interaction skills that require motion and force policies and compliant behavior are referred to as *tactile manipulation skills*.

A. Tactile Skill Representation

Any tactile process such as peg-in-hole followed by releasing is defined with certain boundary conditions, i.e., motion and force. Ideally, the task phase plot, as shown in Fig. 2, demonstrates the entire cycle that the peg goes through, in which the force-velocity relation evolves:

- 1 - starting moving freely $f_{\text{ext}} = \mathbf{0}_{6 \times 1}$ toward the hole, while speeding up to $\dot{x}_z > 0$,
- 2 - smoothly establishing contact $\dot{x}_z = 0$ with the bottom surface, while an external force $f_{\text{ext},z} > 0$ is exerted to it,
- 3 - breaking contact $\delta f_{\text{ext},z} = 0$ while moving $\dot{x}_z < 0$,
- 4 - back to the free condition $f_{\text{ext},z} = 0$ and $\dot{x}_z > 0$.

However, the actual motion of the peg and the force exerted are formed based on force/wrench and twist constraints imposed by the environmental conditions, such as undesired contacts due to positioning imprecision. Here, the phases of tactile interaction repeat within the phase plot as contact is established multiple times before the desired goal state is achieved. Consequently, a tactile manipulation strategy is required to adapt the desired force and motion of the peg

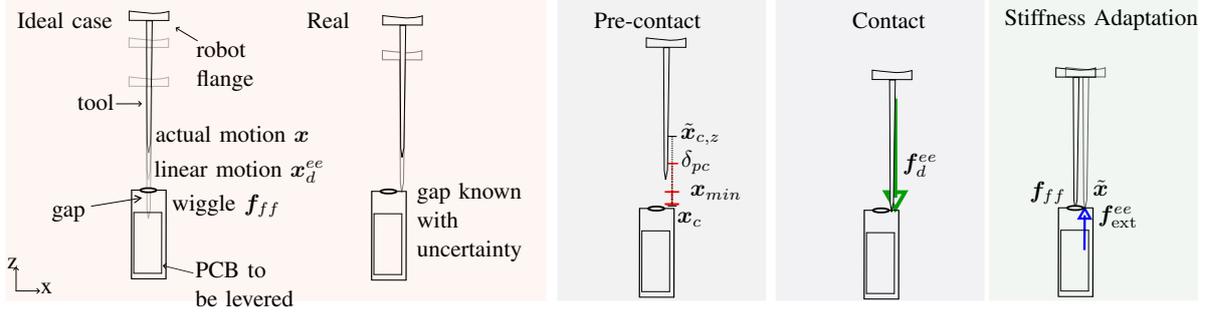


Fig. 3: A simplified scheme of elements in the dismantling process of a heat cost allocator to recycle the battery. The gap location x_c is assumed to be obtained externally. Impedance control is followed by desired motion x_d . To insert the tool, a wobble motion is applied via a feed-forward force f_{ff} . When force control is activated, a desired force f_d establishes contact with the surface to make tool alignment possible. Stiffness adaptation enables the robot to have compliant behavior to adjust itself to the environment.

between the phases, leading to robust and successful task execution.

B. Tactile Manipulation Strategy

Using force-impedance control, we develop a tactile manipulation plan for robust contact initiation and flexible manipulation under positioning inaccuracies, as shown in the example of a peg-in-hole task for heat cost allocator disassembly in Fig. 3. First, we design a dynamic process for impedance shaping based on the external force exerted on the robot and the skill motion error to allow the robot to adjust its end-effector in response to incidental contact. Second, we create a force shaping function to enable (i) a smooth transition from free motion to contact (pre-contact shaping) and (ii) force regulation based on the desired tool alignment (contact shaping).

1) **Control design:** To control robot arm motion and force policies with n -DOF, the desired pose of the end-effector w.r.t. the robot base frame is $x_d \in \mathbb{R}^6$. The robot's dynamics equation in Cartesian space in the base frame is

$$M_C(q)\ddot{x} + C_C(q, \dot{q})\dot{x} + f_g(q) = f_{in} + f_{ext} \quad (1)$$

where $f_{ext} \in \mathbb{R}^6$ is the external wrench w.r.t. the base frame. $M_C(q)$ is the robot mass matrix, $C_C(q, \dot{q}) \in \mathbb{R}^{6 \times 6}$ is the Coriolis and centrifugal matrix, and $f_g(q)$ is the gravity vector in Cartesian space. Furthermore, f_{in} is the wrench applied by the robot and relates to the joint control torque $\tau_{in} \in \mathbb{R}^n$ by $\tau_{in} = J^T(q)f_{in}$. Next, we design a control algorithm for the input torque τ_{in} to perform the desired task. The proposed control law for adaptive tactile skills shown in Fig. 4 is extended from unified force-impedance control [6], [14] and comprises four main components:

- I) tracking the desired motion x_d and adapting the robot stiffness level (ρ_{imp}) with impedance control,
- II) applying feed-forward force f_{ff} ,
- III) regulating the external force f_{ext} w.r.t. the desired force f_d ,
- IV) gravity compensation for the robot.

The input torque $\tau_{in} \in \mathbb{R}^n$ is:

$$\tau_{in} = \tau_{imp} + \tau_{ff} + \tau_{frc} + \tau_g, \quad (2)$$

where τ_{imp} , τ_{ff} , τ_{frc} , and $\tau_g \in \mathbb{R}^n$ are the input torques for (i) impedance control, (ii) feed-forward torque, (iii) force control, and (iv) gravity compensation.

2) **Variable impedance control:** The following control law is defined to establish the desired Cartesian impedance behavior on the tooltip.

$$\begin{aligned} \tau_{imp} &= J^T(q)(K_C\tilde{x} + D_C\dot{\tilde{x}} + M_C(q)\ddot{x}_d + C_C(q, \dot{q})\dot{x}_d), \\ \tilde{x} &= x_d - x, \end{aligned} \quad (3)$$

where $x \in \mathbb{R}^6$ is the actual pose of the end-effector in the base frame, and the pose error is \tilde{x} . Moreover, K_C and $D_C \in \mathbb{R}^{6 \times 6}$ are diagonal stiffness and damping matrices, respectively. The desired Cartesian inertia is assumed to be the robot inertia in Cartesian space. The difference between the desired and actual contact leads to a deviation from the desired pose x_d , which alters \tilde{x} , or the exerted force f_{ext} . Therefore, we use a metric h to adapt the stiffness matrix K_C in impedance control according to the external force and the pose error:

$$S = \|f_{ext}^T \tilde{x}\|, \quad (4)$$

$$h = 1 - \frac{S}{S_t}. \quad (5)$$

Here, having the threshold S_t is crucial to compensate for minor effects of the environment, i.e., surface friction and measurement error¹. The h is then coupled to K_C via ρ_{imp} :

$$K_C = \rho_{imp}(t)K_{max}, \quad (6)$$

where the adaptation parameter ρ_{imp} is obtained by

$$\rho_{imp} = \begin{cases} \min\{\rho, 0\}, & \rho_{imp} = 1 \\ \rho, & 0 < \rho_{imp} < 1, \rho_{imp}(0) = 0, \\ \max\{\rho, 0\}, & \rho_{imp} = 0 \end{cases} \quad (7)$$

and ρ is given by

$$\rho = h\rho_{imp} + \rho_{min}. \quad (8)$$

Note that, to have an initial increment for the case $\rho_{imp} = 0$, a small positive constant ρ_{min} has been introduced into the shaping function dynamics.

¹Please note that using pose instead of velocity or acceleration leads to having a comparably less noisy signal

3) **Force control:** The force control is defined to maintain the desired contact force $\mathbf{f}_d^{ee} \in \mathbb{R}^6$ applied by the robot w.r.t. the external force $\mathbf{f}_{ext}^{ee} \in \mathbb{R}^6$ as follows

$$\boldsymbol{\tau}_{\text{frc}} = \rho_{\text{frc}} \mathbf{J}(\mathbf{q})^T \mathbf{f}_{\text{frc}}, \quad (9)$$

$$\mathbf{f}_{\text{frc}} = \begin{bmatrix} [\mathbf{R}_{ee}^O]_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & [\mathbf{R}_{ee}^O]_{3 \times 3} \end{bmatrix} (\mathbf{f}_d^{ee} + \mathbf{K}_p \tilde{\mathbf{f}}_{\text{ext}} + \mathbf{K}_i \int \tilde{\mathbf{f}}_{\text{ext}} dt + \mathbf{K}_d \dot{\tilde{\mathbf{f}}}_{\text{ext}}), \quad (10)$$

$$\tilde{\mathbf{f}}_{\text{ext}}^{ee} = \mathbf{f}_d^{ee} + \mathbf{f}_{\text{ext}}^{ee}, \quad (11)$$

where $\mathbf{f}_{\text{frc}} \in \mathbb{R}^6$ is a feedback force controller in the base frame rotated by \mathbf{R}_{ee}^O . The PID controller gains are the diagonal matrices of \mathbf{K}_p , \mathbf{K}_i , $\mathbf{K}_d \in \mathbb{R}^{6 \times 6}$. Moreover, the force shaping function ρ_{frc} decides to activate or deactivate the force controller based on the defined conditions

$$\rho_{\text{frc}} = \rho_{\text{imp}} \rho_{\text{pc}} \rho_c. \quad (12)$$

To avoid phase switching, we define the force shaping function ρ_{frc} by combining ρ_{pc} and ρ_c . When the tool is close to the desired contact surface by $\delta_{\text{pc}} > 0$, force control is activated smoothly, and its weight is equal to one at \mathbf{x}_{min} . During the contact, the robot tolerates the tool alignment error up to a certain threshold of $\delta_c > 0$. For instance, force control is deactivated if the robot loses contact with the surface due to a large tool alignment error. Thus, the robot becomes only impedance-controlled and tracks the desired motion. Pre-contact shaping ρ_{pc} is designed based on the distance between the tool and the desired contact surface in the z-direction in the task frame $\tilde{\mathbf{x}}_c = \mathbf{x}_c - \mathbf{x}$:

$$\rho_{\text{pc}} = \begin{cases} 1, & x_{\text{min}} \leq \tilde{\mathbf{x}}_{c,z} \\ \frac{1}{2} (1 + \cos((\frac{x_{\text{min}} - \tilde{\mathbf{x}}_{c,z}}{\delta_{\text{pc}}})\pi)), & x_{\text{min}} > \tilde{\mathbf{x}}_{c,z} \geq x_{\text{min}} - \delta_{\text{pc}} \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Contact shaping ρ_c as a function of tool alignment error $\mathbf{f}_d^T \tilde{\mathbf{x}}$ by bounding it with the limits of S_{min} and $\delta_c > 0$:

$$\rho_c = \begin{cases} 1, & \mathbf{f}_d^T \tilde{\mathbf{x}}_c \leq S_{\text{min}} \\ \frac{1}{2} (1 + \cos((\frac{\mathbf{f}_d^T \tilde{\mathbf{x}}_c - S_{\text{min}}}{\delta_c})\pi)), & S_{\text{min}} < \mathbf{f}_d^T \tilde{\mathbf{x}}_c \leq S_{\text{min}} + \delta_c \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

When the robot is fully compliant $\rho_{\text{imp}} = 0$, it becomes adjustable to the environment. After reaching its maximum stiffness, the robot is adapted to the current environmental conditions and restarts the desired motion from its current configuration.

Finally, the closed-loop equation for the overall system becomes the following

$$\mathbf{M}_C(\mathbf{q})\ddot{\tilde{\mathbf{x}}} + \mathbf{C}_C(\mathbf{q}, \dot{\tilde{\mathbf{x}}})\dot{\tilde{\mathbf{x}}} + \mathbf{D}_C\dot{\tilde{\mathbf{x}}} + \mathbf{K}_C\tilde{\mathbf{x}} + \mathbf{f}_{\text{frc}} + \mathbf{f}_{\text{ff}} + \mathbf{f}_{\text{ext}} = \mathbf{0}. \quad (15)$$

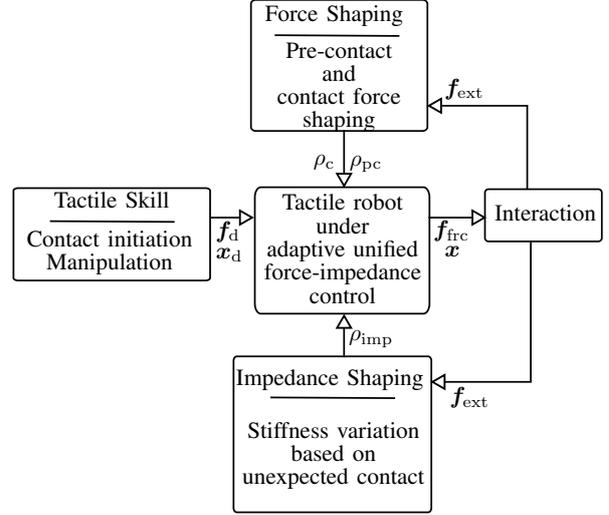


Fig. 4: **Tactile manipulation strategy.** The controller shaping functions ensure robust and safe interaction with the environment.

In addition, to ensure controller stability, which might be compromised due to variable impedance [29], the Lyapunov candidate function is

$$V = \frac{1}{2} \dot{\tilde{\mathbf{x}}}^T \mathbf{M}_C \dot{\tilde{\mathbf{x}}} + \frac{1}{2} \tilde{\mathbf{x}}^T \mathbf{K}_C \tilde{\mathbf{x}}. \quad (16)$$

Differentiating V and rearranging the corresponding terms for $\mathbf{f}_{\text{ext}} = 0$ and the constant \mathbf{M}_C result in

$$\dot{V} = -\dot{\tilde{\mathbf{x}}}^T \mathbf{D}_C \dot{\tilde{\mathbf{x}}} + \frac{1}{2} \tilde{\mathbf{x}}^T \dot{\mathbf{K}}_C \tilde{\mathbf{x}}. \quad (17)$$

Due to the term $\dot{\mathbf{K}}_C$, the eigenvalues of the stiffness matrix \mathbf{K}_C should be constant or decreasing for $\tilde{\mathbf{x}} \neq 0$. It can be deduced from Eq. 6 that the rate of change of the stiffness matrix $\dot{\mathbf{K}}_C$ is directly proportional to the rate of change of the adaption parameter $\dot{\rho}_{\text{imp}}$:

$$\dot{\mathbf{K}}_C = \dot{\rho}_{\text{imp}} \mathbf{K}_{\text{max}}. \quad (18)$$

Furthermore, once the motion error $\tilde{\mathbf{x}}$ exceeds defined thresholds in Eq. 4, 5, and 8, by the definition of $\dot{\rho}_{\text{imp}} = \min\{\rho, 0\}$ in Eq. 7, \mathbf{K}_C decreases due to $\dot{\rho}_{\text{imp}} < 0$. Stiffness \mathbf{K}_C increases again; only the motion error and external force are within the thresholds. Briefly, while the motion error $\tilde{\mathbf{x}}$ exists, the stiffness \mathbf{K}_C decreases:

$$\mathbf{K}_C = \begin{cases} \text{constant stiffness } \mathbf{K}_{\text{max}}, & \rho_{\text{imp}} = 1 \wedge \tilde{\mathbf{x}} = 0 \\ \text{decreases,} & \tilde{\mathbf{x}} \neq 0 \\ \text{increases to } \mathbf{K}_{\text{max}}, & \rho_{\text{imp}} \neq 1 \wedge \tilde{\mathbf{x}} = 0 \end{cases} \quad (19)$$

Please note that one might also install virtual energy tanks to ensure the passivity of the whole system as a sufficient condition to achieve stability [30].

TABLE I: Control parameters used in the experiments.

Parameter	Unit	Value
\dot{x}_z	m/s	0.002
$K_{x,imp}$ and $K_{x,afc}$	N/m	diag[1000, 1000, 1000, 150, 150, 150]
ξ	Ns/m	diag[0.7, 0.7, 0.7, 1, 1, 1]
f_d^{ee}	N	[0, 0, 10, 0, 0, 0]
K_p, K_i, K_d	—	$2.0I_{6 \times 6}, 1.0I_{6 \times 6}, 0I_{6 \times 6}$
$K_{x,ffc}$	N/m	diag[1000, 1000, 100, 150, 150, 150]
S_t	Nm	1.0
δ_{pc}	m	0.002
δ_c	Nm	0.5
x_{min}	m	-0.001
S_{min}	Nm	0.002

III. EXPERIMENTAL VALIDATION

In particular, the protocol for recycling the battery from a heat cost allocator is: (i) placing the tool in contact with the gap (pre-contact and contact initiation); (ii) pushing the pin; (iii) levering the lid and PCB, and (iv) separating the battery. In a simplified form, the requirements of the dismantling protocol are (i) contact initiation, going to the gap with a specific orientation; (ii) establishing contact, tool alignment with the desired contact (gap); and (iii) manipulation: force and motion profile. Finally, the task performance is evaluated regarding position accuracy, motion profile error, force tolerance, force profile error, and compliance. We formulate contact initiation and establish contact under perception uncertainties as a peg-in-hole problem to obtain a reproducible reference setup to measure the controller performances. A Franka Emika robot is used for the experiments, and the robot’s internal sensing records the position, velocity, and external end-effector force.

A. Contact Initiation: Peg-in-hole

A peg manufactured with fitting tolerance (< 0.1 mm) can be inserted via wiggle motion. In experiments, wiggle motion is realized by adding a feed-forward force term to the controller as a function of amplitude $a = -3$ N, frequency $\omega = 3$ Hz, and time t ,

$$\mathbf{f}_{ff} = [0, a \cos(2\pi\omega t), 0, 0, 0, 0]. \quad (20)$$

The length and diameter of the peg are 20 mm and 3 mm, respectively.

Experiment 1: In the first set of experiments, four different control methods are compared: (i) impedance, (ii) force, (iii) force-impedance, and (iv) adaptive force-impedance control. The robot starts with non-contact 2 mm above the hole. Afterward, it freely moves and establishes contact. Finally, the robot is expected to insert the peg until it is in contact with the bottom of the hole. The control parameters used in the experiments are listed in Tab. I.

Experiment 2: The application of force-impedance control promises to enable coping with displacement imprecision, especially for interaction tasks. We introduce the *soft displacement* metric to quantify this capability. This metric

analyses the displacement from the ideal insertion line for a peg-in-hole application starting from which the peg-in-hole operation is still successful based on robot compliance. The time to complete the task and the peak forces can be used to evaluate the controller’s quality. In the experiment, the initial robot position \mathbf{x}_0 is displaced towards the location of the hole, \mathbf{x}_c , by $\delta_{pc} = 2$ mm in the z-direction:

$$\mathbf{x}_{c,z} = \mathbf{x}_{0,z} - \delta_{pc}. \quad (21)$$

Once the robot is compliant, the initial pose is updated with the current pose. Thus, the robot reactivates the force controller within the defined thresholds. The experiments are repeated three times, using the displacements 3, 9, and 15 mm in the y-direction successively.

Experiment 3: We present a possible use case for our method in human-robot interaction to enable collaborative working. The impedance shaping function decreases the robot’s stiffness when a human applies force. Using this compliant behavior, the human expert puts the robot into another contact, and then the robot is expected to insert the peg.

IV. RESULTS AND DISCUSSION

The following section presents our experimental results in order of the three conducted experiments. All results are plotted based on the distance between the initial and current end-effector position w.r.t the base frame. As the peg length is of 20 mm, it is also the maximum distance the robot may travel in the z-direction. Therefore, if 20 mm distance along the z-direction and an approximate velocity of 0 mm/s are reached, we rate the task completion as successful. Additionally, the task phase plot is developed with the external force and velocity in the z-direction, which also translates to the power evolution during the task. It is presented in the robot task space.

The results of position and force over time and the task phase plots for **Experiment 1** are shown in Fig. 5.

For controller the impedance controller, Fig. 5a), at first, the wiggle motion is dominant in the position plot. Then, the insertion starts, and the corresponding external forces increase. The robot successfully inserts the peg after 13 s. However, there is no significant contact force until after 10 s, the robot senses the external force due to tolerance adjustment during insertion. After hitting the bottom of the hole, the robot measures a force value of around 10 N. Moreover, as there is no specified goal, the robot tries to move further at the bottom and gives a Cartesian reflex error.

Using the force controller shows that the robot controls the force well below the desired maximum of 10 N, Fig. 5b). However, as can be seen in the task phase and position plot, the robot fails to slow down and gives an error after 0.2 s due to embedded safety around 150 mm/s.

Next, in Fig. 5c) unified force-impedance control is applied, which commands low stiffness in the insertion direction to utilize compliant behavior as well as the tactility of the robot. Similarly to the force controller results, the force-impedance controlled force stays mainly below the

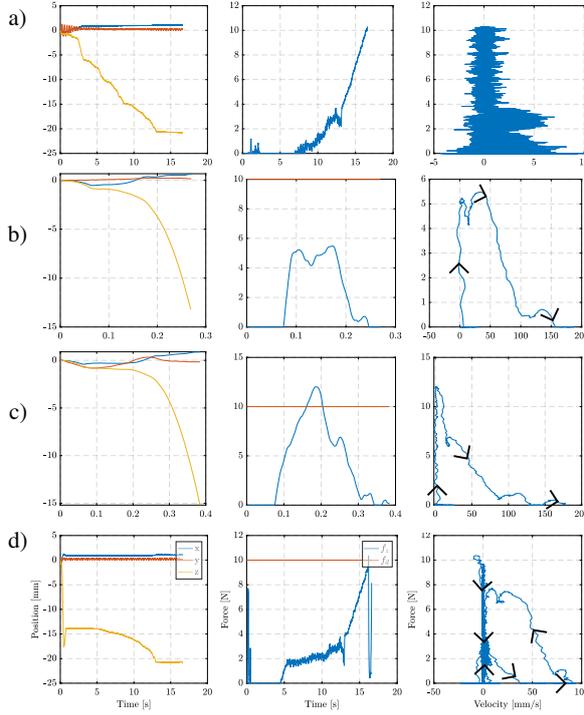


Fig. 5: Results of Experiment 1: Comparative results for different controllers. Position (current distance to the initial position in the x, y, and z direction in the base frame) vs. time, force in the z-direction in the task frame vs. time, and force in the z-direction in the task frame vs. velocity in the task frame (task phase plot) Arrows proceed in time. a) Impedance control b) Force control with 10 N in the z-direction c) Force impedance control with 10 N in the z-direction d) Adaptive force impedance control with 10 N in the z-direction

desired threshold, besides a short overshoot with 0.05 s duration. Nevertheless, just like the force controller, the force-impedance control cannot close the loop in the task phase plot without a specified end condition.

Finally, unified force-impedance control with adaptive stiffness is tested, shown in Fig. Fig. 5d). If there is no contact, the controller acts as pure impedance control. When the robot is close to the hole by δ_{pc} , force control is activated, and its weight reaches 1 after x_{min} . When the peg enters the hole without obstruction, as there is no contact to maintain due to the tool alignment error, the force control's weight decreases. Thus, the robot becomes only impedance-controlled and tracks the desired linear motion in the z-direction. Whenever the robot reaches the bottom, it stops due to the low stiffness, as seen in the task phase plot.

Using adaptive force-impedance control, in **Experiment 2**, we observe how compliant behavior allows to correct previous robot tool displacement of varying distances to achieve the peg-in-hole even if the robot is imperfectly positioned, as shown in Fig. 6.

Overall, in all cases, the peg-in-hole task is successfully conducted, regardless of the initial displacement of 0, 3, 9,

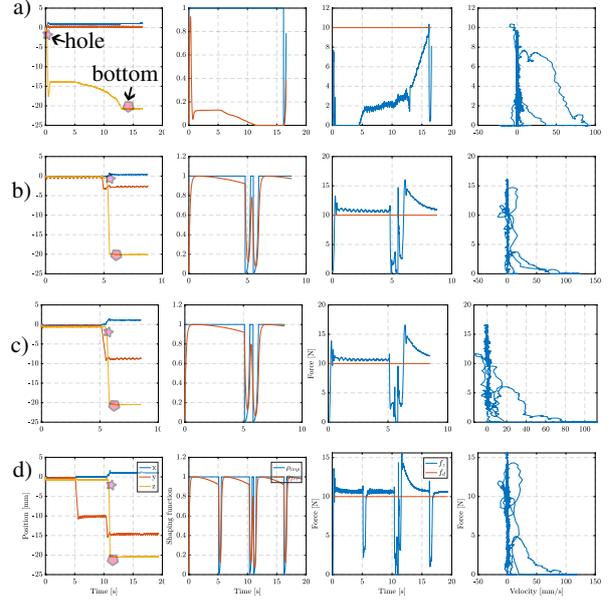


Fig. 6: Results of Experiment 2: Soft displacement. Using adaptive force-impedance control, starting at various distances in the y direction in the base frame to the center of the hole. Position (current distance to the initial position in the x, y, and z-direction in the base frame) vs. time, force-impedance shaping function vs. time, force in the z-direction in the task frame vs. time, and force in the z-direction in the task frame vs. velocity in the task frame (task phase plot) a) 0 mm b) 3 mm c) 9 mm d) 15 mm

or 15 mm as shown in Fig. 6 a), b), c), and d), respectively. In the position plot, the robot's motion along the y-direction shows the distance from the hole where the robot starts to move. The robot begins the wiggle motion, and the stiffness becomes low due to external force and motion error. Using the momentum caused by the feed-forward force, the robot moves towards the hole compliantly. Notably, the direction of the feed-forward force influences the soft displacement experiments, and it should be towards the hole. This procedure repeats until the robot reaches the desired contact at the bottom of the hole.

In **Experiment 3**, the robot encounters an additional contact with the human operator while inserting the peg, shown in Fig. 6. At this contact, the robot becomes compliant again owing to stiffness adaptation. After this contact is released, the robot adjusts to the new environment and updates the initial position with the current one. The force control is again activated to maintain contact up to a certain threshold δ_c , as shown in the force plots in Fig. 6b), c), and d). This helps the robot restart applying force as we choose the desired contact 2 mm away from the initial position. In other words, the robot can perceive touch as a human does with a fingertip.

Incidental contact with the human expert leads to a decrease in the robot's stiffness. Thus, the human moves the robot to another region of interest as if in guiding mode. The human expert puts the robot into another contact to insert the

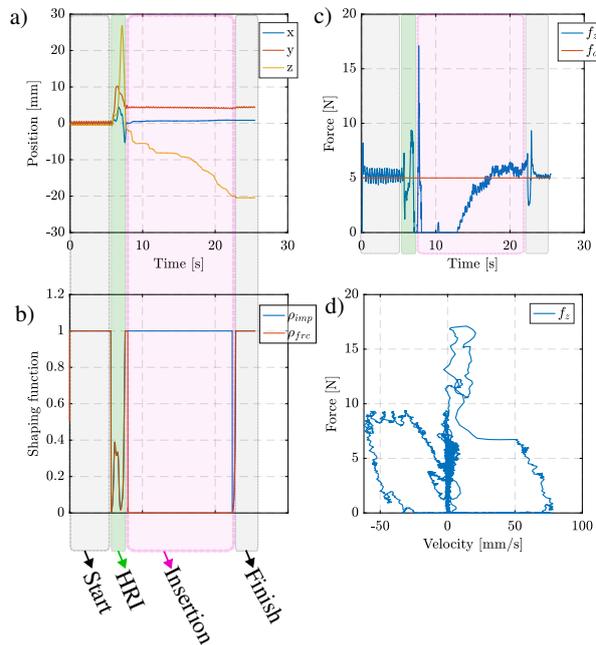


Fig. 7: Results of Experiment 3: Results for human-robot interaction using adaptive unified force-impedance control a) Distance between the end-effector's current and initial position in x, y, and z-direction b) Force-Impedance shaping functions c) Force vs. time d) Task phase plot

peg, as shown in Fig.7. According to ISO 15066, the robot's velocity can be considered within the safe limits for hand-guided human-robot collaboration (250 mm/s) [31].

The limitation of the presented control scheme is to design the control parameters wisely beforehand. For instance, δ_{pc} should not be smaller than the vertical distance to the surface.

V. CONCLUSION

Digital transformation of traditional factories requires robots to perform tactile skills under varying conditions. However, a tactile manipulation strategy is yet to be developed to handle unexpected situations and inaccurate environment models. Therefore, we propose designing such a manipulation strategy by analysis of the underlying system. For that, we apply the inherent representation of manipulation tasks, namely the force-velocity task phase plot, to identify relevant phases and enable the design of controllers that can smoothly adapt between these phases. Based on this representation, we design a unified force-impedance control method for reliable contact initiation and flexible manipulation under positioning and environmental uncertainties. First, based on the external force applied to the robot and the skill motion error, we design a dynamic process for impedance shaping to enable the robot to adjust its end-effector in response to unforeseen contact. Second, we establish a force shaping function that includes (i) a smooth transition from free motion to contact (pre-contact shaping) and (ii) force regulation based on the desired tool alignment (contact

shaping). Third, to compare our results with a reference setup, we observe our control scheme's capability to succeed in the desired task under positioning inaccuracy using the example of a peg-in-hole skill.

Furthermore, we compare four controllers to evaluate the success rate based on their respective task phase plots. As a result of the controller comparison, the importance of commanding force and motion policies with compliant behavior becomes apparent. Finally, we show how our force-impedance shaping in unified force-impedance control increases robustness for contact initiation and flexibility during manipulation. We also demonstrate a potential extension of our method to human-robot interaction.

The presented work shows the potential of the force-velocity task phase plot as a foundation for robotic skill design and comparison based on peg-in-hole insertion and removal. Future work will investigate in depth the applicability of this manipulation representation for creating versatile tasks.

REFERENCES

- [1] A. Lambert and S. Gupta, "Disassembly modeling for assembly, maintenance, reuse and recycling," 12 2004.
- [2] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada, "The evolution of robotics research," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 90–103, 2007.
- [3] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019. [Online]. Available: <https://www.science.org/doi/abs/10.1126/science.aat8414>
- [4] S. Haddadin, L. Johannsmeier, and F. Díaz Ledezma, "Tactile robots as a central embodiment of the tactile internet," in *Proceedings of the IEEE*, vol. 107, no. 2, Feb. 2019, pp. 471–487.
- [5] M. Simonič, R. Pahič, T. Gašpar, S. Abdolshah, S. Haddadin, M. G. Catalano, F. Wörgötter, and A. Ude, "Modular ros-based software architecture for reconfigurable, industry 4.0 compatible robotic workcells," in *2021 20th International Conference on Advanced Robotics (ICAR)*, Dec 2021, pp. 44–51.
- [6] K. Karacan, H. Sadeghian, R. Kirschner, and S. Haddadin, "Passivity-based skill motion learning in stiffness-adaptive unified force-impedance control," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2022, pp. 9604–9611.
- [7] J. Lloyd and N. F. Lepora, "Goal-driven robotic pushing using tactile and proprioceptive feedback," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 1201–1212, 2022.
- [8] K. Karacan, D. Grover, H. Sadeghian, F. Wu, and S. Haddadin, "Tactile exploration using unified force-impedance control," Jul 2023, p. under publication, 22nd IFAC World Congress.
- [9] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin, "The franka emika robot: A reference platform for robotics research and education," *IEEE Robotics & Automation Magazine*, vol. 29, no. 2, pp. 46–64, 2022.
- [10] R. J. Kirschner, A. Kurdas, K. Karacan, P. Junge, S. Birjandi, N. Mansfeld, S. Abdolshah, and S. Haddadin, "Towards a reference framework for tactile robot performance and safety benchmarking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4290–4297.
- [11] E. Shahriari, A. Kramberger, A. Gams, A. Ude, and S. Haddadin, "Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 136–142.
- [12] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American Control Conference*, 1984, pp. 304–313.
- [13] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robotics Autom.*, vol. 3, pp. 43–53, 1987.

- [14] C. Schindlbeck and S. Haddadin, "Unified Passivity-Based Cartesian Force / Impedance Control for Rigid and Flexible Joint Robots via Task-Energy Tanks," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 440–447, 2015.
- [15] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse, "Collaborative manufacturing with physical human–robot interaction," *Robotics and Computer-Integrated Manufacturing*, vol. 40, pp. 1–13, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0736584515301769>
- [16] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human–robot physical interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, Aug 2015.
- [17] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 620–629, March 2016.
- [18] F. Kulakov, G. V. Alferov, P. Efimova, S. Chernakova, and D. Shymanchuk, "Modeling and control of robot manipulators with the constraints at the moving objects," in *2015 International Conference "Stability and Control Processes" in Memory of V.I. Zubov (SCP)*, Oct 2015, pp. 102–105.
- [19] C. Ott, A. Dietrich, and A. Albu-Schäffer, "Prioritized multi-task compliance control of redundant manipulators," *Automatica*, vol. 53, pp. 416–423, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109815000163>
- [20] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, "Towards associative skill memories," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Nov 2012, pp. 309–315.
- [21] A. Kramberger, A. Gams, B. Nemeč, C. Schou, D. Chrysostomou, O. Madsen, and A. Ude, "Transfer of contact skills to new environmental conditions," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 668–675.
- [22] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 493–506, 2013.
- [23] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE transactions on robotics*, vol. 27, no. 5, pp. 918–930, 2011.
- [24] F. J. Abu-Dakka and M. Saveriano, "Variable impedance control and learning—a review," *Frontiers in Robotics and AI*, vol. 7, 2020. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/frobt.2020.590681>
- [25] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, "Skill learning and task outcome prediction for manipulation," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3828–3834.
- [26] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1711–1718, 2018.
- [27] M. Qin, J. Brawer, and B. Scassellati, "Robot tool use: A survey," *Frontiers in Robotics and AI*, vol. 9, 2023. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/frobt.2022.1009488>
- [28] M. Suomalainen, Y. Karayiannidis, and V. Kyrki, "A survey of robot manipulation in contact," *Robotics and Autonomous Systems*, vol. 156, p. 104224, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889022001312>
- [29] K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, 2016.
- [30] E. Shahrari, S. A. B. Birjandi, and S. Haddadin, "Passivity-based adaptive force-impedance control for modular multi-manual object manipulation," *IEEE Robotics Autom. Lett.*, vol. 7, no. 2, pp. 2194–2201, 2022. [Online]. Available: <https://doi.org/10.1109/LRA.2022.3142903>
- [31] DIN ISO/TS 15066:2016-02, Robots and robotic devices — Collaborative robots (ISO/TS 15066:2016).

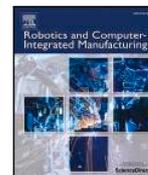
A.3 “Hierarchical learning of robotic contact policies”

This is a copy of an open-access article: M. Simonič, A. Ude, and B. Nemec. “Hierarchical learning of robotic contact policies”. In: *Robotics and Computer-Integrated Manufacturing* 86 (2024), 102657, 12 pages.



Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

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

Full length article

Hierarchical learning of robotic contact policies

Mihael Simonič^{a,b}, Aleš Ude^{a,b}, Bojan Nemeč^{a,*}

^a Dept. of Automatics, Biocybernetics, and Robotics, Jožef Stefan Institute, Jamova c. 39, Ljubljana, 1000, Slovenia

^b Faculty of Electrical Engineering, University of Ljubljana, Tržaška c. 25, Ljubljana, 1000, Slovenia

ARTICLE INFO

Keywords:

Autonomous robot learning
Learning from experience
Compliance and impedance control
Perception-action coupling
Compliant assembly

ABSTRACT

The paper addresses the issue of learning tasks where a robot maintains permanent contact with the environment. We propose a new methodology based on a hierarchical learning scheme coupled with task representation through directed graphs. These graphs are constituted of nodes and branches that correspond to the states and robotic actions, respectively. The upper level of the hierarchy essentially operates as a decision-making algorithm. It leverages reinforcement learning (RL) techniques to facilitate optimal decision-making. The actions are generated by a constraint-space following (CSF) controller that autonomously identifies feasible directions for motion. The controller generates robot motion by adjusting its stiffness in the direction defined by the Frenet-Serret frame, which is aligned with the robot path. The proposed framework was experimentally verified through a series of challenging robotic tasks such as maze learning, door opening, learning to shift the manual car gear, and learning car license plate light assembly by disassembly.

1. Introduction

Many robot tasks require tight contact with the environment. Such tasks are common in industrial environments, e.g. in assembly operations and in domestic environments, where the robot has to perform operations like opening doors and drawers to access different rooms or objects. They are generally considered hard to learn, as the robot needs to learn a policy composed of poses and wrenches while interacting with an unknown and possibly changing environment. Imitation learning is a widely used paradigm to effectively specify tasks in contact with the environment [1]. However, involving a human teacher in the learning process is not always desirable. Especially for robots operating in unstructured environments, it is often beneficial if they can learn new contact skills by themselves. This requires lengthy task adaptation procedures, which are usually realized based on reinforcement learning (RL) [2] or iterative learning control (ILC) [3].

While user-friendly programming approaches and hardware reconfigurability capabilities have long been used to enhance the capabilities of industrial applications [4,5], autonomous learning is still considered too time-consuming for such settings. Lengthy policy learning and refinement processes hinder the practical application of autonomous learning algorithms and the deployment of robots in unstructured and complex industrial environments. For applications in flexible, small-scale production, characterized by a wide variety of assembly tasks, it is very important to reduce the programming effort and the required skill level of the operator. This problem is even more pronounced when

introducing robots into inherently unstructured home environments, where we cannot expect the help of experienced operators.

The aim of this paper is to introduce a new methodology that enables a robot to quickly and autonomously acquire new contact skills, even without previous imitation learning. To this end, we propose a new approach to learning robotic tasks where physical contact with the environment contributes to faster learning. The proposed approach is based on the observation that learning physically constrained tasks, can be structured more efficiently than learning tasks where a robot moves freely in space. The reason for this is that the environment constrains the admissible movement directions, thereby limiting the search space. Consequently, the number of parameters that need to be learned is significantly reduced. To implement this type of learning, we need to allow the environmental constraints to determine the robot's motion. The concept of compliant robot control provides a suitable framework for implementing such a strategy. The early stage of this concept was applied in our previous work, where we demonstrated how robots could learn tasks with physical constraints, e.g., opening doors and drawers [6], and how to learn assembly tasks by disassembly [7].

The main contribution of this paper is a comprehensive framework for autonomous learning of complex skills where the robot maintains contact with the environment. The key components of this framework are:

- Graph-based task representation: We propose a graph-based representation that enables hierarchical decomposition of complex

* Corresponding author.

E-mail addresses: mihael.simonic@ijs.si (M. Simonič), ales.ude@ijs.si (A. Ude), bojan.nemec@ijs.si (B. Nemeč).

<https://doi.org/10.1016/j.rcim.2023.102657>

Received 12 September 2022; Received in revised form 7 June 2023; Accepted 8 September 2023

Available online 18 September 2023

0736-5845/© 2023 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

contact policies. This representation allows for efficient and fully autonomous learning of these skills.

- Constraint-space following (CSF) controller: The paper introduces a CSF controller that facilitates autonomous exploration within constrained spaces. This controller relies on robust estimation of the tangential direction of motion and utilizes the Frenet–Serret (FS) formulas to specify variable compliance along the robot’s motion trajectory.
- Graph topology determination algorithm: The paper presents a novel algorithm that autonomously discovers the graph topology of the task representation. This algorithm surpasses the previous approach [7] in terms of execution speed, allowing for faster learning.
- Learning the optimal sequence of movements: The framework includes RL algorithm for learning the optimal sequence of movements within the graph representation of the task. This way, the robot can efficiently perform complex contact tasks.

In summary, the paper contributes to the field of autonomous robotics by presenting a comprehensive framework that combines a graph-based task representation, an algorithm for graph topology discovery, optimal control policy learning, and a CSF controller. To the best of our knowledge, the proposed methodology is the first framework capable of entirely autonomous learning of tasks where the environment constrains the robot’s motion. Moreover, the learning speed is comparable to that of humans. The validity of the proposed approach has been demonstrated by learning four challenging tasks taken from everyday life and industrial environments.

This paper consists of five sections. In Section 2, we briefly review existing approaches to learning contact policies. Our main contribution is a new scheme for learning such policies based on a Hierarchical Reinforcement Learning (HRL), which is presented in Section 3. The experimental evaluation presented in Section 4 considers four challenging tasks: maze learning, door opening, learning to shift a manual car gearbox, and learning to assemble a car license plate by disassembly. The discussion and final conclusions are provided in Section 5.

2. Related work

A lot of research on learning tasks that involve contact-rich manipulation in unstructured environments has been performed in the past. A comprehensive survey paper on robots performing manipulation tasks that require varying contact with the environment has been published recently [8]. This section focuses on learning-based approaches [9].

The most basic and straightforward approaches are based on the Iterative Learning Control paradigm (ILC), which has been used to adapt the trajectory of tasks involving contact with the environment [10, 11]. These approaches minimize force tracking errors but cannot improve the desired force–torque contact profile. More general are the approaches based on reinforcement learning (RL), which has been successfully applied for learning tasks such as door opening and picking a pen from the table [12]. However, this type of learning requires approximately a hundred trials to learn the policy and is slow compared to humans. Learning of contact-rich tasks is closely related to the learning of impedance parameters, as proven in [13], where the authors addressed deep learning in action space. Imitation can increase learning speed, as demonstrated in the wood planing experiment [14], where RL enhanced the skill transfer from humans to robots. In [15], the authors proposed Guided Policy Search to handle contact-rich tasks. Initialized by demonstration and optimized by PI^2 RL algorithm, the local policies were used to generate a global policy using deep neural networks (DNN) and robot joint torques directly from the visual input. A DNN was also applied for learning a peg-in-hole (PIH) task without prior demonstration [16], where a robot learned the desired search and insertion policy using the deep Q-learning algorithm. After 100–200 trials, the robot learned a robust search and insertion policy. To increase

the data efficiency and learning speed, a hierarchical RL scheme was proposed and applied to a dual PiH task [17]. PiH was also addressed in [18], where the authors applied movement primitives encoded as a neural network and a deep deterministic policy gradient algorithm for learning neural network parameters. The learning efficiency was also pursued in [19], where the authors proposed learning meta-parameters to encode a specific skill in combination with an adaptive impedance controller. Another recent approach to autonomous learning of contact tasks is based on observing time-reversed visual cues, enabling one to learn the policy without previous demonstration or exploration [20]. Very similar objectives were also pursued in the recent study [21], where the aim was to learn a shape descriptor that establishes geometric correspondences between object surfaces and their target locations directly from the visual stream.

In this paper, we propose a new approach to learning contact-rich tasks based on a hierarchical learning scheme. In this respect, our approach is related to the approach described in [17], except that in our case, the highest hierarchical level is decision learning, and the lowest level includes a variable compliance controller to move along the physical constraints of the environment. It is also partly related to the approaches proposed in [20,21], as we also learn the reverse policies of assembly tasks [7]. There are also some similarities between our research and [19], where they also used a directed graph to represent the learned policy. However, there is a significant difference between the two approaches in the purpose and construction of the graph.

One of the main advantages of our approach is its high learning speed, which exceeds the performance of reinforcement and deep reinforcement learning methods that do not exploit the constraints of contact tasks. The high learning speed is achieved by a meaningful decomposition of the task into the determination of environmental constraints and the decision level, which is implemented by a hierarchical learning scheme. In addition, the proposed graph-based task representation provides a better insight into the nature of the problem. The proposed algorithm does not require prior knowledge of this graph but builds it autonomously through exploration.

3. Learning of contact policies with hierarchical learning

Our approach to learning contact tasks exploits the configurations that allow only partial freedom of motion, both in terms of position and orientation. The boundary between the region where the robot’s motion is constrained by the environment and the region where the motion is free is called a *C-surface* [22]. Motion is possible along the tangential directions of the C-surface and constrained along the orthogonal directions. The dimension of the C-surface determines the number of degrees of freedom of a physically feasible robot motion. Our research considers tasks with one dimensional C-surface, which is typical for assembly tasks. We further assume that the task consists of different motion primitives that can be combined to form a longer action sequence. Each motion primitive is associated with a one-dimensional C-surface. The sequence of motion primitives can be such that we have more choices on how to proceed with the task. For example, consider the shifting of car gears, as discussed in Section 4.3. When moving the gear lever from neutral to the left, we have two options: turn up to the first gear or turn down to the second gear. Such a sequence with branches can be represented with a graph, where nodes represent various key states of the task and edges represent motion primitives constrained by the environment [7].

An example of such a graph is shown in Fig. 1. The nodes represent the key states and the edges represent the robot movements between the key states of the task. The start node (colored yellow) can be anywhere in the graph and denotes the initial state of the task. The node is characterized by how many branches start from each node, called the degree of the node. In a node of degree 2 or more (colored orange), the robot must decide how to proceed. In a node of degree 1 (colored blue), we do not have multiple options of how to continue the task, but

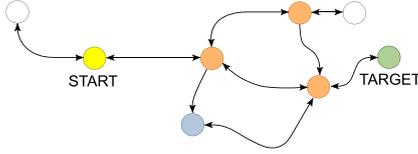


Fig. 1. The proposed graph-based representation of contact tasks in a general form.

to continue we must choose a completely different direction or type of motion; for example, a translational motion changes to a rotational or vice versa. A node of degree zero (colored white) represents stages of the task where the motion can only be continued by turning back. The goal is to reach the target node (colored green).

Graphs are often used to represent sequences of assembly and disassembly motions [23–25]. In general, a graph, possibly directed, is a conceptual representation of a sequence of activities. In our approach, a graph represents a single, optimal policy for the execution of a contact task. By joining multiple such graphs, more complex tasks that consist of several contact tasks can be represented (as shown in Fig. 12, where a human demonstrated the necessary movement between two contact tasks).

Let us now assume that we do not know how to execute a contact task in advance, but we do know the starting point of the task and what the target state is. Consequently, we do not know the topology of the graph, the intermediate nodes, and the actions for the transition between individual nodes. Thus the robot should learn this through autonomous exploration. Furthermore, it has to learn the optimal policy from the start node to the target node, i.e. the optimal sequence of movement primitives to accomplish the given task.

3.1. Hierarchical reinforcement learning

The complexity of learning problems can sometimes be reduced by hierarchical learning schemes, which split the learning problem into sub-tasks with multiple levels of hierarchy. In this section, we explain our hierarchical scheme (see Fig. 2) on the example of maze learning, where the goal is to move from the start to the target node (see Fig. 5 and Fig. 6). Note that the maze constrains the possible robot motion. It is natural to use a graph representation to represent the points in the maze where there are multiple directions in which the robot can continue its motion or where the robot can only continue its motion by turning back.

The graph representation of the task is not known in advance but must be learned. We initialize the learning process with a graph that has only one node, i.e. the start node. Each node is associated with the robot pose in this state (and, in some cases, several preceding poses). The initial node is associated with the robot pose at the beginning of the task execution. Using the proposed approach, the robot autonomously explores the environment along its constraints (lowest level) and identifies new states (nodes) and actions (edges) that cause a transition between the states (middle level). The actions are represented by the motion trajectories that specify the robot motion between the connected nodes. The reinforcement learning algorithm at the top of the hierarchical scheme learns the optimal movement sequence from the start to the target node.

On the shortcomings of traditional shortest path searching algorithms. The design of the search algorithm at the highest hierarchical level depends on the complexity of the problem. In less complex cases, it is sufficient to find the shortest path from the start node to the target node. In general, however, this is not sufficient; the robot must learn that it may be necessary to visit a specific node in the graph to reach the target node successfully or even that it may have to pass through a certain node more than once. Consider, for example, the previously mentioned

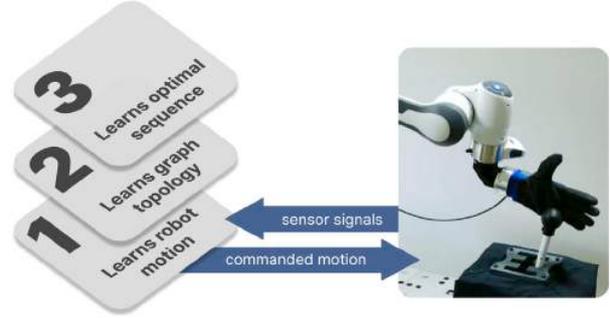


Fig. 2. The proposed hierarchical scheme for learning contact policies represented with graphs.

gear shifting case, addressed in detail in Section 4.3. The robot must not shift from the first gear directly to the fifth, which would be the shortest path in the graph, but must learn to shift continuously from a lower gear to a higher gear. To address such tasks, we chose reinforcement learning (RL) to guide the search at the highest hierarchical level. In some cases, however, it makes sense to use computationally more efficient graph search algorithms [26].

3.2. Lowest level: Constraint-space following controller

The detailed presentation of our hierarchical scheme starts from the bottom up by introducing the Constraint-Space Following (CSF) controller, which is used to move the robot's end-effector along the environmental constraints. At the lowest level, the controller is given a direction (selected by the algorithm at the highest level described in Section 3.4) in which the robot can start exploring the environment from the current node (state). Here the focus is on how to control the robot motion in the selected direction while exploring whether motion in any other direction is possible, which indicates the existence of the next node in the graph.

For this purpose, we developed an impedance controller that is stiff in the current direction of movement and compliant in the orthogonal directions. A similar approach was proposed in [22,27], where they introduced the concept of *Compliant frame* and *Task frame*. In our framework, we formalize this motion control by utilizing the *Frenet-Serret (FS) frame* [28,29], which is attached to the robot tool center point (TCP). A sequence of FS frames is illustrated in Fig. 3. An FS frame at position t_p is defined by a rotation matrix \mathbf{R}_p with the first column aligned with the tangential direction of motion, i.e. \dot{p} , and the other two columns orthogonal to it. They are referred to as normal and binormal vectors. \mathbf{R}_p and the corresponding coordinate axes can be computed as follows

$$\mathbf{R}_p = [t_p \quad n_p \quad b_p], \quad (1)$$

$$t_p = \frac{\dot{p}}{\|\dot{p}\|}, \quad b_p = \frac{\dot{p} \times \ddot{p}}{\|\dot{p} \times \ddot{p}\|}, \quad n_p = b_p \times t_p,$$

where $p \in \mathbb{R}^3$ are the measured robot TCP positions. For the rotational part of motion, the FS frame is defined as

$$\mathbf{R}_o = [t_o \quad n_o \quad b_o], \quad (2)$$

$$t_o = \frac{\omega}{\|\omega\|}, \quad b_o = \frac{\omega \times \dot{\omega}}{\|\omega \times \dot{\omega}\|}, \quad n_o = b_o \times t_o,$$

where $\omega = 2\bar{q} * \dot{q}$ is the angular velocity. Here \bar{q} denotes the conjugate quaternion and we exploit the fact that the product of a conjugate of unit quaternion with its own derivative results in a quaternion with zero scalar part, which can be interpreted as a vector.

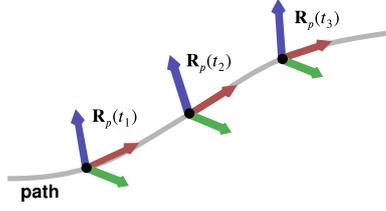


Fig. 3. Frenet-Serret frames $\mathbf{R}_p(t_k)$ for the positional path. The tangential direction is in red, the normal in green, and the binormal in blue.

The above equations require accelerations, which are usually low and therefore very noisy during operations like assembly and disassembly. Different measures were proposed to provide for a robust FS frame estimation, such as using Kalman filtering [30], Bishop frames [31], and an optimization-based calculation [32]. All these methods require knowledge of the entire trajectory before processing and thus cannot be used for a real-time FS frame calculation, as it is required by our approach. On the other hand, our policy learning algorithm requires a controller stiff in the tangential direction of motion and compliant in the normal and binormal direction of the corresponding FS frame. In most practical cases it is best to assume equal compliance in the normal and binormal direction. In such cases (see the lemma in the Appendix), the robot control torques are independent of the direction of the normal and binormal vector. Hence, given the tangential direction, the other two columns of the coordinate frame can be selected arbitrarily in the plane orthogonal to the tangential direction of motion. In our approach, we take the second and third columns of the coordinate frame from the previous sample and then apply a Gram-Schmidt orthogonalization procedure to obtain an orthonormal basis for the current coordinate frame. We call the resulting coordinate frames *modified FS frames*. The initial modified FS frame is computed with an arbitrary selection of vectors \mathbf{n} and \mathbf{b} that are further orthogonalized.

The remaining concern is how to estimate the tangential direction of motion in a robust way. Niemeyer and Slotine [33] proposed a spatial filter, which does not affect the normalization of the velocity vector. The spatial filter extended for the rotational movement was used for the tangent estimation in our previous research [6,7]. However, since the spatial filter is a first-order filter, it introduced a lag. In our experiments, it turned out that the main problem for tangent estimation is not the sensor noise but the compliance of the tool and the tolerances of the environment, which allow small motions in other directions, although the robot's motion is primarily constrained to only one direction. Therefore we applied a filter with a variable rate, inspired by the approach proposed in [34]. For the positional part of the motion, the tangent is estimated using

$$t_p(t_k) = \begin{cases} \frac{\mathbf{p}(t_k) - \mathbf{p}(t_{k-w})}{\|\mathbf{p}(t_k) - \mathbf{p}(t_{k-w})\|}, & \delta_p(t_{k-1}) \leq \epsilon_p \\ \frac{\mathbf{p}(t_k) - \mathbf{p}(t_{k-1})}{\|\mathbf{p}(t_k) - \mathbf{p}(t_{k-1})\|}, & \text{otherwise} \end{cases}, \quad (3)$$

where t_k is the current time, w a suitably chosen delay factor that determines the smoothing of the digital filter, $\delta_p(t_{k-1})$ is the distance between the line $[\mathbf{p}(t_k), \mathbf{p}(t_{k-w})]$ and position $\mathbf{p}(t_{k-1})$, and ϵ_p a constant that determines the switching between the filtered and non-filtered estimation of the tangent. To prevent filter chattering, a small hysteresis is usually applied to ϵ_p . The filtering of sampled points $\mathbf{p}(t_k)$ is illustrated in Fig. 4. For a rotational motion represented by a quaternion trajectory, the filter takes the form

$$t_o(t_k) = \begin{cases} \frac{2 \log(\mathbf{q}(t_k) * \bar{\mathbf{q}}(t_{k-w}))}{\|2 \log(\mathbf{q}(t_k) * \bar{\mathbf{q}}(t_{k-w}))\|}, & \delta_o(t_{k-1}) \leq \epsilon_q \\ \frac{2 \log(\mathbf{q}(t_k) * \bar{\mathbf{q}}(t_{k-1}))}{\|2 \log(\mathbf{q}(t_k) * \bar{\mathbf{q}}(t_{k-1}))\|}, & \text{otherwise} \end{cases}, \quad (4)$$

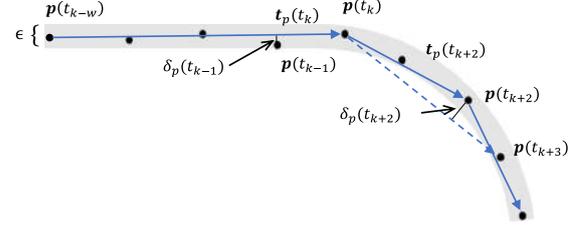


Fig. 4. Tangent estimation with the proposed variable rate filter. Solid dots represent the measured robot positions. The dotted line shows the candidate tangent at time t_{k+2} that was discarded because it does not meet the condition $\delta(t_{k+2}) \leq \epsilon_p$.

where \log is the quaternion logarithm.

Given the FS frame, we need a control law that enables the application of arbitrary compliance along the FS frame axes. Our approach is based on the passivity-based variant of impedance control for manipulators with flexible joints [35]. We implemented a modification to set the compliance along the Cartesian space trajectory with FS frames attached. The commanded torque $\rho_c \in \mathbb{R}^N$, which is passed to the robot motors, is calculated as

$$\rho_c = \mathbf{B}\mathbf{B}_\theta^{-1}\mathbf{u} + (\mathbf{I} - \mathbf{B}\mathbf{B}_\theta^{-1})\rho, \quad (5)$$

$$\mathbf{u} = \mathbf{J}^T(\theta) \left(\begin{bmatrix} \mathbf{f}_c \\ \mathbf{m}_c \end{bmatrix} + \begin{bmatrix} \mathbf{f}_s \\ \mathbf{m}_s \end{bmatrix} \right) + \mathbf{g}(\theta) + \mathbf{N}(\theta)\dot{\rho}_0, \quad (6)$$

where N is the number of robot joints, $\theta \in \mathbb{R}^N$ is the vector of joint angles computed from the motor angles $\Theta \in \mathbb{R}^N$ [35], $\mathbf{J} \in \mathbb{R}^{N \times 6}$ is the manipulator Jacobian, while \mathbf{B} , $\mathbf{B}_\theta \in \mathbb{R}^{6 \times 6}$ denote the positive definite diagonal matrices of the actual and the desired joint inertia, respectively. The aim of the term $\mathbf{B}\mathbf{B}_\theta^{-1}$ is to reduce the joint inertia. ρ is the vector consisting of the measured joint torques and $\mathbf{g}(\theta)$ is the gravity vector [36]. To control the configuration of the robot, a nullspace term is added [37], where $\mathbf{N}(\theta) = \mathbf{I} - \mathbf{J}^T(\theta)\mathbf{J}^{+T}(\theta) \in \mathbb{R}^{N \times N}$ is the null space projection operator, $\mathbf{J}^+(\theta)$ denotes Moore-Penrose pseudo-inverse of the Jacobian and $\dot{\rho}_0 \in \mathbb{R}^n$ is the null space joint torque vector. \mathbf{f}_s and \mathbf{m}_s are additional forces and torque vectors in task coordinates, which are used for searching for a feasible motion direction. The selection of the probing forces \mathbf{f}_s and torques \mathbf{m}_s depends on a task. It is discussed in Section 3.3 for planar maze learning and in Section 4 for other tasks. The motor torque controller (5) reduces the motor inertia and compensates for the robot's non-linear dynamics, while Eq. (6) provides for the desired impedance and damping, additional task force, gravity compensation, and null space motion. The task command input $[\mathbf{f}_c^T, \mathbf{m}_c^T]^T$ is computed as

$$\mathbf{f}_c = -\mathbf{R}_p\mathbf{D}_p\mathbf{R}_p^T\dot{\rho} + \mathbf{R}_p\mathbf{K}_p\mathbf{R}_p^T\mathbf{e}_p, \quad (7)$$

$$\mathbf{m}_c = -\mathbf{R}_o\mathbf{D}_o\mathbf{R}_o^T\dot{\omega} + \mathbf{R}_o\mathbf{K}_o\mathbf{R}_o^T\mathbf{e}_o, \quad (8)$$

where position and orientation tracking errors are defined as $\mathbf{e}_p = \mathbf{p}_d - \mathbf{p}$ and $\mathbf{e}_o = 2 \log(\bar{\mathbf{q}} * \mathbf{q}_d)$. \mathbf{K}_p and $\mathbf{K}_o \in \mathbb{R}^{3 \times 3}$ are the diagonal matrices defining the positional and rotational stiffness along and around coordinate axes, respectively. \mathbf{D}_p and $\mathbf{D}_o \in \mathbb{R}^{3 \times 3}$ are diagonal damping matrices, which are set as diagonal elements of the block diagonal matrix¹

$$\mathbf{D} = 2\sqrt{\mathbf{B}_\theta + \begin{bmatrix} \mathbf{K}_p & 0 \\ 0 & \mathbf{K}_o \end{bmatrix}}. \quad (9)$$

With the proposed approach, the robot is able to move along the environmental boundaries while probing for possible movements in

¹ In [35] the authors proposed double-diagonalization method to shape the damping. However, the resulting damping matrix is not diagonal, as our approach requires. Our experiments showed that the performance degradation due to diagonal damping matrices was negligible.

other directions, given that we apply high stiffness in the direction of motion and low gains in the orthogonal directions.

3.3. Middle level: Graph topology determination

The CSF controller described above enables the robot to move along the environmental constraints while applying probing wrenches that indicate the possibility of moving in directions other than the commanded tangential direction. The next task in our hierarchical learning scheme is to determine when the robot has arrived to the next node in the graph. When the robot arrives to the next node for the first time, the newly identified node should be added to the graph. In this section, we first define the representation of nodes and edges, outline the criteria for node detection, and present the algorithm that systematically explores the graph topology.

3.3.1. Representation of nodes and edges

Each node in the graph is characterized by the current robot pose and possibly some previous poses that the robot has reached in the nodes immediately preceding the current node. Such a sequence of poses defines a state s_k associated with the k -th node of the graph:

$$s_k = \{p_k, q_k, p_{k-1}, q_{k-1}, \dots, p_{k-\kappa}, q_{k-\kappa}\}, \quad (10)$$

where $p_k \in \mathbf{R}^3$ is the position and $q_k \in \mathbb{R}^4$ a unit quaternion representing the orientation. $0 \leq \kappa < k$, i.e. the number of poses in s_k , is chosen so that s_k satisfies the Markov decision property, i.e. each transition is determined only by the current node and the action selected in that node. To ensure the Markov decision property, it is sometimes necessary to include previous poses in the state description, i.e. $\kappa \geq 1$. This effectively means that some parts of the graph are explored again, since this results in the creation of new states (see Fig. 8). This way it becomes possible to solve problems such as opening locked doors, where some states only become reachable after a certain other state has been visited. In practice, the parameter κ is chosen manually by the user. In most of the experiments described in Section 4, κ was set to 0.

The transition from one node to another defines an edge in the graph. In the proposed system, a robot motion trajectory starting in one and ending in another node is encoded with a speed-scaled Cartesian space dynamic movement primitive (CDMP), which handles Cartesian space policies [38] and non-uniform velocity scaling [39]. The benefit of such encoding is twofold: it allows a compact, smooth, and scalable representation of the learned policy, and it removes the explicit time dependence of the trajectory. This enables the robot to slow down or speed up the execution of the learned assembly policy if needed [39]. The CDMPs starting in k -th node are stored in a set $\mathcal{A}(s_k) = \{a_k^i\}_{i=0}^d$, where d is the degree of the node. The i -th CDMP in state s_k is represented as a_k^i :

$$a_k^i = \{w_k^i, g_k^i, \tau_k^i\}, \quad (11)$$

where w_i are the CDMP weights, g_i is the CDMP goal, and τ_i is the temporal scaling factor of the CDMP. See [38] for more details.

3.3.2. Node discovery by examining each branch with different probing wrenches

We explain our search algorithm on the example of the planar maze learning problem, where there are at most two possible directions to continue at each node, other than turning back (see the graphs in Fig. 5). In the graph representation of such a maze, the maximum node degree is 2.

In our previous paper [7] we proposed an algorithm that discovers new graph nodes by probing for the possible crossings that indicate the nodes in small steps. However, this approach is slow because the robot must constantly stop to make probes with search wrenches in different directions. In the approach proposed in this paper, the maze is explored

using a method inspired by the wall following algorithm [40]. In this approach, each possible motion (branch in the graph) is examined twice by adding a constant probing force in either the positive or negative direction of the normal.² Next, we compare the robot paths executed during the two passes. If the path splits, we have found a new node of degree 2. If there is no split but the robot cannot continue in the same tangential direction, we have found the node of either degree 0 (where the robot can only turn back) or degree 1 (where the robot can continue only by changing the type of motion). The robot continues until it reaches the target node. To prevent the robot from entering a loop, we check that the robot has not already visited the same segment with the same probing force.

As explained above, the example shown in Fig. 5 treats maze learning as a planar problem. Therefore we only applied probing forces along the normal axis of the FS frame. An extension of the proposed algorithm to the 3D case, rotational motion, or nodes of a higher degree is straightforward. Depending on the problem, probing wrenches in all relevant directions must be given as input to the node discovery algorithm. For each additional probing wrench, additional searches from the currently visited node are necessary, which increases the search time. In a practical implementation, it is, therefore, advisable to consider only those search directions that are really necessary. For most problems, these can be defined by a user in advance. We provide some examples in Section 4.

3.3.3. Exploring the entire graph

The approach described above explores the selected branch with all possible probing forces. To ensure that the entire graph is explored, it is necessary to examine each node s_k and examine all possible directions d_k^i .

Initially, the problem description contains only the start node defined by the initial robot pose. The motion can be initiated by moving the robot in one of the four orthogonal directions aligned with the coordinates of the maze.³ In the proposed system, this is characterized as $\mathcal{D}(s_k) = \{d_k^i\}_{i=0}^n$, where d_k^i denotes the possible directions to continue the motion and n is the maximum number of them. In the case of the maze learning $n = 3$. In all our experiments, the initial set of search directions was the same for all nodes.

Starting from the first node, we search in all possible directions with all probing wrenches $[\mathbf{f}_s^T, \mathbf{m}_s^T]^T$ to find other nodes in the graph. If the robot cannot move in the given direction, d_k^i is removed from the set $\mathcal{D}_k(s_k)$. The process repeats until all nodes have been fully examined.

At this stage, we can uniquely construct all edges a_k^i in the graph using the trajectories stored during the search process. The developed procedure for the discovery of new nodes and edges is summarized in Algorithm 1.

3.4. Highest level: Reinforcement learning

The algorithm at the highest level learns the optimal sequence of movements from the start to the target node. Nodes and edges in the graph-based representation of the task are closely related to the states and actions in RL algorithms (see Section 3.3.1). In RL, an action causes a transition from one state to another. In our graph-based task representation, this corresponds to following the CDMP associated with the chosen edge from the current node to the next node.

² In the maze exploration literature, this type of search procedure is referred to as the left-hand and right-hand rule.

³ As the robot is compliant, we can apply a suitable wrench at the robot's tip so that the robot attempts to move in the given direction. If the applied motor command results in motion, we can continue the motion using the approach described in Section 3.2. Note that due to robot compliance, the direction of the applied force or torque does not need to be perfectly aligned with the direction of the free motion.

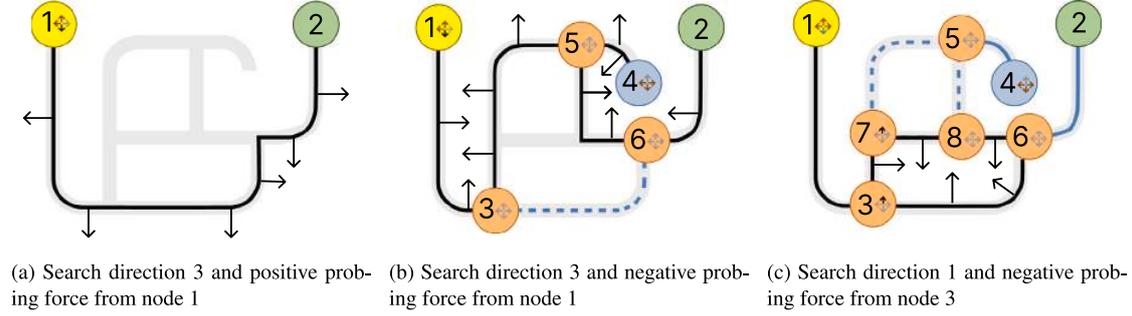


Fig. 5. Key steps in the search process to explore a planar maze. (a) The robot starts in node 1 and selects a direction (down) after trying two other directions before. It applies a positive force and continues until it reaches the target node (marked green). The route is marked with a black line. (b) The robot examines the same branch again with a negative force. Doing so the robot must also turn back and discovers node 4. As the path diverges from the previous (blue dashed line), the algorithm discovers new nodes 3, 5 and 6. The edges between nodes 1 and 3, 4 and 5 and 6 and 2 have been explored with both probing forces and do not need to be examined anymore. (d) The robot continues with the closest node with unexamined branches. It goes up in node 3 and applies a positive force, which leads to the discovery of further nodes and edges. The arrow emblems indicate continuation directions (gray for unexamined, black for selected, blue for examined, and red for impossible).

Algorithm 1: Graph topology determination algorithm for $\kappa = 0$. The algorithm identifies all nodes and edges in the graph by making sure that at every node, the robot searches in all possible continuation directions with all probing wrenches. The algorithm terminates when all nodes are fully examined.

Input: Initial robot pose, probing wrenches $\{f_s^T, m_s^T\}^T$, search directions set $\{d_i\}_{i=0}^n$
Output: Graph representation of the problem

- 1 initialize the graph with start node s_1 at the initial robot pose, $K = 1$
- 2 assign the possible search directions $D(s_1) = \{d_i\}_{i=0}^n$
- 3 **while** $\exists k \leq K$ so that $d_i \in D(s_k)$ and not all probing wrenches have been applied yet in direction d_i **do**
- 4 select node s_k closest to the start node that fulfills the above condition
- 5 select a not yet fully examined direction $d_i \in D(s_k)$
- 6 select probing wrench $\{f_s^T, m_s^T\}^T$ that has not yet been applied while examining the direction d_i
- 7 **if** motion in the selected direction possible **then**
- 8 **while** not in target node and not in loop **do**
- 9 apply probing wrench $\{f_s^T, m_s^T\}^T$ while following the environmental constraints (see Section 3.2)
- 10 record trajectory
- 11 **if** not reached existing node **then**
- 12 **if** motion cannot be continued in any other way but turning back **then**
- 13 $K = K + 1$, create node s_k at the current robot pose, $D(s_k) = \{d_i\}_{i=0}^n$
- 14 **if** motion diverged from a path explored in one of the previous iterations or rejoins an existing path **then**
- 15 $K = K + 1$, create node s_k at the current robot pose, $D(s_k) = \{d_i\}_{i=0}^n$
- 16 mark $d_i \in D(s_k)$ as examined with the selected probing wrench
- 17 **else**
- 18 remove d_i from $D(s_k)$
- 19 identify the graph edges associated with search directions $d_i \in D(s_k)$ by analyzing the stored search trajectories
- 20 encode trajectories associated with the identified edges as CDMP d_k^c

In our approach, the selection of the next action to be explored is guided by ϵ -greedy strategy, which is defined as follows

$$\pi(s) = \begin{cases} \operatorname{argmax}_a Q(s, a), & \text{with probability } 1 - \epsilon \\ \text{random action}, & \text{with probability } \epsilon, \end{cases} \quad (12)$$

where parameter ϵ is the ratio between the exploration and exploitation [41].

Since finding optimal paths in a graph is a finite horizon discrete problem [42], we can apply any classical RL algorithm. In our experiments, we used off-policy Q-learning [41]. We assign a positive reward when the robot reaches the goal state. No intermediate rewards are assigned before reaching the goal state. In every state s_k , the action-value function $Q(s_k, a_k)$ is updated according to the Q-learning algorithm

$$Q(s_k, a_k) \leftarrow Q(s_k, a_k) + \alpha(r_k + \gamma \max_{a_{k+1}} \{Q(s_{k+1}, a_{k+1})\} - Q(s_k, a_k)), \quad (13)$$

where s_k is the k -th state, a_k denotes the action taken in s_k , r_k is the reward obtained in state s_k , $0 < \alpha < 1$ is the learning gain, and $0 < \gamma < 1$ is the discount factor, which gives recent rewards higher importance.

The algorithm at the highest level returns a walk from the start node to the end node, which defines the learned sequence of CDMPs.

The learned CDMPs can be further improved with various robot policy refinement methods such as ILC [10].

4. Experimental evaluation

This section experimentally verifies the proposed hierarchical learning of contact policies. All experiments were performed with a seven-degrees-of-freedom collaborative robot Franka Emika Panda. For this purpose, we implemented the CSF controller using the libfranka library and the ros_control framework in C++. The highest hierarchical level with the Q-learning RL algorithm and the middle level with the graph topology detection algorithm were implemented in Matlab and communicated with the CSF controller using ROS at 100 Hz. The Q learning and CSF controller parameters were the same for all experiments. They were set to $\alpha = 0.9$, $\gamma = 0.92$ and $\epsilon = 0.8$. The delay factor w used in the filtering was chosen by trial and error to be $w = 5$. The reward assignment was also the same in all cases. When the robot reached the desired state, we assigned a reward of $r = 20/k$, where k is the number of steps in Eq. (13) to reach the desired state. The gain of the CSF controller was set to 2000 N/m and 60 N/rd in the tangential component and to zero in the normal and binormal of the FS frame. The probing force and torque magnitudes f_0 and m_0 were set to 7 N and 0.1 N m, respectively. An anthropomorphic qb SoftHand [43] was used in the door opening and gear shifting experiments. In the remaining experiments, the original Panda's two-finger gripper was used.

4.1. Maze learning

In the first experiment, we applied our approach to the maze learning problem (see Fig. 6). We already studied maze learning in our

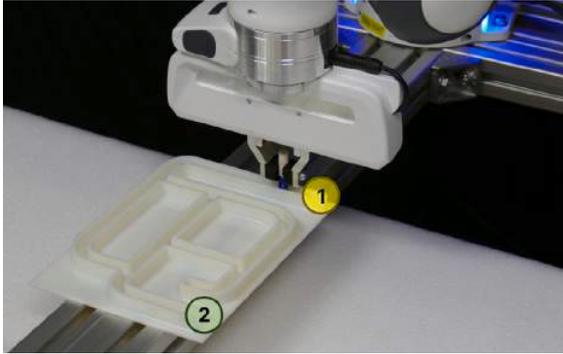


Fig. 6. Panda robot during the maze learning process. The start and the goal states are shown in yellow and green, respectively.

previous research [7], where we demonstrated the greatly improved efficiency of hierarchical learning compared to a single-level RL scheme. In this research, we additionally improved the efficiency of learning, because the proposed approach does not need to stop in small steps to check for the existence of new nodes (see Section 3.3). To initiate the graph search, we manually guided the robot to the start node, denoted with 1 in Fig. 6. This was the starting point in all subsequent search iterations. The normal direction of the modified FS frame was aligned with the global z axis, which is orthogonal to the maze. As the robot motion during maze exploration is planar, the tangent to the robot motion is guaranteed to lie in the plane defined by the maze. The third orthogonal axis (binormal) can thus be easily computed as the vector product of the tangent and the z axis.

Once all branches are investigated, the algorithm cannot find new nodes. The key stages of the search procedure are explained in Fig. 5. The algorithm at the highest level has all the necessary information and can continue learning without the robot having to perform the movements in the maze. In the case of our maze, the CSF controller at the lowest and the search algorithm at the middle level needed three iterations to find all eight nodes, and two more to ensure no further nodes exist. The Q-learning needed five or fewer episodes to discover an optimal sequence of nodes on average. The convergence of the Q-learning is shown in Fig. 7 left.

The selection of the algorithm for discovering new nodes does not affect the learned policy. However, it affects how fast the robot moves along the corridor during learning. The total learning time with the proposed search strategy was approx. ten times shorter than in our previous experiment [7]. This difference is not due to the greater or lesser efficiency of the individual algorithm but rather due to the robot movements generated by the individual algorithm. In the previous algorithm, the robot advances in small steps as it searches for path forks. In contrast, the new algorithm generates continuous paths where branch points are sought by applying a force perpendicular to the robot's direction of motion. The video of the experiment is accessible in supplementary materials as [Maze Learning video](#).

We repeated the learning algorithm for a more complex case where the maze exit is locked. To unlock it, the robot must first visit state four. The Q-learning convergence for this example is shown in Fig. 7 right. Note also that an algorithm for finding the shortest path in graphs at the highest hierarchical level could not solve this problem. We considered this problem as many mechanisms require a specific sequence of states to be visited before reaching the desired state. An example from everyday life is a retractable ballpoint pen. Another such example, door opening, where the doors are equipped with a multi-point locking mechanism, is discussed next.

4.2. Door opening learning

In order to demonstrate the limitations of traditional graph algorithms for searching the shortest path, an experiment was conducted in which a door opening task was performed, as depicted in Fig. 8 left. The results of this experiment reveal why not all nodes can be discovered by these methods, as new nodes might become reachable only after some others have been visited. The door was equipped with mechanical locking for lever-handle-operated doors. To open the locked door, it was first necessary to turn the hook up to unlock it before proceeding with the ordinary door-opening procedure (see Fig. 8). The robot's task was to learn the policy for door opening autonomously. The door hook pose was defined with kinesthetic teaching.

After the robot grabbed the hook, it generated forces and torques in random directions. When a randomly selected force caused a motion, it continued in that direction as described in the 3.2 section. This way, it tested all possible combinations of actions and learned the optimal sequence of movements to open the door, regardless of whether it was unlocked or locked. Moreover, it autonomously learned a general policy for opening unlocked and locked doors in approx. seven episodes. Due to the limited workspace, the robot could not fully open the door. Therefore, we assigned a positive reward when the door opening angle exceeded 20 deg. The exploration algorithm autonomously discovered the graph which describes the above learning process, presented in Fig. 8. Since learning to open a door with multi-point locking is more complex than a standard locking system and requires knowledge of previous actions to fulfill the MDP property, the graph states were determined with $\kappa = 1$ in Eq. (10). The learned policy starts in state 1 and continues visiting states denoted by 2 and 3, the final state describing the door open state. If the door is locked, we set $\kappa = 1$ as explained in Section 3.3.1. The robot can initially not proceed to state 6. Instead, it moves to states 3, 4, 5, 6, and 7 to unlock and open the door. The algorithm finds all possible actions autonomously and encodes them as DMPs in the set $\mathcal{A}(s_i)$. State 3 deserves attention since the system applying action 1 remains in the same position B. This way, the algorithm recognizes that the door is locked. Note again that the state labeling depends on the initial robot motion and the initial state of the door. However, state labeling does not affect the learned policy. Fig. 8 right shows the case where the robot started to turn the hook down on the initially unknown state (locked/unlocked) of the doors. The convergence of the RL algorithm at the top of the proposed hierarchical scheme, which learns the appropriate action sequences for the door opening, is shown in Fig. 9. In this experiment, the Q-learning algorithm (13) was enhanced with eligibility traces [41]. On average, the algorithm took six episodes to learn the policy.

Previous research has often discussed learning to open doors, allowing us to compare the effectiveness of these approaches. First, we mention that none of the studies addressed the mechanical door-locking mechanism and limited themselves to the problem of learning to open a standard door. With our graph-based approach, we can immediately notice that no decision is required when opening a standard door. Consequently, the CSF controller can discover the required sequence of moves in a single trial. In contrast, standard door opening learning by PI^2 algorithm required more than 300 trials to learn the policy [44]. In [45], authors investigate accelerated learning of door opening with two robot agents. They learned the final policy across 20 consecutive trials while learning with one agent was much slower. The approach proposed in [6], which combines PI^2 learning and compliant controller, required 9 roll-outs on average. Another approach [46] proposed policy learning in the simulated environment and application of learned policy in a real environment. Learning in a simulated environment took approx 300 trials. Although compared approaches also perused additional objectives, such as end-to-end learning, the comparison demonstrates the efficiency of the proposed framework.

Learning to open doors and execution of the learned policy is shown in the [Door Opening video](#) in the supplementary materials.

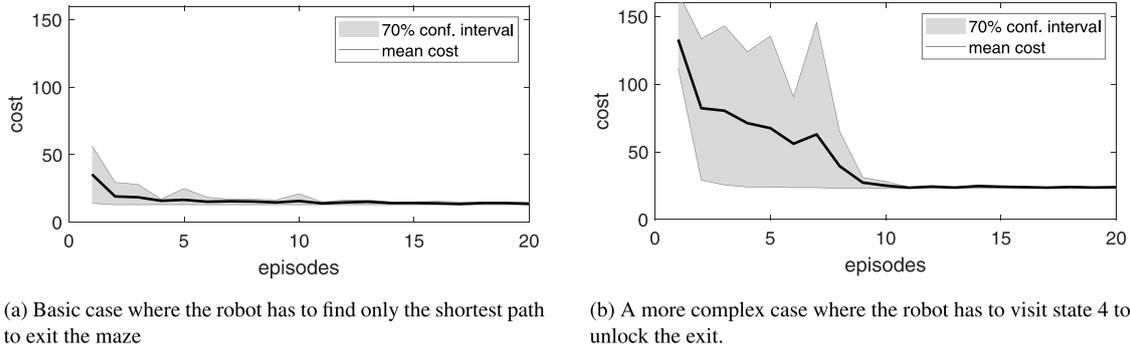


Fig. 7. Convergence of the RL for maze learning shows learning cost vs. episodes. Learning cost is the path length for escaping the maze. The shaded region denotes a 70% confidence interval of 20 epochs of learning.

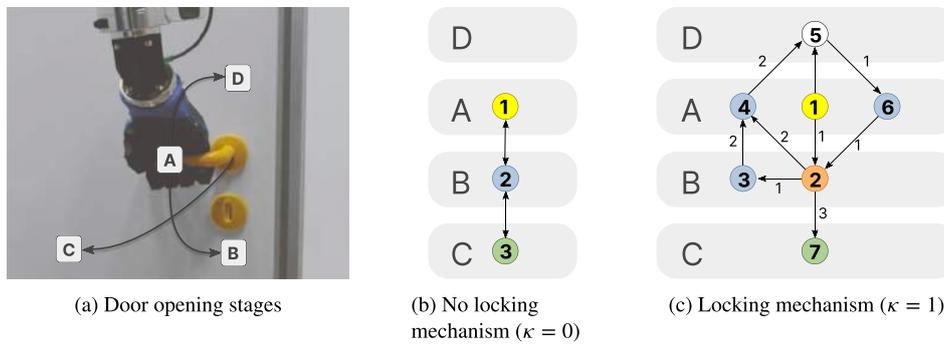


Fig. 8. Different stages of door opening are shown in scheme (a). If the door is locked, it is first necessary to unlock the mechanism by going to position D before proceeding with the ordinary opening procedure (going from A through B in position C). The graph in subfigure (b) corresponds to the situation when the locking mechanism is disabled. The graph shows that no learning is required at the top level of the hierarchy in this case. The graph in subfigure (c) shows the resulting graph from an episode of the learning procedure when the door is initially locked. Note that multiple nodes are created as the nodes also contain information about the previously visited nodes. The action labels describe the semantic meaning of the action: 1 - pushing down, 2 - pushing up, and 3 - pulling the hook.

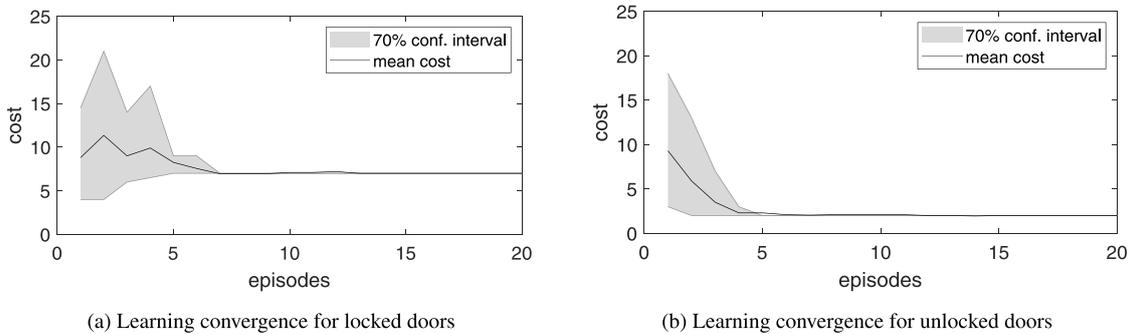


Fig. 9. Convergence of the RL for door opening learning shows learning cost vs. episodes. The cost is the number of states the robot visits before opening the door. Due to the different costs for opening unlocked and locked doors, the learning convergence is shown in two plots for unlocked and locked doors, respectively. Note also that the aim is to learn the general policy regardless of whether the doors are locked or unlocked. However, if we knew the door's state in advance, the cost of opening the locked door would be 4. During the learning, the algorithm occasionally finds this policy but rejects it as it does not fit the general case of the unknown door's state.

4.3. Learning to shift car gears

In this experiment, the robot autonomously learns to shift manual car gear transmissions (See Fig. 10). The goal was to learn how to shift from the neutral position to gears 1, 2, 3, 4, and 5 and from the neutral position to the reverse gear. First, we show the robot how to grip the gear lever with kinesthetic guidance. As the initial pose for learning, we chose third gear, but any other position could be chosen. The robot autonomously tests all possible ways to move the gear lever

in all Cartesian axes. We instructed the robot not to test the gear lever orientations. Therefore, the robot was all the time compliant in all orientational d.o.f. The semantic mapping between the robot pose and the gear numbers is given in advance. The goal was to learn the policies of how to shift into any gear from any position.

Learning began with exploiting additional force in the positive normal axis of the FS frame. The resulting robot trajectory is denoted with red in Fig. 10(d) Given that choice, the search algorithm labeled states as shown in Fig. 10(b). Note again that a new node is added

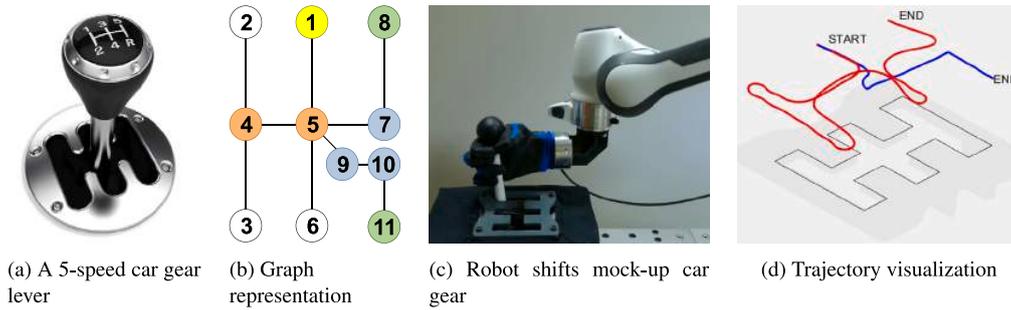


Fig. 10. (a) Standard gear lever. (b) Graph with states during the learning to shift manual transmission, as discovered and labeled by the search algorithm. Circles in two different colors represent states that can be considered target states, depending on the task requirements. (c) Robot shifts a mockup manual transmission. (d) The trajectory discovered with additional force in the normal direction and search force in the binormal direction of the FS frame are denoted with red and blue, respectively.

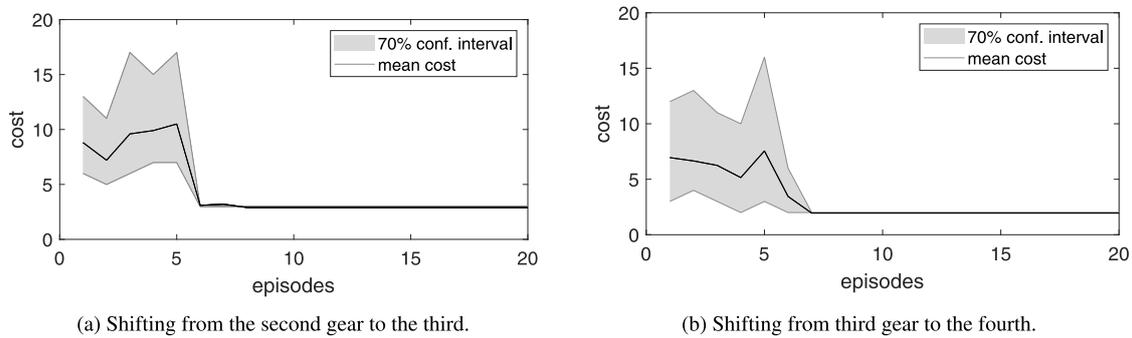


Fig. 11. Convergence of the RL for gear shifting learning shows learning cost (i.e. number of states visited) vs. episodes for two shifting sequences.

whenever the robot has to search in a completely new direction and whenever there are multiple ways to continue the motion. We set the fifth gear as a target node. Search with additional force in a negative normal direction did not discover any new nodes. Next, we repeated the search with an additional force in the negative binormal direction. The binormal axis of the FS frame in this experiment almost always coincides with the global z coordinate. The robot found new nodes this time and ended in the reverse gear. The resulting robot trajectory is denoted with blue in Fig. 10(d). After the robot discovered all possible nodes, it also learned how to shift from the neutral to the first, second, fourth, and fifth gear. The exception is shifting from the second to the third gear and from the third to the fourth gear because the robot did not encounter this combination while discovering new nodes. Q learning at the top of the hierarchical scheme was assigned to discover two missing aforementioned policies. The algorithm learns them in 6 to 7 cycles, as shown in Fig. 11.

A demonstration can be found in the [Gear Shift video](#) in the supplementary materials.

4.4. Car licence plate light disassembly and assembly

With this experiment, we intended to show how to learn the assembly/disassembly of an object which consists of multiple parts, such as a car license plate light. It consists of a base part with the bulb case, bayonet bulb, and transparent cover. The base of the light is firmly attached to the base plate, as shown in Fig. 12 left and center. Again, we start the disassembly process with a manual guide to the suitably chosen pose, where the robot gripper can firmly grasp the cover of the light. The cover is attached to the base with two side pins. To release them, it is necessary to apply a force in the z direction and a torque around the y axis. The CSF controller finds the appropriate direction by exploration as described in Section 3.2. The corresponding graph is

trivial in this case. For the bayonet bulb disassembly, the actions are more complex, as it is necessary to push the bulb down in z axis, rotate it around z axis, and pull it in z axis. At this stage, it is also possible to push the bulb in the z axis, but in this case, the robot arrives at a node of degree 0 and has to turn back. The corresponding graph is shown in Fig. 12 right. Note that this graph differs from the others presented in this work as it also includes actions demonstrated by a human. State 1 describes the manual guidance of the robot to the light cover. State 4, on the other hand, describes the actions where the human operator guides the robot to the place where he releases the cover and, after that, to the place where he grabs the light bulb. The robot could also perform these actions autonomously using robotic vision. However, this was not the subject of our research. The role of RL at the highest hierarchical level, in this case, is minor since the graph has a single decision state, and the learning algorithm finds the final solution in two steps in the worst-case scenario.

During disassembly, the applied forces and torques inherently align the bulb to slide along the casing, as the center of compliance is in the robot gripper. During assembly, however, the situation is reversed, and it is necessary to actively control the bulb orientation to align it with the casing. In robotics, this is the well-known PiH problem, where it is necessary to obtain the remote center of compliance using an appropriate force control strategy [47] or apply additional learning [19]. This is also one of the reasons why the disassembly is easier than the corresponding assembly. This makes it easier to learn the assembly semantics through disassembly easier, as already suggested in [7]. The robot generates the corresponding force and torque with an impedance controller by applying a displacement in z coordinate and a rotation around the z coordinate. This experiment is also demonstrated with the corresponding [Licence Plate video](#) in the supplementary materials.

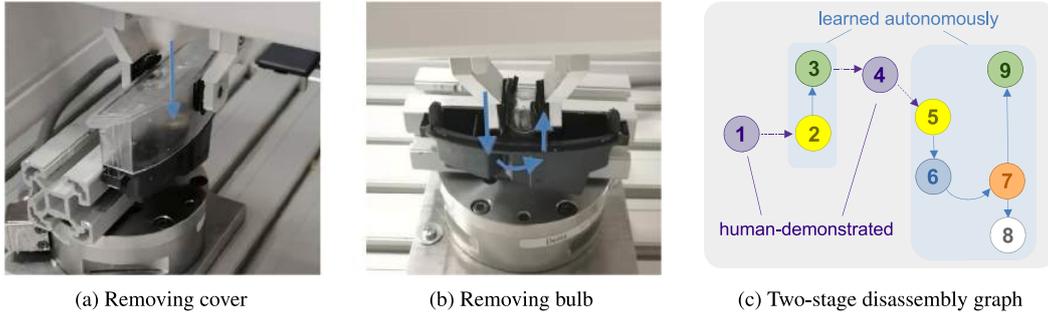


Fig. 12. Panda robot during the license car plate light disassembly and assembly. (a) removal of the transparent cover. (b) disassembly of the bayonet bulb. (c) disassembly graph. The states in violet color represent human demonstration.

5. Conclusion

Autonomous learning of robotic tasks in close contact with the environment is one of the still open challenges in modern robotics. This paper presents a novel approach based on a task representation with directed graphs. Based on this formulation, we propose a three-level hierarchical learning scheme to solve the learning problem. The highest hierarchical level makes decisions based on exploratory movements generated by the newly developed CSF controller at the lowest level. The distinguishing feature of the CSF controller is that it allows the specification of variable compliance along the robot motion. The middle level is dedicated to the discovery of new nodes and edges in the task graph. The main advantage of the hierarchical scheme is accelerated learning, which requires only a few learning roll-outs for typical tasks encountered in everyday life and industrial plants. Another advantage is that it generates continuous-time control policies using classical discrete-time RL methods such as Q-learning or SARSA.

We also considered the possibility of using alternative algorithms to find the optimal sequence of transitions at the highest hierarchical level of the proposed framework. Some problems can be successfully solved by finding the shortest paths between the nodes in a graph. These methods are generally faster than RL algorithms but less general. For example, learning to assemble a car license plate light, gear shifting, and the first maze learning problem could also be solved using graph exploration methods (breadth-first search or depth-first search) and Dijkstra's algorithm [48]. On the other hand, graph searching with RL can solve more complex problems where finding the shortest paths in the graph fails. Two of them were considered in our research, i.e., opening the door equipped with a lever-handle-locking mechanism and the second example of maze learning. A truly autonomous robot should be able to solve various problems regardless of their complexity. Thus RL is a more reasonable choice for finding the optimal path through the graph in our hierarchical scheme.

Our framework for autonomous learning of contact policies was verified in four experiments. In the first experiment, we chose the well-known maze learning problem because it nicely illustrates the essence of our approach based on directed graphs and algorithms for finding paths through such graphs. Next, we performed two experiments from everyday life: opening the door and shifting the car gearbox. Finally, we provide an example of learning assembly tasks by autonomous disassembly.

Our experiments focused on tasks where movement is allowed in a single, constantly changing direction. There are many such tasks in our daily lives and in industrial production processes. In addition to the discussed and numerous other assembly and disassembly tasks, there are tasks such as screwing, connecting BNC connectors, handling valves

and levers, opening drawers and cabinets, etc. The proposed approach is directly applicable to such tasks. Moreover, it can be extended to tasks that allow movement in several degrees of freedom, which will be the subject of our future research.

CRedit authorship contribution statement

Mihael Simonič: Methodology, Software, Visualization, Writing. **Aleš Ude:** Conceptualization, Methodology, Writing. **Bojan Nemec:** Conceptualization, Methodology, Software, Supervision, Visualization, Writing.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgments

The research leading to these results has received funding from the EU Horizon 2020 Research and Innovation Action ReconCycle, GA no. 871352, and program group P2-0076 Automation, Robotics, and Biocybernetics supported by the Slovenian Research and Innovation Agency.

Appendix A

Lemma 1. *Let the robot be controlled using impedance control law defined by Eqs. (5)–(8), with diagonal stiffness and damping matrices specified in Frenet–Serret (FS) coordinate frames (1) distributed along the robot path. If we choose equal stiffness and damping constants in the normal and binormal direction of motion, then the commanded torque (5) and thus the robot motion is independent of the direction of the normal and binormal vector that defines the FS frame.*

Proof. As a first step, let us recall that the control accelerations are computed according to Eq. (7). Assuming that stiffness and damping in the directions orthogonal to the tangential direction are equal, i.e., $k_{yz} = k_y = k_z$, $d_{yz} = d_y = d_z$, we have

$$\mathbf{K}_p = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_{yz} & 0 \\ 0 & 0 & k_{yz} \end{bmatrix}, \quad \mathbf{D}_p = \begin{bmatrix} d_x & 0 & 0 \\ 0 & d_{yz} & 0 \\ 0 & 0 & d_{yz} \end{bmatrix}. \quad (14)$$

Next we rewrite Eq. (7) with the above stiffness and damping matrices

$$\begin{aligned} \ddot{\mathbf{p}}_c &= \begin{bmatrix} \mathbf{t} & \mathbf{n} & \mathbf{b} \end{bmatrix} \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_{yz} & 0 \\ 0 & 0 & k_{yz} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{t}}^T \\ \dot{\mathbf{n}}^T \\ \dot{\mathbf{b}}^T \end{bmatrix} \mathbf{e}_p \\ &+ \begin{bmatrix} \mathbf{t} & \mathbf{n} & \mathbf{b} \end{bmatrix} \begin{bmatrix} d_x & 0 & 0 \\ 0 & d_{yz} & 0 \\ 0 & 0 & d_{yz} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{t}}^T \\ \dot{\mathbf{n}}^T \\ \dot{\mathbf{b}}^T \end{bmatrix} \dot{\mathbf{p}} \\ &= (k_x \mathbf{t}\mathbf{t}^T + k_{yz}(\mathbf{n}\mathbf{n}^T + \mathbf{b}\mathbf{b}^T))\mathbf{e}_p + (d_x \mathbf{t}\mathbf{t}^T + d_{yz}(\mathbf{n}\mathbf{n}^T + \mathbf{b}\mathbf{b}^T))\dot{\mathbf{p}}. \end{aligned} \quad (15)$$

Let us now define another coordinate frame $\mathbf{R}'_p = [\mathbf{t}' \ \mathbf{n}' \ \mathbf{b}']$, with the first column defined by the tangent of robot motion, i.e., $\mathbf{t}' = \mathbf{t}$, but with the other two orthogonal axes \mathbf{n}' and \mathbf{b}' chosen arbitrarily. If the diagonal stiffness and damping matrices (14) are defined in this frame and with equal stiffness and damping in the direction of \mathbf{n}' and \mathbf{b}' , then the corresponding control acceleration (7) in the robot base coordinate frame is given by

$$\ddot{\mathbf{p}}'_c = (k_x \mathbf{t}\mathbf{t}^T + k_{xy}(\mathbf{n}'\mathbf{n}'^T + \mathbf{b}'\mathbf{b}'^T))\mathbf{e}_p + (d_x \mathbf{t}\mathbf{t}^T + d_{yz}(\mathbf{n}'\mathbf{n}'^T + \mathbf{b}'\mathbf{b}'^T))\dot{\mathbf{p}}. \quad (16)$$

For any rotational matrix \mathbf{R} it holds $\mathbf{R}\mathbf{R}^T = \mathbf{I}$, hence

$$\mathbf{I} = \mathbf{t}\mathbf{t}^T + \mathbf{n}\mathbf{n}^T + \mathbf{b}\mathbf{b}^T = \mathbf{t}'\mathbf{t}'^T + \mathbf{n}'\mathbf{n}'^T + \mathbf{b}'\mathbf{b}'^T. \quad (17)$$

As the tangential components are equal in the above equation, it follows that $\mathbf{n}\mathbf{n}^T + \mathbf{b}\mathbf{b}^T = \mathbf{n}'\mathbf{n}'^T + \mathbf{b}'\mathbf{b}'^T$, hence $\ddot{\mathbf{p}}_c = \ddot{\mathbf{p}}'_c$. In the same way, we can prove that also for the commanded angular velocities (8) the relation $\dot{\boldsymbol{\omega}}_c = \dot{\boldsymbol{\omega}}'_c$ holds. Thus under the assumptions of the lemma, the commanded torque (5) is independent of the direction of normal and binormal vectors that defines the FS frame. \square

Appendix B. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.rcim.2023.102657>.

References

- [1] A. Hussein, M.M. Gaber, E. Elyan, C. Jayne, Imitation learning: A survey of learning methods, *ACM Comput. Surv.* 50 (2) (2017).
- [2] J. Kober, J. Peters, Reinforcement learning in robotics: A survey, in: *Learning Motor Skills*, in: Springer Tracts in Advanced Robotics, vol. 97, Springer, Cham, 2014, pp. 9–67.
- [3] D. Bristow, M. Tharayil, A. Alleyne, A survey of iterative learning control, *IEEE Control Syst. Mag.* 26 (3) (2006) 96–114.
- [4] T. Gašpar, M. Deniša, P. Radanovič, B. Ridge, T.R. Savarimuthu, A. Kramberger, M. Priggemeyer, J. Roßmann, F. Wörgötter, T. Ivanovska, S. Parizi, Ž. Gosar, I. Kovač, A. Ude, Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells, *Robot. Comput.-Integr. Manuf.* 66 (2020) 101979.
- [5] Z. Liu, Q. Liu, W. Xu, L. Wang, Z. Zhou, Robot learning towards smart robotic manufacturing: A review, *Robot. Comput.-Integr. Manuf.* 77 (2022) 102360.
- [6] B. Nemeč, L. Žlajpah, A. Ude, Door opening by joining reinforcement learning and intelligent control, in: 18th International Conference on Advanced Robotics (ICAR), Hong Kong, 2017, pp. 222–228.
- [7] M. Simonič, L. Žlajpah, A. Ude, B. Nemeč, Autonomous learning of assembly tasks from the corresponding disassembly tasks, in: IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), Toronto, Canada, 2019, pp. 230–236.
- [8] M. Suomalainen, Y. Karayiannis, V. Kyrki, A survey of robot manipulation in contact, *Robot. Auton. Syst.* 156 (2022) 104224.
- [9] I. Elguea-Aguinaco, A. Serrano-Muñoz, D. Chrysostomou, I. Inziarte-Hidalgo, S. Bøgh, N. Arana-Areola, A review on reinforcement learning for contact-rich robotic manipulation tasks, *Robot. Comput.-Integr. Manuf.* 81 (2023) 102517.
- [10] F.J. Abu-Dakka, B. Nemeč, J.A. Jørgensen, T.R. Savarimuthu, N. Krüger, A. Ude, Adaptation of manipulation skills in physical contact with the environment to reference force profiles, *Auton. Robots* 39 (2) (2015) 199–217.
- [11] B. Nemeč, T. Petrič, A. Ude, Force adaptation with recursive regression iterative learning controller, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 2015, pp. 2835–2841.
- [12] M. Kalakrishnan, L. Righetti, P. Pastor, S. Schaal, Learning force control policies for compliant manipulation, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, 2011, pp. 4639–4644.
- [13] R. Martín-Martín, M.A. Lee, R. Gardner, S. Savarese, J. Bohg, A. Garg, Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macao, China, 2019, pp. 1010–1017.
- [14] M. Hazara, V. Kyrki, Reinforcement learning for improving limited in-contact skills, in: IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, Mexico, 2016, pp. 194–201.
- [15] Y. Chebotar, M. Kalakrishnan, A. Yahya, A. Li, S. Schaal, S. Levine, Path integral guided policy search, in: IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 3381–3388.
- [16] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, R. Tachibana, Deep reinforcement learning for high precision assembly tasks, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada, 2017, pp. 819–825.
- [17] Z. Hou, J. Fei, Y. Deng, J. Xu, Data-efficient hierarchical reinforcement learning for robotic assembly control applications, *IEEE Trans. Ind. Electron.* 68 (11) (2021) 11565–11575.
- [18] Y.-L. Kim, K.-H. Ahn, J.-B. Song, Reinforcement learning based on movement primitives for contact tasks, *Robot. Comput.-Integr. Manuf.* 62 (2020) 101863.
- [19] L. Johannsmeier, M. Gerchow, S. Haddadin, A framework for robot manipulation: Skill formalism, meta learning and adaptive control, in: 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, 2019.
- [20] S. Nair, M. Babaizadeh, C. Finn, S. Levine, V. Kumar, TRASS: Time reversal as self-supervision, in: IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 115–121.
- [21] K. Zakka, A. Zeng, J. Lee, S. Song, Form2Fit: Learning shape priors for generalizable assembly from disassembly, in: IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 9404–9410.
- [22] M.T. Mason, Compliance and force control for computer controlled manipulators, *IEEE Trans. Syst. Man Cybern.* 11 (6) (1981) 418–432.
- [23] A.J.D. Lambert, Disassembly sequencing: A survey, *Int. J. Prod. Res.* 41 (16) (2003) 3721–3759.
- [24] J. Aleotti, S. Caselli, Physics-based virtual reality for task learning and intelligent disassembly planning, *Virtual Real.* 15 (1) (2011) 41–54.
- [25] E. Tuncel, A. Zeid, S. Kamarthi, Solving large scale disassembly line balancing problem with uncertainty using reinforcement learning, *J. Intell. Manuf.* 25 (4) (2014) 647–659.
- [26] K. Erciyes, *Distributed Graph Algorithms for Computer Networks*, Springer, London, 2013.
- [27] H. Bruyninckx, J. De Schutter, Where does the task frame go? in: Y. Shirai, S. Hirose (Eds.), *Robotics Research*, Springer London, London, 1998, pp. 55–65.
- [28] R. Ravani, A. Meghdari, Velocity distribution profile for robot arm motion using rational frenet-serret curves, *Informatica* 17 (1) (2006) 69–84.
- [29] B. Nemeč, N. Likar, A. Gams, A. Ude, Human robot cooperation with compliance adaptation along the motion trajectory, *Auton. Robots* 42 (5) (2018) 1023–1035.
- [30] M. Pilté, S. Bonnabel, F. Barbaresco, Tracking the frenet-serret frame associated to a highly maneuvering target in 3D, in: IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, Australia, 2017, pp. 1969–1974.
- [31] D. Carroll, E. Köse, I. Sterling, Improving frenet's frame using bishop's frame, *J. Math. Res.* 5 (4) (2013) 97–106.
- [32] M. Vochten, T. De Laet, J. De Schutter, Robust optimization-based calculation of invariant trajectory representations for point and rigid-body motion, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 5598–5605.
- [33] G. Niemeyer, J.-J.E. Slotine, A simple strategy for opening an unknown door, in: IEEE International Conference on Robotics and Automation (ICRA), Albuquerque, NM, 1997, pp. 1448–1453.
- [34] J.-O. Lachaud, A. Vialard, F. de Vieilleville, Analysis and comparative evaluation of discrete tangent estimators, in: Proceedings of the 12th International Conference on Discrete Geometry for Computer Imagery, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 240–251.
- [35] A. Albu-Schaffer, C. Ott, G. Hirzinger, A unified passivity-based control framework for position, torque and impedance control of flexible joint robots, *Int. J. Robot. Res.* 26 (1) (2007) 23–39.
- [36] C. Ott, A. Albu-Schaffer, A. Kugi, S. Stramigioli, G. Hirzinger, A passivity based cartesian impedance controller for flexible joint robots - part I: torque feedback and gravity compensation, in: IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA, 2004, pp. 2659–2665.

- [37] A. Dietrich, A. Abu-Schäffer, G. Hirzinger, On continuous null space projections for torque-based, hierarchical, multi-objective manipulation, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, Saint Paul, Minnesota, 2012, pp. 2978–2985.
- [38] A. Ude, B. Nemeč, T. Petrič, J. Morimoto, Orientation in Cartesian space dynamic movement primitives, in: IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, pp. 2997–3004.
- [39] R. Vuga, B. Nemeč, A. Ude, Speed adaptation for self-improvement of skills learned from user demonstrations, *Robotica* 34 (12) (2016) 2806–2822.
- [40] A.M. Sadik, M.A. Dhali, H.M. Farid, T.U. Rashid, A. Syeed, A comprehensive and comparative study of maze-solving techniques by implementing graph theory, in: International Conference on Artificial Intelligence and Computational Intelligence, 2010, pp. 52–56.
- [41] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, Second Edition, The MIT Press, Cambridge, MA, 2015.
- [42] S. Russell, P. Norvig, Artificial Intelligence, fourth ed., Pearson, Upper Saddle River, NJ, 2020, p. 648.
- [43] C.D. Santina, G. Grioli, M.G. Catalano, A. Brando, A. Bicchi, Dexterity augmentation on a synergistic hand: The pisa/IIT SoftHand+, in: IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, Korea, 2015, pp. 497–503.
- [44] E. Theodorou, F. Stulp, J. Buchli, S. Schaal, An iterative path integral stochastic optimal control approach for learning robotic tasks, *IFAC Proc. Vol. 44 (1)* (2011) 11594–11601.
- [45] S. Gu, E. Holly, T. Lillicrap, S. Levine, Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 3389–3396.
- [46] Y. Wang, L. Wang, Y. Zhao, Research on door opening operation of mobile robotic arm based on reinforcement learning, *Appl. Sci.* 12 (10) (2022).
- [47] J. Jiang, Z. Huang, Z. Bi, X. Ma, G. Yu, State-of-the-art control strategies for robotic PiH assembly, *Robot. Comput.-Integr. Manuf.* 65 (2020) 1–26.
- [48] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, second ed., The MIT Press, Cambridge, MA, 2001.



Mihael Simonič received B.Sc. and M.Sc. degrees in cognitive science from the University of Tübingen, Germany and the doctoral degree from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia in 2023. He works at the Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia. His current research focuses on the intersection between robot learning and human–robot collaboration.



Aleš Ude received the diploma degree in applied mathematics from the University of Ljubljana, Ljubljana, Slovenia, and the Dr.Eng. degree from the University of Karlsruhe, Karlsruhe, Germany, in 1990 and 1995, respectively. He is currently the Head of the Department of Automatics, Biocybernetics, and Robotics, Jožef Stefan Institute, and a professor at the Faculty of Electrical Engineering, Ljubljana. He is also associated with the ATR Computational Neuroscience Laboratory, Kyoto, Japan. His research interests include robot learning, programming by demonstration, reconfigurable robotic systems, and humanoid robotics.



Bojan Nemeč received B.Sc and M.Sc. degrees in electrical engineering and Ph.D. degree in robotics from the University of Ljubljana, Slovenia. He is currently the head of the Humanoid and Cognitive Robotics Lab and Research Councillor with the Department of Automatics, Biocybernetics, and Robotics, Jožef Stefan Institute, and a professor at Jožef Stefan International Postgraduate School, Ljubljana. His research interests include robot control, robot learning, service robotics, and sports biomechanics.

A.4 “Robust Execution of Assembly Policies Using Pose Invariant Task Representation”

This is the post-print version (author accepted manuscript) of the following publication: B. Nemeč, M. M. Hrovat, M. Simonič, S. Shetty, S. Calinon, and A. Ude. “Robust Execution of Assembly Policies Using Pose Invariant Task Representation”. In: *20th International Conference on Ubiquitous Robots (UR)*. 2023, pp. 779–786.

Robust Execution of Assembly Policies Using a Pose Invariant Task Representation

Bojan Nemeč¹, Matevž Majcen Hrovat¹, Mihael Simonič¹, Suhan Shetty², Sylvain Calinon², and Aleš Ude¹

Abstract—This paper discusses the robustness of executing robot tasks in contact with the environment. For example, in assembly, even the slightest error in the initial pose of the assembled object or grasp uncertainties can lead to large contact forces and, consequently, failure of the assembly operation. Force control can help to improve the robustness only to a certain extent. In this work, we propose using the position and orientation invariant task representation to increase the robustness of assembly and other tasks in continuous contact with the environment. We developed a variable compliance controller which constantly adapts the policy to environmental changes, such as positional and rotational displacements and deviations in the geometry of the assembled part. In addition, we combined ergodic control and vision processing to improve the detection of the assembled object’s initial pose. The proposed framework has been experimentally validated in two challenging tasks; The first example is a mock-up of an assembly operation, where the object moves along a rigid wire, and the second is the insertion of a car light bayonet bulb into the housing.

I. INTRODUCTION

Assembly is one of the most common tasks in industrial robotics. However, assembly operations are not necessary only in industrial environments but also in our homes, as many of the daily tasks we perform are assembly tasks. One of the problems in assembly is accurate calibration. Consider, for example, the peg-in-a-hole (PiH) task depicted in Fig 1. Even the smallest errors in the orientation of the assembled object can result in large positional errors and, consequently, large forces during the assembly using a predefined assembly sequence. In automated robot assembly, complex calibration procedures and specialized hardware are usually applied to determine the position and orientation of workpieces. On the contrary, this is often hard to achieve in a domestic environment. The same applies to small series and craft production. Therefore, developing procedures that automatically adapt assembly tasks to environmental changes is one of the main challenges for the faster introduction of robotics in our homes and small-scale production [1].

This problem is not new and has been under investigation since the beginning of robotics. Early approaches tend to solve the problem by force control [2]. However, force control is often slow in adaptation and can become unstable, especially in admittance-based control schemes. Therefore,

force control is often combined with various learning procedures that can effectively eliminate the stability problems but are unsuccessful with stochastic error sources [3], [4], [5]. One of the most successful approaches turned out to be compliant control, which can naturally adapt to small environmental deviations [6].

Alternatively, one can exploit the arising contact forces and torques to create more robust assembly policies. Based on this paradigm, we aim to develop an algorithm that can adapt to larger deviations in object position where compliance and force control become unsuccessful. The basic idea of our approach is that while adapting to the environment, the robot also updates the task according to the current position and orientation. To adapt to the environment in real-time, we use two techniques: firstly, the task formulation using a pose invariant description and, secondly, the execution of the task with a controller, which inherently adapts to the variable environmental constraints. We have developed a new position and orientation invariant trajectories method, distinguished by its compact form and computational robustness. They are specially adapted to describe assembly operations. The proposed approach allows us to avoid lengthy and demanding calibration procedures usually required in robotic assembly.

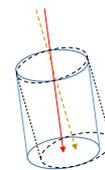


Fig. 1. The solid and the dashed outline show the real and the estimated cylinder, which is subject to peg-in-hole operation. Arrows show the corresponding true and estimated insertion trajectories.

However, the proposed framework requires at least a coarse knowledge of the robot’s initial pose when assembling. To this end, we propose a multimodal approach that combines robotic vision and measured contact forces. The appropriate initial pose is then explored with an ergodic controller.

This paper consists of five sections. In Section II we introduce an incremental policy representation for assembly tasks based on position and orientation invariant descriptors. Next, in Section III, we introduce Adaptive Impedance Controller (AIC) as a basic enabler of our approach. It enables following the desired policy and adaptation to the unknown and variable environmental constraints. The search for the

¹ The authors are with the Humanoid & Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia, email: bojan.nemec@ijs.si, matevz.majcen@ijs.si, mihael.simonic@ijs.si, ales.ude@ijs.si.

² The authors are with the Idiap Research Institute, Martigny, Switzerland, email: suhan.shetty@idiap.ch, sylvain.calinon@idiap.ch.

initial assembly pose, which combines vision processing and ergodic control, is explained in Section IV. Experimental evaluation in Section V considers the validation using a mockup, which comprises all assembly challenges considered in our approach, and, finally, the challenging task of bayonet bulb insertion. A short discussion and conclusions are provided in Section VI.

II. DESCRIBING CONTACT POLICY USING POSE-INVARIANT TRAJECTORY NOTATION

Our approach aims to encode the task incrementally, i.e., at any time, we specify the motion relative to the robot's current pose. The incremental task formulation is provided using a task description where the position and orientation of the resulting motion trajectory depend only on the initial pose of the task. We refer to this formulation as *pose invariant* task definition. This section proposes a novel approach for formulating invariant trajectories applicable to assembly tasks.

Different formulations have been developed for a position and orientation invariant task description [7], [8]. They are mainly used to recognize and understand human intentions and actions in collaborative robotics. Some approaches aim to compute invariant motion trajectories directly from the video stream [9]. Others apply invariant geometrical properties derived from the definition of curvature and torsion [8], [10] and Frenet-Serret frames [11]. However, most approaches based on differential geometrical properties suffer from the limitations that they can only be applied to non-degenerate trajectories. Roughly speaking, these are trajectories in which the curvature is different from zero everywhere. As straight-line motions are common in robot assembly, many assembly trajectories are degenerate in this sense.

Our approach differs from all previously presented ones in that it does not deal with the general 6-dimensional motion of a rigid body. We consider tasks that allow only partial positional and orientation freedom of motion. The border between the region where the robot motion is constrained by the environment and the region where motion is free is called a *C-surface* [12]. The motion is possible along the tangential direction of the *C-surface* and is constrained in orthogonal directions. The dimension of the *C-surface* determines the number of degrees of freedom of robot motion in contact with the environment. A typical assembly task is characterized by an at most two-dimensional *C-surface*. Take, for example, a round peg in a hole problem, where the hole is aligned with the x axis of the robot. Here, only two coordinates are freely definable, x and the rotation around x axis. Orientation often changes with position, such as when screwing or assembling a BNC connector. Tasks can be composed as a sequence of such movements, such as inserting a bayonet bulb, composed of a translation followed by a rotation. In such tasks, a maximum of two degrees of freedom change simultaneously, one translational and one rotational. On the other hand, during the execution of the task, the robot follows environmental constraints. The robot

can follow them more accurately with prior knowledge of these constraints.

For this reason, we have chosen a four-dimensional task formulation. This formulation can describe most assembly operations and operations where the robot is in physical contact with the environment. In our approach, these degrees of freedom are aligned with the x -axis of the local coordinate system, named the *object* coordinate system. This way, one can describe the motion as sliding a ring on a rigid wire, as shown in Fig. 2. This problem comprises all assembly cases considered in our research.

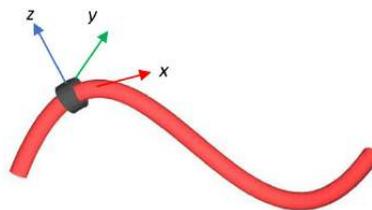


Fig. 2. A movement of a ring on a rigid wire. The coordinate system attached to the ring is referred to as the *object* coordinate system.

Consider that a trajectory of an object (in our case, sliding a ring on a rigid wire) can be represented by a series of trajectory points, which are obtained as a sequence of homogeneous transformation matrices applied to an initial transformation matrix. Each point on the trajectory is characterized as:

$$\mathbf{T}(k) = \mathbf{T}(0)\Delta\mathbf{T}(1)\cdots\Delta\mathbf{T}(k) = \mathbf{T}(0)\prod_{i=1}^k\Delta\mathbf{T}(i), \quad (1)$$

where the homogeneous transformation matrices are calculated as

$$\mathbf{T}(0) = \begin{bmatrix} \mathbf{R}(0) & \begin{bmatrix} x(0) \\ y(0) \\ z(0) \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2)$$

and

$$\begin{aligned} \Delta\mathbf{T}(i) &= \begin{bmatrix} \Delta\mathbf{R}(i) & \begin{bmatrix} \Delta x(i) \\ 0 \\ 0 \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}^T(i-1)\mathbf{R}(i) & \begin{bmatrix} x(i)-x(i-1) \\ 0 \\ 0 \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix} \end{aligned} \quad (3)$$

The trajectory is thus parameterized with the local coordinate system $\mathbf{R}(k)$ and the displacement in the direction of the x axis of $\mathbf{R}(k)$. Variable k denotes discrete time. It is related to the continuous time by $t = k\Delta t$, Δt being the sampling interval. The main advantage of using the above formulation is an incremental computation of the policy, which enables adaptation to environmental changes. The next trajectory

sample depends only on the previous, as can be seen from the relation

$$\mathbf{T}(k) = \mathbf{T}(k-1)\Delta\mathbf{T}(k). \quad (4)$$

This notation, however, is highly redundant, as it requires 10 variables to describe a 4 d.o.f motion. A more compact representation can be achieved by utilizing the following transformations

$$\Delta x(k) = \nu(k)\Delta t \quad (5)$$

$$\Delta\mathbf{R}(k) = \exp(\boldsymbol{\omega}(k)\Delta t) \quad (6)$$

where the exponential mapping ($\exp: \mathbb{R}^3 \mapsto \mathbf{S}^3, \forall \boldsymbol{\omega} \in \mathbb{R}^3, \|\boldsymbol{\omega}\| < 2\pi$), which gives the rotation matrix, obtained using the Rodrigues formula

$$\exp(\boldsymbol{\omega}\Delta t) = \mathbf{I} + \sin(\|\boldsymbol{\omega}\|\Delta t) \frac{[\boldsymbol{\omega}]_{\times}}{\|\boldsymbol{\omega}\|} + (1 - \cos(\|\boldsymbol{\omega}\|\Delta t)) \frac{[\boldsymbol{\omega}]_{\times}^2}{\|\boldsymbol{\omega}\|^2}, \quad (7)$$

where $[\boldsymbol{\omega}]_{\times}$ denotes a skew-symmetric matrix composed of elements of vector $\boldsymbol{\omega}$. Scalar $\nu(k)$ and vector $\boldsymbol{\omega}(k)$ constitute a unique differential time-based description of a space curve describing the assembled object position and orientation. Moreover, the spatial position and orientation of the curve depends only on the initial pose, given by $\mathbf{T}(0)$. For this reason, we call this formulation a pose-invariant notation of motion.

A more compact representation can be obtained using unit quaternions. The desired pose passed to the robot controller is often given with a generalized vector in a form

$$\chi(k) = [\mathbf{p}^T(k) \ \mathbf{q}(k)] \quad (8)$$

where $\mathbf{p} \in \mathbb{R}^3$ is the position vector composed of spatial coordinates x, y and z , while $\mathbf{q} \in \mathbb{R}^4$ is a unit quaternion, describing object orientation. We use the notation $\mathbf{q} = v + \mathbf{u}$, where v and \mathbf{u} are the scalar and vector part of the quaternion. The Eq. (4) can be represented in a form

$$\chi(k) = [\mathbf{p}(k-1) + \nu(k)\Delta t \Delta\mathbf{q}(k) * [0 + [1 \ 0 \ 0]^T] * \overline{\Delta\mathbf{q}(k)}, \quad \Delta\mathbf{q}(k) * \mathbf{q}(k-1)] \quad (9)$$

Operator $\overline{(\cdot)}$ stands for a conjugate quaternion. Applying the well-known relation

$$\Delta\mathbf{q}(k) = \exp(\boldsymbol{\omega}(k)\Delta t) \quad (10)$$

we obtain

$$\chi(k) = [\mathbf{p}(k-1) + \nu(k)\Delta t \exp(\boldsymbol{\omega}(k)\Delta t) * [0 + [1 \ 0 \ 0]^T] * \exp(\overline{\boldsymbol{\omega}(k)\Delta t(k)}), \exp(\boldsymbol{\omega}(k)\Delta t) * \mathbf{q}(k-1)]. \quad (11)$$

For quaternions, the exponential map is defined as

$$\exp(\boldsymbol{\omega}) = \begin{cases} \cos(\|\boldsymbol{\omega}\|) + \sin(\|\boldsymbol{\omega}\|) \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}, & \boldsymbol{\omega} \neq 0 \\ 1 + [0, 0, 0]^T, & \text{otherwise} \end{cases} \quad (12)$$

The new trajectory sample depends only on the previous generalized vector $\chi(k-1)$ and four properties, calculated from $\nu(k)$ and $\boldsymbol{\omega}(k)$, that are pose invariant. In the remainder of the text we refer to them as invariances.

A. Learning of pose-invariant trajectory representation

Unlike other invariant trajectory representations, which allow describing unconstrained 6 d.o.f motion, calculating invariances is straightforward. Moreover, the solution always exists, and it is unique. From a set of sampled trajectory tuples $[\mathbf{p}(k) \ \mathbf{q}(k)], k = 1 \dots T$, translational and angular velocities of the object frame are calculated as

$$\nu(k) = \|\mathbf{p}(k) - \mathbf{p}(k-1)\|/\Delta t \quad (13)$$

$$\boldsymbol{\omega}(k) = 2 \log(\mathbf{q}(k) * \overline{\mathbf{q}(k-1)})/\Delta t \quad (14)$$

The quaternion logarithm $\log: \mathbf{S}^3 \mapsto \mathbb{R}^3$, that maps the quaternion \mathbf{q} to the angular velocity $\boldsymbol{\omega}$, is defined as

$$\log(\mathbf{q}) = \log(v, \mathbf{u}) = \begin{cases} \arccos(v) \frac{\mathbf{u}}{\|\mathbf{u}\|}, & \mathbf{u} \neq 0 \\ [0, 0, 0]^T, & \text{otherwise} \end{cases} \quad (15)$$

Next, we define a path variable s as a weighted arc length

$$s(t) = \int_0^{t_{max}} (\nu(t) + \gamma\|\boldsymbol{\omega}(t)\|) dt. \quad (16)$$

Scalar γ compensates for the different translational and rotational motion metrics. For the discrete-time, it turns to

$$s(k) = \sum_1^T \|\mathbf{p}(k) - \mathbf{p}(k-1)\| + 2\gamma\|\log(\mathbf{q}(k) * \overline{\mathbf{q}(k-1)})\|, \quad (17)$$

$t_{max} = T\Delta t$. Using $s(k)$ as a phase variable, we encode $\nu(k)$ and $\boldsymbol{\omega}(k)$ with a sum of M Gaussian radial base functions Ψ in the form

$$\hat{\nu}(s) = \mathbf{x}(s)\mathbf{w}_{\nu}, \quad (18)$$

$$\hat{\boldsymbol{\omega}}(s) = \mathbf{X}(s)\mathbf{W}_{\boldsymbol{\omega}} \quad (19)$$

$$\mathbf{x}(s) = \frac{[\Psi_1(s), \dots, \Psi_N(s)]}{\sum_{j=1}^N \Psi_j(s)}, \quad (20)$$

$$\mathbf{X}(s) = \begin{bmatrix} \mathbf{x}(s) \\ \mathbf{x}(s) \\ \mathbf{x}(s) \end{bmatrix} \quad (21)$$

$$\Psi_j(s) = \exp(-h_j(s - c_j)^2), \quad j = 1, \dots, M. \quad (22)$$

The vector $\mathbf{w}_{\nu} \in \mathbb{R}^M$ and matrices $\mathbf{W}_{\boldsymbol{\omega}} \in \mathbb{R}^{3 \times M}$ contain free parameters that determine the shape of the encoded variables $\hat{\nu}(s)$ and $\hat{\boldsymbol{\omega}}(s)$, c_j are the centers of RBFs and h_j their widths. Usually, they are selected so that RBFs are evenly distributed along the trajectory. RBF weights are learned after the trajectory demonstration by regression [13].

B. Reconstruction of robot policy from pose-invariant trajectory representation

Invariant representations are very suitable for motion recognition, as we can directly compare invariances of two translated and rotated trajectories [11], [10], [14]. However, invariant policy descriptions can also be used to control a robot. The advantage of such a description is that the policy is the same no matter how the trajectory has to be translated and rotated in space. This is because the rotation

and translation are determined relative to the previous robot pose. Another, perhaps more important benefit is that the assembled object can move freely (with some limitations) during the assembly. At the same time, the policy inherently adapts to the geometric changes of the environment, e.g., the base of an assembled object.

For proper execution of the invariant policies, it is extremely important to estimate proper values of invariances at the robot's current pose and, consequently, precise estimation of the phase variable s . As we will see in the next section, we allow the robot to adapt to environmental changes by setting the controller compliant in all axes except in the tangential direction of motion. For this reason, we can not apply Eq. 17 for phase calculation anymore. Let's consider, for example, the simplest possible case, where the robot has to follow a straight rod (or insert a round peg into a hole, which is the equivalent case). We allow the rod to change its position and orientation constantly. In such a case, the robot performs substantial position and rotational motion. On the other hand, all that matters is the relative path along the rod. Relative paths on the rod can be obtained by projecting the robot's motion to the vector aligned with the current rod orientation. This vector can be calculated from the current object pose and the predicted object pose, given by Eqs. (4–6). During the online reconstruction of the robot policy, the phase variable is thus calculated as

$$s(k) = \sum_1^T (\mathbf{p}(k) - \mathbf{p}(k-1) + 2\gamma \log(\mathbf{q}(k) * \bar{\mathbf{q}}(k-1))) \cdot \mathbf{a}(k). \quad (23)$$

In the above equation, the operator (\cdot) denotes the scalar product and $\mathbf{a}(k) = \mathbf{R}_o(k)[1 \ 0 \ 0]^T$ is the x axis of the object frame. The calculation of the phase variable also reveals the main limitation of our approach to robust assembly. Namely, the assembled object must never move in the direction of the current tangent of the assembly trajectory, as this results in improper phase determination.

When controlling the robot, the robot's actual position lags behind the reference position due to uncompensated friction in the joints and friction with the robot environment. Therefore, in the incremental generation of the trajectory, where we feed an offset from the current position, the robot often may not move at all. To overcome the above-mentioned problem, we propose adaptive lag compensation. The phase error is defined as

$$e_s(s) = \|\dot{\mathbf{p}}_d(s) - \dot{\mathbf{p}}(s)\| + \gamma \|\boldsymbol{\omega}_d(s) - \boldsymbol{\omega}(s)\| \quad (24)$$

Based on the estimated phase error, we calculate a new phase variable

$$s^* = s + \delta \int e_s(s) dt, \quad (25)$$

which is used to encode $\hat{\nu}(k)$ and $\hat{\omega}(k)$ that are feed to the robot controller

$$\hat{\nu}(s) = \mathbf{x}(s^*) \mathbf{w}_\nu, \quad (26)$$

$$\hat{\omega}(s) = \mathbf{X}(s^*) \mathbf{W}_\omega. \quad (27)$$

δ is a suitably chosen or learned positive constant. The proposed lag compensation algorithm effectively compensates

for the unknown friction and assures the robot follows the desired path with a given velocity.

Summarizing, during the reconstruction, the phase is calculated by Eq. (23), translational and angular velocities $\hat{\nu}(s)$ and $\hat{\omega}(s)$ are calculated using Eq. (26, 27) and the next trajectory sample, which is passed to the robot controller, is obtained using Eq. (9).

III. ADAPTIVE IMPEDANCE CONTROLLER (AIC)

This section presents a controller that can autonomously adapt to environmental constraints. We assume that the environment constrains the robot's motion so that only one direction is possible at any given time. Rotations are also constrained so that the robot's orientation can only change around this direction. It is referred to as the tangential direction of the assembly trajectory. We formalize this motion control by utilizing the *object frame* \mathbf{R}_o introduced in the previous section, which is attached to the robot tool center point (TCP). Given the object frame, we need a control law enabling arbitrary compliance application along the frame axes. In our experiments, we used the passivity-based impedance control designed for manipulators with flexible joints [15]. However, it was necessary to modify the control law to freely set compliance in the object frame. The commanded torque $\boldsymbol{\rho}_c \in \mathbb{R}^N$, which is passed to the robot motors, is calculated as ¹

$$\begin{aligned} \boldsymbol{\rho}_c &= \mathbf{B} \mathbf{B}_\Theta^{-1} \mathbf{u} + (\mathbf{I} - \mathbf{B} \mathbf{B}_\Theta^{-1}) \boldsymbol{\rho}, \quad (28) \\ \mathbf{u} &= \mathbf{J}^T(\boldsymbol{\theta}) \left(\begin{bmatrix} \mathbf{f}_c \\ \mathbf{m}_c \end{bmatrix} + \begin{bmatrix} \mathbf{f}_a \\ \mathbf{m}_a \end{bmatrix} \right) + \mathbf{g}(\boldsymbol{\theta}) + \mathbf{N}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}}_0, \end{aligned}$$

where N is the number of robot joints, $\boldsymbol{\theta} \in \mathbb{R}^N$ is the vector of joint angles estimated from the corresponding motor angles $\Theta \in \mathbb{R}^N$ [16], $\mathbf{J} \in \mathbb{R}^{N \times 6}$ is the manipulator Jacobian, while \mathbf{B} , $\mathbf{B}_\Theta \in \mathbb{R}^{6 \times 6}$ denote the positive definite diagonal matrices of the actual and the desired joint inertia, respectively. The aim of the term $\mathbf{B} \mathbf{B}_\Theta^{-1}$ is to reduce the joint inertia. $\boldsymbol{\rho}$ are the measured joint torques, and $\mathbf{g}(\boldsymbol{\theta})$ is the gravity vector [17]. $\mathbf{N}(\boldsymbol{\theta}) = (\mathbf{I} - \mathbf{J}^+(\boldsymbol{\theta}) \mathbf{J}(\boldsymbol{\theta})) \in \mathbb{R}^{N \times N}$ is the null space projection operator, $\mathbf{J}^+(\boldsymbol{\theta})$ denotes the Moore-Penrose pseudo-inverse of the Jacobian and $\dot{\boldsymbol{\theta}}_0 \in \mathbb{R}^N$ is the null space velocity vector. \mathbf{f}_a and \mathbf{m}_a are additional forces and torque vectors in task coordinates, which are used to override the effect of friction forces during the assembly. The motor torque controller (28) reduces the motor inertia and compensates for the robot's non-linear dynamics. In contrast, the second part of Eq. (28) provides for the desired impedance and damping, additional task force, and null space motion. The task command input $[\mathbf{f}_c^T, \mathbf{m}_c^T]^T$ is chosen as

$$\mathbf{f}_c = -\mathbf{R}_o \mathbf{D}_p \mathbf{R}_o^T \dot{\mathbf{p}} + \mathbf{R}_o \mathbf{K}_p \mathbf{R}_o^T \mathbf{e}_p, \quad (29)$$

$$\mathbf{m}_c = -\mathbf{R}_o \mathbf{D}_o \mathbf{R}_o^T \boldsymbol{\omega} + \mathbf{R}_o \mathbf{K}_o \mathbf{R}_o^T \mathbf{e}_q, \quad (30)$$

where position and orientation tracking errors are defined as $\mathbf{e}_p = \mathbf{p}_d - \mathbf{p}$ and $\mathbf{e}_q = 2 \log(\bar{\mathbf{q}} * \mathbf{q}_d)$. \mathbf{K}_p and $\mathbf{K}_o \in$

¹for the sake of simplicity, we omitted discrete time index k in the following equations

$\mathbb{R}^{3 \times 3}$ are the diagonal matrices that define the positional and rotational stiffness along and around coordinate axes. \mathbf{D}_p and $\mathbf{D}_o \in \mathbb{R}^{3 \times 3}$ are diagonal damping matrices, which are set as diagonal elements of the block diagonal matrix calculated so that the overall system is critically damped. The null space velocity has to be controlled to prevent non-conservative motion. An appropriate solution is to set the desired null space velocities to zero, $\dot{\boldsymbol{\theta}}_0 = -\mathbf{K}_n \dot{\boldsymbol{\theta}}$, which results in an energy dissipation controller [18]. $\mathbf{K}_n \in \mathbb{R}^{N \times N}$ is a positive-definite diagonal gain matrix.

Next, it is essential to provide the necessary controller stiffness when the robot is expected to change its direction of motion, i.e. the controller should adapt its compliance according to the commanded motion. We propose the following formulation of variable compliance:

$$\mathbf{K}_p = \begin{bmatrix} k_{p_x} |\nu| & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (31)$$

$$\boldsymbol{\omega}_o = \mathbf{R}_o \boldsymbol{\omega} \quad (32)$$

$$\mathbf{K}_o = \begin{bmatrix} k_{o_x} |\omega_{o_x}| & 0 & 0 \\ 0 & k_{o_y} |\omega_{o_y}| & 0 \\ 0 & 0 & k_{o_z} |\omega_{o_z}| \end{bmatrix} \quad (33)$$

To preserve stability, the diagonal elements of \mathbf{K}_p and \mathbf{K}_o are limited by an upper bound.

The variable compliance along the object frame ensures the basic adaptation of the regulation law according to the limitations of the environment. However, this adaptation alone may not be enough. Such a control law might fail due to uncompensated friction in the robot joints and/or friction between individual objects during assembly, especially when inserting pegs into very tight holes. Different measures should be taken to avoid the above-mentioned problems. Force control is often used to realize a remote center of compliance [19]

$$\mathbf{f}_a = \mathbf{K}_f \mathbf{R}_o (\mathbf{f}_d - \mathbf{f}), \quad (34)$$

$$\mathbf{m}_a = \mathbf{K}_m \mathbf{R}_o (\mathbf{m}_d - \mathbf{m}). \quad (35)$$

\mathbf{K}_f and \mathbf{K}_m are force controllers gains, \mathbf{f}_d and \mathbf{m}_d are the desired forces and torques (usually set to 0), and \mathbf{f} and \mathbf{m} are measured forces and torques from the sensor in tool coordinates. Transformation \mathbf{R}_o maps the measured forces and torques from the tool coordinate system to the robot coordinate system.

Another approach is to learn the appropriate additional forces and torques $\mathbf{f}_a, \mathbf{m}_a$, as proposed in [20]. In their work, the authors propose to superimpose force and torque oscillations, where the meta parameters of the direction, amplitude, and frequency of oscillating force are learned. Others propose a dithering signal with predefined constant parameters [21], [22]. In our approach, we also chose a constant frequency of oscillations. In contrast, the direction of the oscillations was adapted according to the tangent of the movement, and the amplitude depended on the tracking

error. Thus, the superimposed forces are given by

$$\mathbf{f}_a(k) = \mathbf{R}_o [1 \ 0 \ 0]^T \kappa (|\mathbf{e}_p(k)|) (\sin(k\pi/4) + 1) \quad (36)$$

where κ is a constant that determines the superimposed force amplitude and vectors \mathbf{p}_d, \mathbf{p} denote commanded and measured position, respectively.

The corresponding overall control scheme for real-time adaptation of assembly policy is outlined in Fig. 3.

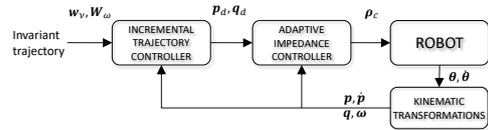


Fig. 3. Incremental controller block scheme.

IV. INITIAL POSE SEARCH WITH ERGODIC CONTROL

For the robot to follow the environmental constraints imposed by the assembly operation, it must reach the required initial pose with generally very low tolerances. Vision processing alone cannot assure the required accuracy; therefore, we propose combining vision with an active search algorithm. For active search, we apply a framework based on ergodic search [23], [24]. In contrast to the standard control problem, where the goal is to track the robot's pose, the goal of ergodic control is to track a distribution that needs to be covered by the robot. Given the probability distribution, the resulting system has natural exploration behaviors by considering information about the regions that should be explored.

Ergodic search can be viewed as a trajectory generator whose input is the probability distribution. The output is a smooth continuous trajectory that visits spatially distributed locations according to the given probability distribution. The original approach [23] is based on quadratic cost minimization and representation of the trajectories with Fourier series. The Fourier series coefficients are computed using multi-dimensional integration over the spatial domain, which can be computationally costly for high-dimensional state spaces. Moreover, the control loop involves algebraic operations on multi-dimensional arrays, making the original approach too slow for real-time applications in high-dimensional tasks. This problem was addressed in [24] by relying on tensor trains, an approach for low-rank factorization of tensor data, used to compute Fourier coefficients efficiently. Consequently, the ergodic trajectory can be computed in real-time, even for six DOF distributions. The proposed approach was experimentally verified in a peg-in-hole task, demonstrating the robustness of the approach to peg displacements within the robot gripper. Comparison with traditional search methods such as Gaussian Mixture Model (GMM) sampling and spiral search demonstrated that an ergodic controller exceeds other methods in terms of speed (time to find the hole) and success rate. Fig. 4 shows a typical trajectory generated by ergodic control and a trajectory generated with random sampling for a three-DOF problem.

Ergodic control is thus a perfect candidate for an active search for the initial pose in assembly tasks. However, to limit the search space, we apply vision processing. The assembled object is captured with an RGB-D camera. The point cloud $\mathcal{P}_0 = \{p_i\}_{i=1}^N$ consisting of N points on the object is associated with a probability distribution for the initial assembly pose. Point cloud and probability distributions are associated by computing a homogeneous transformation \mathbf{T}_a between their centers. We assume that we obtained an assembly policy for the object at a location giving rise to point cloud \mathcal{P}_0 . Next time we perform the assembly on a possibly translated and rotated object, we capture the point cloud \mathcal{P}_1 and compute the transformation matrix \mathbf{T}_b between \mathcal{P}_0 and \mathcal{P}_1 . In our case, transformation matrix \mathbf{T}_b was computed using the *CloudCompare* [25] implementation of the Iterative Closest point (ICP) algorithm [26].

To calculate the probability distribution for the assembled object pose, we apply the transformation $\mathbf{T}_a \mathbf{T}_b \mathbf{T}_a^{-1}$ to the probability distribution at the original pose. Next, we calculate the corresponding ergodic trajectory and execute it. When the robot finds the initial pose² We start the assembly by taking the current robot pose as the initial. Since the trajectory is computed incrementally using Eq. (9), we do not need to apply any other transformation.

V. EXPERIMENTAL EVALUATION

In this section, we experimentally verify the proposed framework for the robust execution of contact policies, applied to the seven degrees-of-freedom collaborative robot Franka Research 3 equipped with a two fingers gripper. The AIC controller was implemented using the *libfranka* library and *ros_control* framework in C++. The incremental controller, ergodic controller, and vision processing were implemented in Matlab and Python and communicated with the AIC using ROS. Point clouds were captured using the Intel RealSense D435i RGB-D camera.

The first experiments were done on a mockup covering all the essential assembly problems. It consists of a curved rod on which there is a sliding bearing. The sliding bearing is equipped with a screw mechanism, which is used to fix

²Usually, we use position and force measurements to detect that the robot has found the initial pose. This detection method is case-specific.

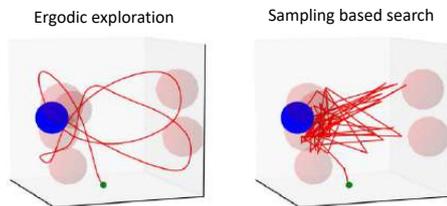


Fig. 4. Example trajectories of ergodic exploration strategy (left) and the sampling-based search (right). The GMM has six equally weighted mixture components (red spheres). Blue sphere is the selected target region within the reference probability distributions. Note that the target region is unknown to the search algorithm.

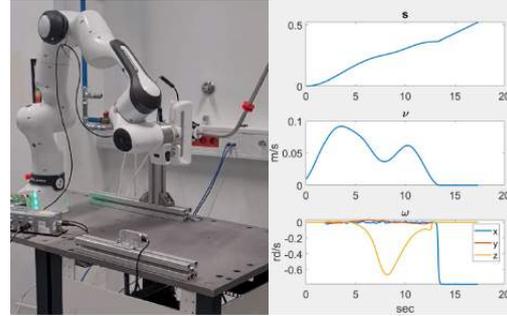


Fig. 5. Left: A mockup for testing assembly operations Right: Phase variable and commanded trajectory invariances.

it to the thread attached to the rod's other end. The rod is clamped at one end only; the other end can be moved freely. The mockup is shown in Figure 5.

The initial policy was learned by kinesthetic teaching with the robot in gravity compensation mode and recalculated in a pose-invariant incremental representation. The corresponding invariant variables are shown in Fig. 5. A simple replay of the learned trajectory failed. The robot got stuck in the curved part of the rod. The results are shown in Fig. 6 left, where we can notice increased force in the local x direction as the robot gets stuck. Next, we applied the AIC proposed in Section III. The robot successfully accomplished the task, as shown in Fig. 6 center. The increased force in x direction is due to the contact with the thread attached at the rod end during the screwing. However, it could not complete the task if we randomly moved the other end of the rod extensively during the task. Finally, we performed the same experiment by applying incremental policy updates from pose invariances (Subsection II-B) and AIC. The robot accomplished the task successfully despite extensive disturbances induced by randomly moving the free side of the rod, as long as the robot joints stayed within physical limitations. The robot trajectory, disturbed by the operator and the corresponding forces and torques, are shown in Fig. 6 right. Note also that the contact forces didn't increase substantially despite the disturbances, as AIC suppressed them effectively.

The next experiment was the bayonet bulb insertion for experimental evaluation. The first part of the task is a peg-in-hole task, where a gap in the bayonet bulb casing determines the bulb's orientation. The next part is the rotation of the bulb to lock it in the final position. The car bulb and the casing (see Fig. 8 left) were 3D printed and fit each other well³. The assembly policy was learned by demonstration. After learning, we captured the bulb casing point cloud \mathcal{P}_0 . Next, we captured 32 images of the bulb casing from 32 perspectives and calculated the centers and principal axes of the gathered point clouds using the *CloudCompare* framework. The point clouds and the probability distributions for positions in a

³Please note that due to the low tolerances, assembly of the printed bulb was more challenging than the assembly of the real bulb, considered in our previous research [27]

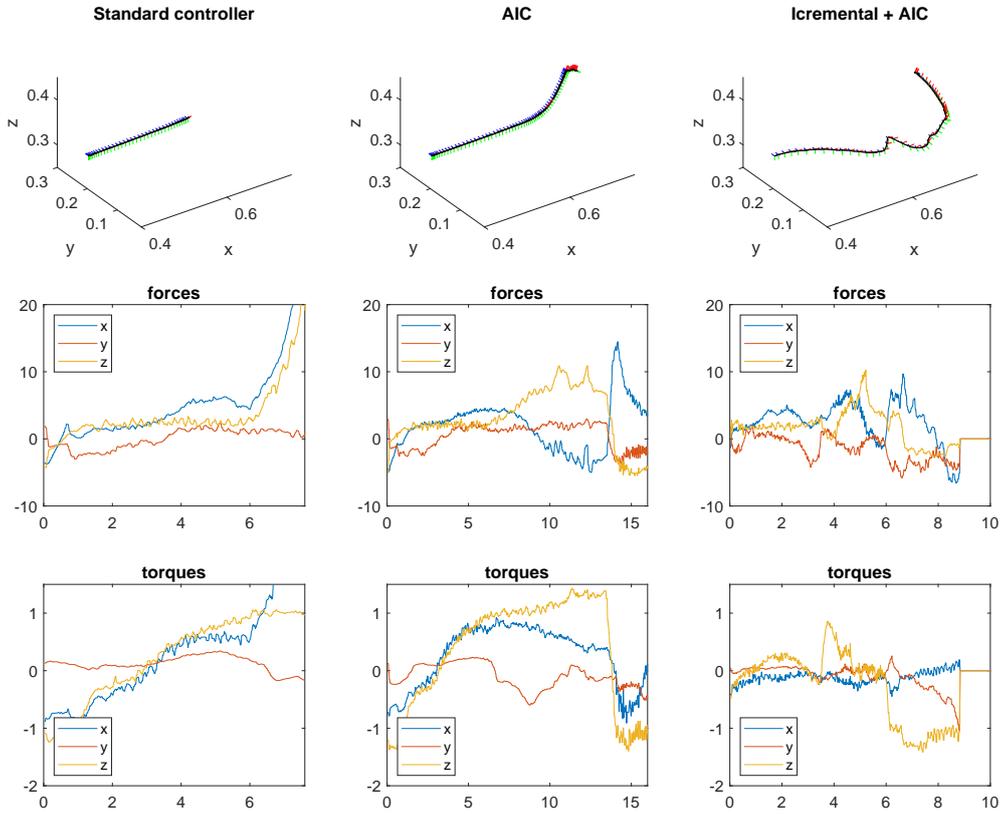


Fig. 6. Robot trajectories, contact forces, and torques obtained with different control algorithms and trajectory generation schemes.

GMM form are shown in Fig. 7, left. Finally, we encoded the assembly policy in a position/orientation invariant form, as described in Section II.

To test the robustness of assembly regarding the unknown position and orientation of the bulb casing, we mounted it on an inclined plate whose position and rotation were unknown to the robot (see Fig. 8). We recorded the point cloud \mathcal{P}_1 and displaced the GMM distribution associated with \mathcal{P}_0 to the new pose (Fig. 7 right). Based on the displaced distribution, the ergodic controller found a new initial pose for the assembly in 9 seconds on average. After that, the robot successfully inserted the bulb in a casing by incremental policy updates from pose invariances and the proposed AIC. The phase variable and invariances are shown in Fig. 8. Finally, we repeated the same experiment, but this time the case was held by a human performing random displacements of the housing. Nevertheless, the robot successfully inserted the bulb into the housing using the proposed framework. The corresponding videos of all experiments illustrate the robust assembly using invariant policy representation.

VI. CONCLUSION

In this research, we considered the problem of robust execution of the assembly tasks, where the pose of the assembled object is not known in advance. Regardless of whether we can precisely determine the starting point of the assembly, even minimal changes in the orientation assessment can cause large deviations, especially for extended objects. In addition, the usual assembly procedures fail if the object's posture changes during assembly. Our main challenge was developing a methodology for assembly policy generation that seamlessly adapts to environmental changes.

We addressed two problems, a) how to generate an environmentally adaptable assembly policy and b) determine the starting point for implementing an assembly policy. To solve the first problem, we proposed incremental execution of the trajectory using AIC and pose-invariant formulation of the policy. We combined point clouds and an ergodic controller to solve the second problem.

The proposed approach is suitable for assembly operations and can be used for many other tasks where the robot is in

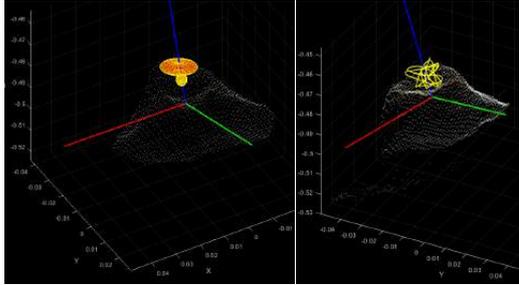


Fig. 7. Left: Point cloud of the bulb casing \mathcal{P}_0 (grey) and the GMM for the positional distribution used in ergodic controller (yellow-red); Right: Displaced point cloud \mathcal{P}_1 (grey) and a positional trajectory for the ergodic search (yellow).

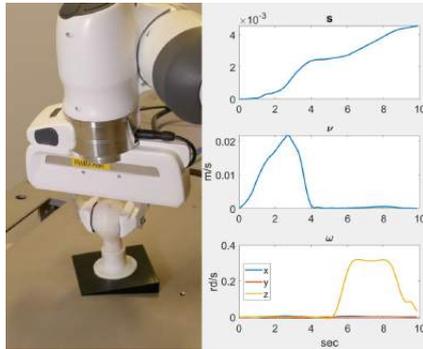


Fig. 8. Left: Franka robot during the execution of the position and orientation invariant policy for the bayonet bulb insertion into the casing. Right: Phase and commanded trajectory invariances for bayonet bulb insertion.

constant contact with the environment. The requirement that the assembled object never moves in the direction of the current tangent of the assembly trajectory can be overridden by additional sensors, e.g., vision. Further steps in our research will extend our framework to assembly cases where a component carried by the robot may shift in the gripper due to imperfect grasping.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Horizon 2020 Research and Innovation Action ReconCycle, GA no. 871352.

REFERENCES

- [1] L. Evjemo, T. Gjerstad, E. Grøtli, and G. Sziebig, "Trends in smart manufacturing: Role of humans and industrial robots in smart factories," *Current Robotics Reports*, vol. 1, 06 2020.
- [2] I.-W. Kim, D.-J. Lim, and K.-I. Kim, "Active peg-in-hole of chamferless parts using force/moment sensor," in *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, vol. 2, 1999, pp. 948–953 vol.2.
- [3] F. J. Abu-Dakka, B. Nemej, J. A. Jørgensen, T. R. Savarimuthu, N. Krüger, and A. Ude, "Adaptation of manipulation skills in physical contact with the environment to reference force profiles," *Autonomous Robots*, vol. 39, no. 2, pp. 199–217, 2015.
- [4] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, and K. Harada, "Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach," *Applied Sciences*, vol. 10, no. 19, p. 6923, 2020.

- [5] M. Suomalainen, Y. Karayiannidis, and V. Kyrki, "A survey of robot manipulation in contact," *Robotics and Autonomous Systems*, vol. 156, p. 104224, 2022.
- [6] S. K. Yun, "Compliant manipulation for peg-in-hole: Is passive compliance a key to learn contact motion?" *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1647–1652, 2008.
- [7] D. Aarno and D. Kragic, "Motion intention recognition in robot assisted applications," *Robotics and Autonomous Systems*, vol. 56, no. 8, pp. 692–705, 2008.
- [8] M. Saveriano and D. Lee, "Invariant representation for user independent motion recognition," in *2013 IEEE RO-MAN*, 2013, pp. 650–655.
- [9] Y. Piao, K. Hayakawa, and J. Sato, "Space-time invariants for recognizing 3D motions from arbitrary viewpoints under perspective projection," *Third International Conference on Image and Graphics (ICIG'04)*, pp. 200–203, 2004.
- [10] M. Vochten, T. D. Laet, and J. De Schutter, "Generalizing demonstrated motion trajectories using coordinate-free shape descriptors," *Robotics and Autonomous Systems*, vol. 122, p. 103291, 2019.
- [11] R. Soloperto, M. Saveriano, and D. Lee, "A bidirectional invariant representation of motion for gesture recognition and reproduction," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6146–6152.
- [12] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.
- [13] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 800–815, oct 2010.
- [14] Y. Li, R. Xia, and X. Liu, "Learning shape and motion representations for view invariant skeleton-based action recognition," *Pattern Recognition*, vol. 103, p. 107293, 2020.
- [15] A. Albu-Schaffer, C. Ott, and G. Hirzinger, "A unified passivity-based control framework for position, torque and impedance control of flexible joint robots," *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.
- [16] C. Ott, A. Albu-Schaffer, A. Kugi, and G. Hirzinger, "On the Passivity-Based Impedance Control of Flexible Joint Robots," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 416–429, 2008.
- [17] C. Ott, A. Albu-Schaffer, A. Kugi, S. Stramigioli, and G. Hirzinger, "A passivity based Cartesian impedance controller for flexible joint robots - part I: torque feedback and gravity compensation," in *IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, 2004, pp. 2659–2665.
- [18] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Robotics and Automation, IEEE Journal of*, vol. 3, pp. 43–53, 1987.
- [19] J. Jiang, Z. Huang, Z. Bi, X. Ma, and G. Yu, "State-of-the-art control strategies for robotic PiH assembly," *Robotics and Computer-Integrated Manufacturing*, vol. 65, pp. 1–26, 2020.
- [20] L. Johansmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5844–5850, 2019.
- [21] S. Ipri and H. Asada, "Tuned dither for friction suppression during force-guided robotic assembly," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 1, 1995, pp. 310–315 vol.1.
- [22] A. Stolt, A. Robertsson, and R. Johansson, "Robotic force estimation using dithering to decrease the low velocity friction uncertainties," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3896–3902.
- [23] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4, pp. 432–442, 2011.
- [24] S. Shetty, J. Silvério, and S. Calinon, "Ergodic exploration using tensor train: Applications in insertion tasks," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 906–921, 2022.
- [25] "Cloud compare v2.13 GPL software," <http://www.cloudcompare.org/>, 2023.
- [26] Z. Zhang, *Iterative Closest Point (ICP)*. Boston, MA: Springer US, 2014, pp. 433–434.
- [27] M. Simonič, A. Ude, and B. Nemej, "Hierarchical learning of robotic contact policies," *submitted to Robotics and Computer-Integrated Manufacturing*, 2023.

A.5 “Tactile Exploration Using Unified Force-Impedance Control”

This is the author’s manuscript of the following publication: K. Karacan, D. Grover, H. Sadeghian, F. Wu, and S. Haddadin. “Tactile Exploration Using Unified Force-Impedance Control”. In: *IFAC Papers OnLine, 22nd IFAC World Congress* 56.2 (2023), pp. 5015–5020.

Tactile Exploration Using Unified Force-Impedance Control

Kübra Karacan, Divij Grover, Hamid Sadeghian, Fan Wu, and Sami Haddadin

The Chair of Robotics and Systems Intelligence, MIRMI - Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, Germany, Centre for Tactile Internet with Human-in-the-Loop (CeTI). H. Sadeghian also has an affiliation with University of Isfahan.

Abstract: Tactile robots can perform complex interaction skills, e.g., polishing. Such robots should therefore be designed to be adaptive to environmental uncertainties such as changing geometry and contact-loss. To address this, we propose a tactile exploration technique to observe the local curvatures of the physical constraints such as corners, edges, etc. for updating predefined tactile skill policies accordingly. First, we develop a unified force-impedance control approach in which the force controller significantly improves the geometry following performance due to the ensured contact. Second, we use the proposed controller to autonomously investigate the unknown environment via the local curvature observer, designed to be a dynamic process. Finally, the exploration performance of the proposed controller is demonstrated by using a polishing skill on an unknown 3D surface, where the robot is observed to autonomously investigate the unknown surface from top to bottom along the edges and corners.

Keywords: Intelligent robotics, Autonomous robotic systems

1. INTRODUCTION

Robotic systems have been deployed in human sectors such as manufacturing, production, and service since the introduction of industrial robots. Such human sectors involve complex interaction tasks that must be executed in dynamic and unstructured environments. Traditional position-controlled industrial robots lack these adaptable and versatile characteristics. While new generation torque-controlled robots, as discussed by Hirzinger et al. (2001), provide the option of compliant behaviors via admittance control in Shahriari et al. (2017), impedance control in Hogan (1984), or force control in Khatib (1987). Additionally, Kirschner et al. (2021) demonstrated that the type of robot and controller are key factors in interaction skill performance.

Haddadin et al. (2019) discusses extensively the key to the use of tactile robots for performing complex interaction skills such as polishing is to utilize their ability of the recognition of touch. Preferably, as argued by Karacan et al. (2022), the human operator should give the archetypical solution to such robots, along with the desired tactile actions such as force and motion policies. Using polishing as an example, the human operator instructs the robot to polish a scratch on a surface with a desired force. However, the robot should be programmed to be adaptive to environmental uncertainties such as contact loss, which can damage the robot or the surrounding environment during applying force.

The state-of-the-art in tactile robot programming, on the other hand, still demands a fairly accurate model of the environment. Furthermore, not only is the task becoming more complex, but transferring it to a different environment demands re-programming and, as a result, expert knowledge, as shown

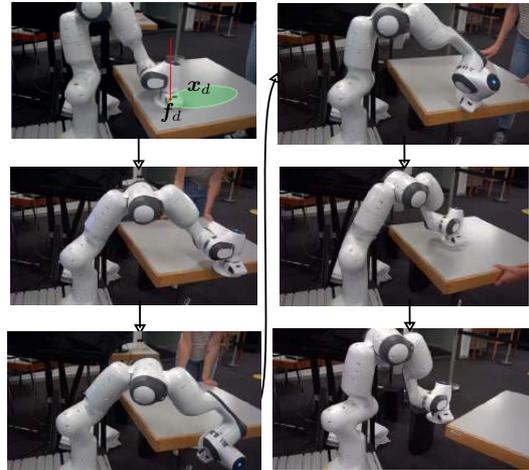


Fig. 1. Tactile exploration via local curvature observer. The robot follows a desired Cartesian motion while constantly applying a desired force in the z-direction. When the robot encounters an edge of the table (a 90-degree cliff), it rotates and adapts itself. If the contact is completely lost, the robot stops the force controller.

in Johannsmeier et al. (2019). Hence, new solutions to ease the (re-)programming of tactile tasks must be developed for increased flexibility and adaptation in tactile robot capabilities, see the study of Haddadin et al. (2022). To address the aforementioned problems, tactile exploration is a useful approach to

* Corresponding Author: Kübra Karacan kuebra.karacan@tum.de

investigate the unknown environment. Haddadin et al. (2010) and Haddadin et al. (2011) previously studied tactile exploration by leveraging novel collision avoidance algorithms and achieved preliminary results.

In this work, we develop an advanced tactile exploration strategy based on unified force-impedance control. Our novel exploration strategy investigates unknown environments via local curvature observer in real-time manner, designed to be a dynamic process (see Fig. 1). First, we develop a unified force-impedance control approach that allows the robot to adjust its impedance parameters to the environment. Here, the force controller is crucial as it significantly increases the geometry following performance due to the ensured contact. Second, we use the proposed controller to investigate autonomously the environment by observing the local curvatures. Finally, the tactile exploration performance of the controller is proven by using a polishing skill on an unknown 3D surfaces. The proposed method is used to polish a table from the top surface to its edges and finally to the bottom surface without knowing the model of the environment.

The rest of the paper is presented as follows. Section 2 discusses the related literature, and Sec. 3 introduces our novel unified force-impedance control-based tactile exploration strategy and the local curvature observer for tactile skills. The experimental protocol and the corresponding results are demonstrated in Sec. 4 and Sec. 5. Lastly, Sec. 6 finalizes the paper.

2. RELATED LITERATURE

Robotic tactile skills such as polishing, filing, and grinding need accurate control of interaction to the end-effector. Namely, force and motion policies should be combined to develop tactile skills. Hence, robotic systems need to develop complex perceptuo-motor abilities to handle such skills, as studied in Pastor et al. (2011). Furthermore, Hogan (1984) shows that position control is insufficient to realize such tactile skills and alternative control strategies should be considered, such as e.g., force control in Khatib (1987), using admittance control in Shahriari et al. (2017), and even unified methods in Schindlbeck and Haddadin (2015).

Additionally, numerous works have taken into account force control, validating the suggested controllers using constant force values or as thresholds or limitations, as discussed in Ficuciello et al. (2015); Kulakov et al. (2015); Ott et al. (2015); He et al. (2016). The robots must, however, be smart enough to operate autonomously in uncertain environments and complete required tasks when faced with perception uncertainties, shown in Pastor et al. (2012); Kramberger et al. (2016).

To attain a more human-like response for the contact at the end-effector, adaptive adjustment of the impedance parameters is helpful in many applications such as Yang et al. (2011). The Wavelet Neural Network has been applied by Hamedani et al. (2021) to integrate compliant force tracking on the unknown geometry.

Force information and historical position from the end-effector is used to predict shape profile in Qian et al. (2019). In another study, Lepora and Lloyd (2020) developed new tactile capabilities with the soft tactile sensors and neural networks include using a pose-based servo control, where a tactile fingertip mounted on a robot arm slides delicately over unknown complex 3D objects. There exist other exploration approaches such

as using mechanics in Balatti et al. (2020), learning (offline) in Simonić et al. (2019), and control theory in Kato et al. (2022).

In this paper, we propose a unified force-impedance control-based tactile exploration technique to observe the local curvatures of the physical constraints to understand corners, edges, etc., for updating the predefined tactile skill policies accordingly. First, we develop a unified force-impedance control approach that allows the robot to adjust its impedance parameters to the unknown environment. Using force control, in particular, makes exploration easier as the contact is ensured during the process. Second, we propose a dynamic process to observe the local curvature of the environment, and use it to align the tool while exploring the corresponding contact surface. Finally, the exploration performance of the controller is demonstrated by using a polishing skill on an unknown 3D surfaces.

3. METHODOLOGY

A robot expert establishes a tactile skill library relying on intuitive estimation of the physical constraints of the environment. Using the polishing skill as an instance, the robot expert trains the robot with certain tactile action policies, including the motion and force profiles. Yet, carrying out of the desired task, the robot should have the autonomy to adjust those intended tactile action policies to the entirely or partially unknown environment. Even if the robot has an external sensing capability, such as a camera, it may still operate within tolerance, particularly under a fairly cluttered environment. Hence, the tactile skills should be developed to allow the robot to adjust the appropriate tactile skills to the surroundings with as little interference from outside as possible.

Considering the fact that the robot runs a tactile skill defined for certain conditions, changes in the geometry could be inferred from the local curvature. Here, we use an exploration strategy to investigate the physical constraints of the environment such as corners and edges via local curvature observer. The exploration strategy utilizes adaptive-stiffness of the impedance controller as well as the robust contact via force controller, see Fig.2. Additionally, the recorded geometry information of the physical constraint might further be used to generate motion and force profiles for a new surface.

3.1 Preliminaries

The dynamics equation for a robot with n -DOF in Cartesian space is

$$\mathbf{M}_C(\mathbf{q})\ddot{\mathbf{x}} + \mathbf{C}_C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}} + \mathbf{f}_g(\mathbf{q}) = \mathbf{f}_{in} + \mathbf{f}_{ext}, \quad (1)$$

where $\mathbf{x}, \dot{\mathbf{x}} \in \mathbb{R}^6$ are the Cartesian pose and twist. The external wrench on the robot is $\mathbf{f}_{ext} \in \mathbb{R}^6$. $\mathbf{M}_C(\mathbf{q})$, $\mathbf{C}_C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{6 \times 6}$, and $\mathbf{f}_g(\mathbf{q})$ are the robot mass matrix, the Coriolis and centrifugal matrix, and the gravity vector in Cartesian space, respectively. Additionally, \mathbf{f}_{in} is the input wrench in Cartesian space and relates to the joint torques via robot's Jacobian matrix $\mathbf{J} \in \mathbb{R}^{6 \times n}$ $\boldsymbol{\tau}_{in} \in \mathbb{R}^n$ by $\boldsymbol{\tau}_{in} = \mathbf{J}^T(\mathbf{q})\mathbf{f}_{in}$.

3.2 Controller Design

The proposed control law for exploration is extended from unified force-impedance control by Schindlbeck and Haddadin (2015) with four main components:

- I) tracking the desired Cartesian pose \mathbf{x}_d with the impedance controller,

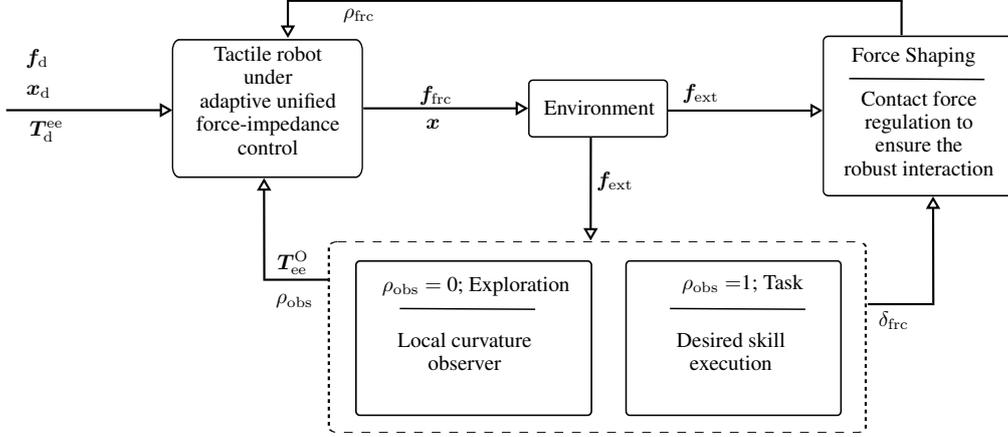


Fig. 2. Block diagram for unified force-impedance control-based local curvature observer for tactile exploration architecture. The predefined tactile skills are adjusted to the unknown environment by using force shaping ρ_{frc} and exploration blocks. The state switching is handled via ρ_{obs} .

- II) exploring the transformation matrix T_{ee}^0 ,
- III) following the commanded force f_d , and
- IV) gravity compensation τ_g .

The input torque $\tau_{in} \in \mathbb{R}^n$ is:

$$\tau_{in} = \tau_{imp} + \tau_{frc} + \tau_g, \quad (2)$$

where τ_{imp} , τ_{frc} , and $\tau_g \in \mathbb{R}^n$ are the input torques with respect to (i) impedance control (ii) force control and (iii) gravity compensation.

Impedance Control In order to establish a desired Cartesian impedance behavior at the tool, the following control law is defined

$$\tau_{imp} = J^T(q)(K_C \ddot{x} + D_C \dot{x} + M_C(q) \ddot{x}_d + C_C(q, \dot{q}) \dot{x}_d), \quad (3)$$

$$\ddot{x} = \ddot{x}_d - \ddot{x},$$

where $x \in \mathbb{R}^6$ is the actual pose and the pose error is \tilde{x} . Moreover, $K_C, D_C \in \mathbb{R}^{6 \times 6}$ are time-varying stiffness and damping matrices, respectively. The desired Cartesian inertia is assumed to be the actual robot inertia in Cartesian space.

Force Control To control the desired force at the end effector $f_d^{ee} \in \mathbb{R}^6$ w.r.t the external force $f_{ext}^{ee} \in \mathbb{R}^6$, we design the controller as follows,

$$\tau_{frc} = \rho_{frc} J(q)^T \begin{bmatrix} R_{ee}^O & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & R_{ee}^O \end{bmatrix} f_{frc}^{ee}, \quad (4)$$

$$f_{frc}^{ee} = f_d^{ee} + K_p \tilde{f}_{ext}^{ee} + K_i \int \tilde{f}_{ext}^{ee} dt + K_d \dot{\tilde{f}}_{ext}^{ee}, \quad (5)$$

$$\tilde{f}_{ext}^{ee} = f_d^{ee} + f_{ext}^{ee}, \quad (6)$$

where $f_{frc}^{ee} \in \mathbb{R}^6$ is a feedback force controller output with the diagonal matrices of the PID gains $K_p, K_i, K_d \in \mathbb{R}^{6 \times 6}$. Furthermore, force shaping function ρ_{frc} decides to activate or deactivate the force controller based on certain conditions.

Force Shaping Function In unified force-impedance control, if the pose error in any direction is beyond the predefined threshold for a direction δ_{frc} , for example, due to the environment change, the shaping function presents the contact-loss behavior and starts the force shaping function ρ_{frc} , which

degrades the effect of the force controller. In order to ensure a smooth transition, instead of a chattering behavior, ρ_{frc} interpolates in a user defined threshold δ_{frc} . δ_{frc} is the distance range in which the robot can move safely after the contact-loss takes place, while still exerting some force in the desired direction. The magnitude of force is however regulated by the force shaping function ρ_{frc} . Increasing δ_{frc} also increases the range of motion in which the robot is allowed to move while keep applying the force in the desired direction. Consequently, ρ_{frc} is

$$\rho_{frc} = \begin{cases} 1, & f_d^{eeT} \tilde{x}^{ee} \geq 0 \\ 0.5(1 + \cos(\pi(\frac{|\tilde{x}_z^{ee}|}{\delta_{frc}}))), & f_d^{eeT} \tilde{x}^{ee} < 0 \\ 0, & \wedge 0 < |\tilde{x}_z^{ee}| \leq \delta_{frc} \\ & \text{else} \end{cases} \quad (7)$$

Finally, the closed loop equation for the overall system becomes

$$M_C(q) \ddot{x} + C_C(q, \dot{q}) \dot{x} + D_C \dot{x} + K_C \tilde{x} + f_{frc} + f_{ext} = 0. \quad (8)$$

3.3 Tactile Exploration Strategy

The state machine in Fig. 3 explains the exploration strategy. Exploration consists of three major states: adaptation, task, and contact-loss. State adaptation allows the end-effector to adjust itself to the physical constraints of the environment.

The end effector of the robot is equipped with a tool with certain dimensions. When the tool is at an angle with the surface, the external force in z direction f_z^{ee} creates a moment about x and y-axis $M_{xy,ee} = [M_{x,ee} \ M_{y,ee}] \in \mathbb{R}^2$. L₂ norm of moments about x and y-axis $\|M_{xy,ee}\|_2$ gives us the length of $M_{xy,ee}$. Thus, local curvature lc is given as fraction of $M_{xy,ee}$ and f_z^{ee} . lc greater than the threshold value indicates that the tools need to be realigned to the surface.

Here, the curvature observer ρ_{obs} is computed from a normalized curvature coefficient α

$$\alpha = \frac{lc_{threshold} - lc}{lc_{threshold}}, \quad (9)$$

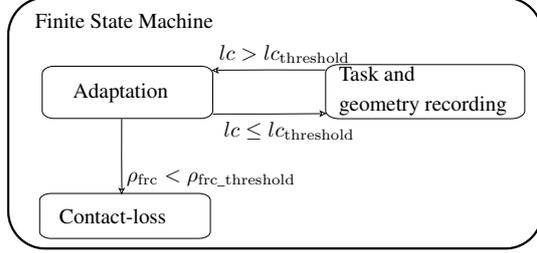


Fig. 3. State machine for exploration in simplified form. When the robot is in exploration state, the rotational stiffness about x- and y- directions $K_{r,x}$ and $K_{r,y}$ is zero. This allows the robot to follow the unknown geometry while it moves via the translational stiffness. And the stiffness matrix of the end-effector is also rotated.

where the curvature observer ρ_{obs} is obtained by the dynamics

$$\dot{\rho}_{obs} = \begin{cases} \min\{\rho, 0\}, & \rho_{obs} = 1 \\ \rho, & 0 < \rho_{obs} < 1, \rho_{obs}(0) = 0, \\ \max\{\rho, 0\}, & \rho_{obs} = 0 \end{cases} \quad (10)$$

and ρ is given by

$$\rho = \alpha\rho_{obs} + \rho_{min}. \quad (11)$$

Note that, in order to have an initial increment for the case $\rho_{obs} = 0$, a small positive constant ρ_{min} has been introduced into the curvature observer dynamics. It is noteworthy that the solution for the curvature observer dynamics allows us to have the behavior for increasing or decreasing exponential function based on the local curvature and, thus, is used to switch the between the exploration motion and the task.

The task state computes the new trajectory $T_{ee,traj}^0$ to follow, once the contact is re-established. And the robot runs the new trajectory until the end effector is no longer in contact with the environment, i.e., $lc > lc_{threshold}$ meaning $\rho_{obs} < 1$. In detailed, lc created due to external force f_z on the end effector indicates if the robot is in complete contact with the surface or not, and therefore is included in the curvature observer dynamics as a transition parameter between different states ($\rho_{obs} = 1$ or 0).

Algorithm 1 State adaptation and local curvature observer

Input $M_{x,ee}, M_{y,ee}, f_z, T_{ee}^0$
Output T_d^0

- 1: **Initialization:** $\{p_{traj}^0, R_{traj}^0\} = T_{traj}^0$,
- 2: **loop:**
- 3: $M_{xy,ee} = \|[M_{x,ee}M_{y,ee}]\|_2$,
- 4: $lc = \frac{M_{xy,ee}}{f_z}$,
- 5: **if** $\rho_{obs} < 1$ **then**
- 6: $R_d^0 = R_{ee}^0$
- 7: $p_d^0 = p_{traj}^0$
- 8: $T_d^0 = \{p_{traj}^0, R_{ee}^0\}$
- 9: $\delta_{frc} \leftarrow \delta_{frc} + \delta_{increment}$
- 10: **end if**
- 11: **goto loop.**
- 12: **close;**

The state of contact-loss makes sure that no unwanted rapid motions occur. If $\rho_{frc} < \rho_{frc,threshold}$, the contact-loss state

stops the robot and prevents unwanted rapid motion that could endanger the robot and the environment.

State adaptation explained in Alg. 1 tries to minimize the moment about x and y-axis. This is done by setting zero rotational stiffness about x and y-axis and at the same time maintaining the last position where the tool unaligned itself. The zero stiffness is achieved by setting the desired orientation around x- and y-axis as current orientation. This can be seen in line 7,8 and 9 in algorithm state adaptation. Therefore, the tool is free to move along x- and y-axis while applying force in desired direction. Moreover, in line 10, the translational

Algorithm 2 State Task and Information Saving

Input $M_{x,ee}, M_{y,ee}, f_z, T_{ee}^0$
Output T_d^0

- 1: **Initialization:** $T_{ee,temp}^0 = T_{ee}^0$,
- 2: **loop:**
- 3: $lc = \frac{M_{xy,ee}}{f_z}$,
- 4: **if** $\rho_{obs} = 1$ **then**
- 5: $T_{ee,traj}^0 = T_{ee,temp}^0 T_d^{ee}$
- 6: $T_d^0 = T_{ee,traj}^0$
- 7: $T_{record}^0 = T_d^0$
- 8: **end if**
- 9: **goto loop.**
- 10: **close;**

distance range δ_{frc} is incremented by the factor of $\delta_{increment}$ until the end-effector is able to apply the desired force onto the surface. Furthermore, if the tool attached to the end-effector is not in the desired contact with the surface, the reaction force applied by the environment creates a moment about x and y-axis. The resulting moment due to increase in δ_{frc} , zero rotational stiffness about x- and y-axis and low stiffness in z direction causes the robot to move in z-direction and at the same time adapt itself to always stay normal to the surface until the end-effector is able to make complete contact with the surface.

State task and information saving in Alg. 2 plans the trajectory and saves the current position and orientation of the end effector. This state is only active when the tool is completely aligned with the surface. The $T_{ee,temp}^0$ is updated only once at the entry of this state. Finally, the saved data may be used for future motion and force profile generation.

4. EXPERIMENTAL SETUP

To evaluate the exploration, adaptation, and control performance of our framework for the tactile skills to run under unknown physical constraints, the experiments are conducted using a Franka Emika robot for a polishing skill. Polishing requires the robot applying a desired wrench f_d^{ee} on the surface while tracking a circular motion $T_d^{ee}(t)$ represented by $R = I_{3 \times 3}$ and

$$p_d^{ee}(t) = [0.1 \sin(0.5t), 0.1 \cos(0.5t), 0]^T, \quad (12)$$

$$f_d^{ee} = [0, 0, 35 \text{ N}, 0, 0, 0]^T. \quad (13)$$

The radius of the tool is 0.06 m and therefore the threshold value $lc_{threshold}$ is taken as 0.025 m (see Table 1 for other parameters used).

The robot follows a trajectory in one direction while constantly maintaining force in the z-direction. The end-effector is able

Table 1. Parameters used throughout the experiments.

Property	Unit	Value
\mathbf{K}_c	N/m	diag[1000,1000,10,0,0,70]
ξ	Ns/m	diag[0.7,0.7,0.7,1,1,1]
$lc_{\text{threshold}}$	m	0.025
$\mathbf{K}_p, \mathbf{K}_i, \mathbf{K}_d$	-	$0.5\mathbf{I}_{6 \times 6}, 0.5\mathbf{I}_{6 \times 6}, 0\mathbf{I}_{6 \times 6}$
$\delta_{\text{frc}}(0)$	m	[0.05, 0.05, 0.05, 0.05, 0.05, 0.05]
$\rho_{\text{frc,threshold}}$	-	0.1
ρ_{min}	-	0.001

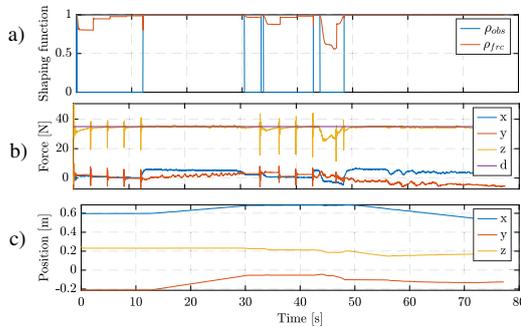


Fig. 4. Experimental results for polishing the car-door. During the contact, ρ_{frc} is 1 and regulates the external force. ρ_{obs} decides the state of the robot (exploration or task) to adapt to the changing geometry. a) Shaping functions ρ_{obs} and ρ_{frc} , b) External force at the end-effector, c) Actual position $x, y,$ and z w.r.t to the robot base frame.

to maintain the full contact, and it moves until the environment changes. In the first set of experiments, the robot should polish a car-door without knowing the geometry of the environment. The expected behavior of the robot is that the end-effector should follow the geometry while polishing it. Second, the robot should polish a table without geometry information. Moreover, when the robot encounters an edge of the table (a 90-degree cliff), it should be able to rotate and adapt itself. The goal is to make sure that the end-effector is always in full contact and aligned with the surface. Additionally, during the experiments, the robot's own internal sensing capabilities are used to obtain the external force/torque measurements, as analyzed by Haddadin et al. (2017).

5. RESULTS AND DISCUSSION

It can be seen in Fig. 4 and Fig. 5 that when the polishing tool is not in complete contact with the environment or the robot starts encountering the cliff, ρ_{frc} starts declining and lc starts increasing (ρ_{obs} decreases to zero), indicating that the tool needs to be realigned to the surface. Thus, the state is switched from the desired task ($\rho_{\text{obs}} = 1$) to tactile exploration ($\rho_{\text{obs}} = 0$). The exploration state is maintained until the polishing tool is completely aligned to the surface, meaning $lc \leq lc_{\text{threshold}}$ and $\rho_{\text{obs}} = 1$. After the polishing tool is adapted to the surface

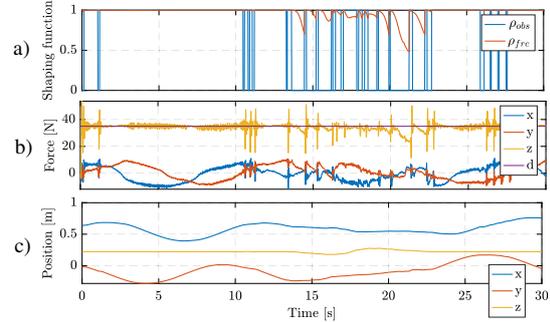


Fig. 5. Experimental results for polishing the table. During the contact, ρ_{frc} is 1 and regulates the external force. ρ_{obs} decides the state of the robot (exploration or task) to adapt to the changing geometry. a) Shaping functions ρ_{obs} and ρ_{frc} , b) External force at the end-effector, c) Actual position $x, y,$ and z w.r.t to the robot base frame.

geometry, the state changes back to the task and the polishing tool starts moving in the desired trajectory again.

It is noteworthy that, the limitation of this approach is that the local curvature observer depends on the type of the desired contact. For instance, the single point contact is unlikely to be achieved by using our method. Additionally, even though the robot adapts the parameters of impedance and force control according to the abrupt changes in the environment, the controller's stability still might be compromised, as discussed by Kronander and Billard (2016). Thus, one might consider installing virtual energy tanks to stabilize the controller, as introduced by Shahriari et al. (2022).

6. CONCLUSION

In this study, we develop a tactile exploration strategy based on unified force-impedance control via local curvature observer to update predefined tactile skill policies by ensuring the tool and contact surface alignment. First, we design a unified force-impedance control that allows the robot to adjust its impedance parameters to the unknown environment. Additionally, the force controller increases the geometry following performance owing to the ensured contact. Second, we use the controller to autonomously observe the local curvature of the environment, designed as a dynamic process. Finally, the tactile exploration performance of the controller is shown by using a polishing skill on the unknown 3D surfaces. As a future work, we will do feature extraction on the saved geometry information in order to further generate motion and force profiles for the new environment.

ACKNOWLEDGEMENTS

We gratefully acknowledge the funding by the European Union's Horizon 2020 research and innovation program as part of the project ReconCycle under grant no. 871352. This work was supported by LongLeif GaPa gGmbH (Geriatrics Project Y). The authors would like to thank the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for supporting the Lighthouse Initiative KI.FABRIK, (Phase 1: Infrastructure as well as the research and development program under, grant no. DIK0249), and the

German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy EXC 2050/1 Project ID 390696704 Cluster of Excellence Centre for Tactile Internet with Human-in-the-Loop (CeTI) of Technische Universität Dresden. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany (BMBF) in the programme of "Souverän. Digital. Vernetzt." Joint project 6G-life, project identification number 16KISK002.

Please note that S. Haddadin has a potential conflict of interest as a shareholder of Franka Emika GmbH.

REFERENCES

- Balatti, P., Kanoulas, D., Tsagarakis, N., and Ajoudani, A. (2020). A method for autonomous robotic manipulation through exploratory interactions with uncertain environments. *Autonomous Robots*, 44(8), 1395–1410. doi:10.1007/s10514-020-09933-w. URL <https://doi.org/10.1007/s10514-020-09933-w>.
- Ficuciello, F., Villani, L., and Siciliano, B. (2015). Variable impedance control of redundant manipulators for intuitive human-robot physical interaction. *IEEE Transactions on Robotics*, 31(4), 850–863. doi:10.1109/TRO.2015.2430053.
- Haddadin, S., Johansmeier, L., and Díaz Ledezma, F. (2019). Tactile robots as a central embodiment of the tactile internet. In *Proceedings of the IEEE*, volume 107, 471–487.
- Haddadin, S., Belder, R., and Albu-Schäffer, A. (2011). Dynamic motion planning for robots in partially unknown environments*. *IFAC Proceedings Volumes*, 44(1), 6842–6850. doi:<https://doi.org/10.3182/20110828-6-IT-1002.02500>. URL <https://www.sciencedirect.com/science/article/pii/S1474667016447051>. 18th IFAC World Congress.
- Haddadin, S., De Luca, A., and Albu-Schäffer, A. (2017). Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6), 1292–1312. doi:10.1109/TRO.2017.2723903.
- Haddadin, S., Parusel, S., Johansmeier, L., Golz, S., Gabl, S., Walch, F., Sabaghian, M., Jähne, C., Hausperger, L., and Haddadin, S. (2022). The franka emika robot: A reference platform for robotics research and education. *IEEE Robotics Automation Magazine*, 29(2), 46–64. doi:10.1109/MRA.2021.3138382.
- Haddadin, S., Urbanek, H., Parusel, S., Burschka, D., Roßmann, J., Albu-Schäffer, A., and Hirzinger, G. (2010). Real-time reactive motion generation based on variable attractor dynamics and shaped velocities. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3109–3116. doi:10.1109/IROS.2010.5650246.
- Hamedani, M.H., Sadeghian, H., Zekri, M., Sheikholeslam, F., and Keshmiri, M. (2021). Intelligent Impedance Control using Wavelet Neural Network for dynamic contact force tracking in unknown varying environments. *Control Engineering Practice*, 113, 104840. doi:<https://doi.org/10.1016/j.conengprac.2021.104840>. URL <https://www.sciencedirect.com/science/article/pii/S0967066121001179>.
- He, W., Chen, Y., and Yin, Z. (2016). Adaptive neural network control of an uncertain robot with full-state constraints. *IEEE Transactions on Cybernetics*, 46(3), 620–629. doi:10.1109/TCYB.2015.2411285.
- Hirzinger, G., Albu-Schäffer, A.O., Hähne, M., Schäfer, I., and Sporer, N. (2001). On a new generation of torque controlled light-weight robots. *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 4, 3356–3363 vol.4.
- Hogan, N. (1984). Impedance control: An approach to manipulation. In *1984 American Control Conference*, 304–313. doi:10.23919/ACC.1984.4788393.
- Johansmeier, L., Gerchow, M., and Haddadin, S. (2019). A framework for robot manipulation: Skill formalism, meta learning and adaptive control. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May, 5844–5850. doi:10.1109/ICRA.2019.8793542.
- Karacan, K., Sadeghian, H., Kirschner, R.J., and Haddadin, S. (2022). Passivity-based skill motion learning in stiffness-adaptive unified force-impedance control. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, in publication. IEEE.
- Kato, Y., Balatti, P., Gandarias, J.M., Leonori, M., Tsuji, T., and Ajoudani, A. (2022). A self-tuning impedance-based interaction planner for robotic haptic exploration. *IEEE Robotics and Automation Letters*, 7(4), 9461–9468. doi:10.1109/LRA.2022.3190806.
- Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. Robotics Autom.*, 3, 43–53.
- Kirschner, R.J., Kurdas, A., Karacan, K., Junge, P., Birjandi, S., Mansfeld, N., Abdolshah, S., and Haddadin, S. (2021). Towards a reference framework for tactile robot performance and safety benchmarking. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4290–4297. IEEE.
- Kramberger, A., Gams, A., Nemeč, B., Schou, C., Chrysostomou, D., Madsen, O., and Ude, A. (2016). Transfer of contact skills to new environmental conditions. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 668–675. doi:10.1109/HUMANOIDS.2016.7803346.
- Kronander, K. and Billard, A. (2016). Stability considerations for variable impedance control. *IEEE Transactions on Robotics*, 32(5), 1298–1305. doi:10.1109/TRO.2016.2593492.
- Kulakov, F., Alferov, G.V., Efimova, P., Chernakova, S., and Shymanchuk, D. (2015). Modeling and control of robot manipulators with the constraints at the moving objects. In *2015 International Conference "Stability and Control Processes" in Memory of V.I. Zubov (SCP)*, 102–105. doi:10.1109/SCP.2015.7342075.
- Lepora, N.F. and Lloyd, J. (2020). Pose-based servo control with soft tactile sensing. *CoRR*, abs/2012.02504. URL <https://arxiv.org/abs/2012.02504>.
- Ott, C., Dietrich, A., and Albu-Schäffer, A. (2015). Prioritized multi-task compliance control of redundant manipulators. *Automatica*, 53, 416–423. doi:<https://doi.org/10.1016/j.automatica.2015.01.015>. URL <https://www.sciencedirect.com/science/article/pii/S0005109815000163>.
- Pastor, P., Kalakrishnan, M., Chitta, S., Theodorou, E., and Schaal, S. (2011). Skill learning and task outcome prediction for manipulation. In *2011 IEEE International Conference on Robotics and Automation*, 3828–3834. doi:10.1109/ICRA.2011.5980200.
- Pastor, P., Kalakrishnan, M., Righetti, L., and Schaal, S. (2012). Towards associative skill memories. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, 309–315. doi:10.1109/HUMANOIDS.2012.6651537.
- Qian, Y., Yuan, J., Bao, S., and Gao, L. (2019). Sensorless hybrid normal-force controller with surface prediction. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 83–88. doi:10.1109/ROBIO49542.2019.8961532.
- Schindlbeck, C. and Haddadin, S. (2015). Unified Passivity-Based Cartesian Force / Impedance Control for Rigid and Flexible Joint Robots via Task-Energy Tanks. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 440–447. doi:10.1109/ICRA.2015.7139036.
- Shahriari, E., Birjandi, S.A.B., and Haddadin, S. (2022). Passivity-based adaptive force-impedance control for modular multi-manual object manipulation. *IEEE Robotics Autom. Lett.*, 7(2), 2194–2201. doi:10.1109/LRA.2022.3142903. URL <https://doi.org/10.1109/LRA.2022.3142903>.
- Shahriari, E., Kramberger, A., Gams, A., Ude, A., and Haddadin, S. (2017). Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 136–142. IEEE.
- Simonič, M., Zlajpah, L., Ude, A., and Nemeč, B. (2019). Autonomous learning of assembly tasks from the corresponding disassembly tasks. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 230–236. doi:10.1109/Humanoids43949.2019.9035052.
- Yang, C., Ganesh, G., Haddadin, S., Parusel, S., Albu-Schaeffer, A., and Burdet, E. (2011). Human-like adaptation of force and impedance in stable and unstable interactions. *IEEE transactions on robotics*, 27(5), 918–930.

A.6 “A Passivity-based Approach on Relocating High-Frequency Robot Controller to the Edge Cloud”

This is the post-print version (author accepted manuscript) of the following publication: X. Chen, H. Sadeghian, L. Chen, M. Tröbinger, A. Swirkir, A. Naceri, and S. Haddadin. “A Passivity-based Approach on Relocating High-Frequency Robot Controller to the Edge Cloud”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 5242–5248.

A Passivity-based Approach on Relocating High-Frequency Robot Controller to the Edge Cloud

Xiao Chen[†], Hamid Sadeghian¹, Lingyun Chen², Mario Tröbinger, Abadalla Swirkir^{2,3},
Abdeldjalil Naceri and Sami Haddadin²

Abstract—As robots become more and more intelligent, the complexity of the algorithms behind them is increasing. Since these algorithms require high computation power from the onboard robot controller, the weight of the robot and energy consumption increases. A promising solution to tackle this issue is to relocate the expensive computation to the cloud. In this pioneering work, the possibility of relocating a state-of-the-art nonlinear control is investigated. To this end, the Unified Force-Impedance Controller (UFIC) is relocated to a remote location and high frequency feedback loop is established by including the remote controller in the loop. Passivity analysis is used to ensure the stability of the whole system, comprising the robot in interaction with the environment, the communication channel, as well as the remote controller. The instability associated with the communication channel is resolved by Time Domain Passivity Approach (TDPA). The performance of the proposed framework is experimentally evaluated on a robot arm in interaction with the environment. The results illustrate the stability of the system to a time-varying delay of up to $50 \pm 10ms$.

I. INTRODUCTION

The application of service robots is growing rapidly. Robots are proposed for new near-to-the-human applications every day. Working in dynamic and cluttered human environment demands fast and reliable sensing and advanced control capabilities. The robot needs to perceive the environment through many sensory data and sophisticated algorithms to plan its motion. This is especially the case for humanoid and mobile platforms, which require a lot of onboard computation on the robot controller, which in turn increases the power consumption. On the other hand, this increases the weight of the robot, reduces the agility and mobility of the system, and making the robot expensive. A promising solution is to offload expensive computations to the edge or cloud.

As the technology of cloud computing develops, Kuffner in [1] introduces the *cloud robotics* terminology for the first time. Since then, the idea of offloading computation to the cloud attracted a lot of attention. Algorithms that process large amount of data can benefit from cloud computing, e.g. collective learning algorithms [2], Convolutional Neural Network (CNN), vision-based applications [3], and motion planning in robotics [4]. Moreover, fast and more reliable

All authors are with the Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, Munich, Germany, ¹ also with Faculty of Engineering, University of Isfahan, 8174673441 Isfahan, Iran, ² and the Centre for Tactile Internet with Human-in-the-Loop (CeTI), ³ and with the Department of Electrical and Electronic Engineering, Omar Al-Mukhtar University (OMU), Albaida, Libya. [†] Corresponding Author (xiaoyu.chen@tum.de)

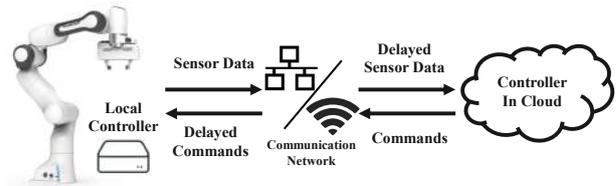


Fig. 1: Framework of relocating robot controller to the cloud.

wired or wireless communication protocols support this idea and make it feasible.

Relocating high-frequency control to the real-time cloud or edge cloud is a raising hot topic. Edge cloud aims to bring the cloud closer to end-users by moving the computation and storage closer to the network edge. The edge cloud provides a range of benefits, including reduced latency, improved performance, increased reliability, and reduced network congestion. [5], [6] show that under certain criteria, it is even feasible to run high-frequency real-time control loops on the cloud. The first edge-based whole-body control algorithm over a 5G wireless link was presented in [7].

A communication channel is non-passive due to delay [8], making a network-coupled robotic system unstable, especially when interacting with the environment. For a typical telepresence setup, the controller runs on the local computation unit with a direct connection to the robot to ensure system stability and performance [9]. The stability of a tactile robot in teleoperation under commercialized communication network like 5G, LTE, and WiFi has been investigated by the author of this paper in [10]. To ensure the passivity of a network-coupled system, an energy-based Time Domain Passivity Approach (TDPA) is applied to monitor the system energy in real-time using a Passivity Observer (PO) and adapt the controller gain accordingly using a Passivity Controller (PC) [11]. TDPA is a powerful method that can stabilize a system with communication delay and package loss [12]. However, it sacrifices the performance such as position tracking and transparent force feedback. Many algorithms are proposed to reduce the conservatism of TDPA [13] and increase the transparency of a teleoperation system [14].

In this work, we will investigate the possibility of relocating the low-level and high-frequency torque controller into the cloud. The overall system structure is illustrated in Fig. 1. In such a setting, the local controller provides sensory data, and the remote controller receives the data and sends commands back. Note that low-level control algorithm such

as Unified Force-Impedance Controllers (UFIC) [15] rely on a fast and real-time feedback control loop (usually 1kHz). This is different from high-level AI planning services or kinematic controllers, which provides/updates for instance a reference trajectory with a lower sampling rate, and thus hard real-time communication with the onboard robot control is usually not crucial. Here, we aim to address the following main questions:

- Can a tactile robot be controlled by running a low level nonlinear controller on a remote location?
- How is the stability and performance of the system in the presence of delay and packet loss in the communication channel?
- To what extent (entirely or partially), can the controller be offloaded? How is the structure of such a shared control?

Passivity analysis is utilized to establish a framework for such cloud-in-loop control. The general architecture of the system is designed to appear as the interconnection of passive elements, comprising the robot, environment, UFIC controller, and the two-ports communication channel. Because there is no robot dynamics on the cloud, the passivity analysis of UFIC need to be adapted. The quality of communication channel is crucial to the stability and performance of the system, and TDPA is used to ensure its passivity through PO/PC located right before and after this channel. Additionally, the energy tank [15] is used to ensure the passivity of the controller located on the remote side. The UFIC, in fact, only requires low computational power. However, we use it here to demonstrate the feasibility of this framework.

The rest of the paper is organized as follows. Section II describes the modeling of the robot and transmission channel. The main results of the paper are presented in Section III. The performance of the proposed framework is experimentally evaluated in Section IV. Finally, the paper is concluded in Section V.

II. PRELIMINARIES

A. Robot Model

The dynamic of a n Degrees of Freedom (DoF) robot manipulator in the joint space is given by

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_g + \boldsymbol{\tau}_{ext} + \boldsymbol{\tau}_c \quad (1)$$

where $\mathbf{q} \in \mathbb{R}^n$ is the joint state, $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ and $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ are the inertia and Coriolis/centrifugal matrices, and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is the gravitational torques. The control torque, external torque, and gravity compensation torques are denoted by $\boldsymbol{\tau}_c \in \mathbb{R}^n$, $\boldsymbol{\tau}_{ext} \in \mathbb{R}^n$ and $\boldsymbol{\tau}_g \in \mathbb{R}^n$, respectively.

On the right-hand side of Equation (1), It is assumed that the gravitational torques are compensated by the local controller, which has the full dynamic model of the robot on the local side. Thus, only $\boldsymbol{\tau}_c$ may need high computational power, which is computed in the cloud. Without loss of generality, the robot dynamics is rewritten as

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \boldsymbol{\tau}_{ext} + \boldsymbol{\tau}_c. \quad (2)$$

B. Unified Force Impedance Controller (UFIC)

The classical impedance and force control can be extended and unified into a single framework of unified force impedance control by composition of an impedance control torque $\boldsymbol{\tau}_{imp}$ with a force control torque $\boldsymbol{\tau}_{fc}$ [16],

$$\boldsymbol{\tau}_r = \boldsymbol{\tau}_{imp} + \boldsymbol{\tau}_{fc}. \quad (3)$$

The impedance controller ensures that the robot follows a desired trajectory with the following control law,

$$\boldsymbol{\tau}_{imp} = \mathbf{J}^T(\mathbf{q})(M_C(\mathbf{q})\ddot{\mathbf{x}}_{des} + C_C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{x}}_{des} + \mathbf{K}_x\tilde{\mathbf{x}} + \mathbf{D}_x\dot{\tilde{\mathbf{x}}}) \quad (4)$$

where $\tilde{\mathbf{x}} = \mathbf{x}_{des} - \mathbf{x}$ is the tracking error in task space \mathbf{x} with the desired trajectory \mathbf{x}_{des} . $\mathbf{J}(\mathbf{q})$ denotes the Jacobian matrix, \mathbf{K}_x and \mathbf{D}_x are the stiffness and damping matrices, and $M_C(\mathbf{q})$ and $C_C(\mathbf{q}, \dot{\mathbf{q}})$ are the robot inertia matrix and Coriolis/centrifugal matrix in Cartesian space.

The force control which regulate the interaction force \mathbf{f}_{ext} to following a desired force profile \mathbf{f}_{des} is given by

$$\boldsymbol{\tau}_{fc} = \mathbf{J}^T(\mathbf{q})(\mathbf{f}_{des} + \mathbf{K}_p(\mathbf{f}_{des} - \mathbf{f}_{ext}) + \mathbf{K}_d(\dot{\mathbf{f}}_{des} - \dot{\mathbf{f}}_{ext}) + \mathbf{K}_i\mathbf{h}_i(\mathbf{f}_{ext}, t)), \quad (5)$$

where $\mathbf{h}_i(\mathbf{f}_{ext}, t) := \int_0^t (\mathbf{f}_{des}(\sigma) - \mathbf{f}_{ext}(\sigma))d\sigma$. \mathbf{K}_p , \mathbf{K}_i and \mathbf{K}_d are the proportional, integral and derivative gains respectively. This controller is intuitive and easy to implement. However, the passivity of the closed-loop system is not preserved during interaction with the environment. An energy tank can be further designed and augmented to the system to ensure the passivity and thus stability of the system [16].

C. Energy Flow Analysis in Two-Port Network

Consider a two-port discrete network system N with ΔT as the sampling period, its energy flow is shown in Fig. 2. The system is passive if and only if at time instance k ,

$$E(0) + \Delta T \sum_{i=0}^k (\dot{\mathbf{q}}_1^T(i) \boldsymbol{\tau}_1(i) + \dot{\mathbf{q}}_2^T(i) \boldsymbol{\tau}_2(i)) \geq 0, \quad (6) \quad \forall k \geq 0$$

holds for velocities $\dot{\mathbf{q}}_j$ and torques $\boldsymbol{\tau}_j$, $j \in \{1, 2\}$ [11]. The product $\dot{\mathbf{q}}_j^T \boldsymbol{\tau}_j$ denotes the power flow at each port and it is denoted positive if energy flows into the network. $E(0)$ is the system initial energy storage and without loss of generality, we assume $E(0) = 0$.

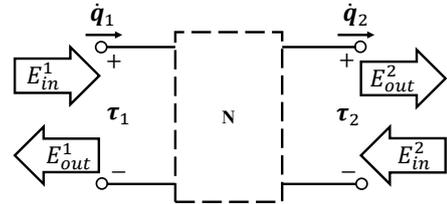


Fig. 2: Energy flow of two-port network. The output energy should be less than the input energy to ensure the network passivity.

In the TDPA, a Passivity Observer (PO) and Passivity Controller (PC) are used to ensure the system time-domain

passivity [12]. The PO observes the system energy in real-time while the PC dissipates the energy generated by the system. For a two-port system, the observed energy, E_{ob} , can be separated as input and output energy at each port, i.e.

$$\begin{aligned} E_{ob}(k) &= \Delta T \sum_{i=0}^k (\dot{\mathbf{q}}_1^T(i) \boldsymbol{\tau}_1(i) + \dot{\mathbf{q}}_2^T(i) \boldsymbol{\tau}_2(i)) \\ &= E_{in}^1(k) - E_{out}^1(k) + E_{in}^2(k) - E_{out}^2(k), \end{aligned} \quad (7)$$

where the superscripts represent the port and the subscripts show whether the energy goes *in* or *out* of the port. With the power flow defined as $P_1(k) = \dot{\mathbf{q}}_1^T(k) \boldsymbol{\tau}_1(k)$ and $P_2(k) = \dot{\mathbf{q}}_2^T(k) \boldsymbol{\tau}_2(k)$ and based on its sign, we can define the input and output powers at each port as

$$\begin{aligned} P_{in}^j(k) &= \begin{cases} P_j(k) & \text{if } P_j(k) > 0 \\ 0 & \text{otherwise,} \end{cases} \\ P_{out}^j(k) &= \begin{cases} -P_j(k) & \text{if } P_j(k) < 0 \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \quad (8)$$

for $j \in \{1, 2\}$. Hence, the input and output energy at each port can be calculated by integrating the power flow as follows,

$$\begin{aligned} E_{in}^j(k) &= \Delta T \sum_{i=1}^k P_{in}^j(i) \\ E_{out}^j(k) &= \Delta T \sum_{i=1}^k P_{out}^j(i). \end{aligned} \quad (9)$$

III. PROPOSED FRAMEWORK

A. General Structure

The general framework structure and data flow between the local controller and the remote controller is illustrated in Fig. 3. The local controller sends the *local* joint velocity $\dot{\mathbf{q}}_l$ and the external torque $\boldsymbol{\tau}_{ext}$ to the remote controller in the cloud. On the remote side, the data are received with a forward delay $d_f(t)$, i.e. $\dot{\mathbf{q}}_{ld}(t) = \dot{\mathbf{q}}_l(t - d_f(t))$ and $\boldsymbol{\tau}_{extd}(t) = \boldsymbol{\tau}_{ext}(t - d_f(t))$, $t \in \mathbb{R}$. After computation on the remote site, the *remote* command torque $\boldsymbol{\tau}_r$ is transferred back to the robot through the communication network with backward delay $d_b(t)$. The robot receives a delayed command torque $\boldsymbol{\tau}_{rd}(t) = \boldsymbol{\tau}_r(t - d_b(t))$.

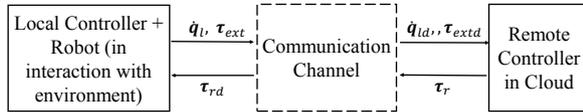


Fig. 3: Data flow between the local controller at the robot side and the remote controller. The communication network introduces time varying delay to the data.

Since in practice, the robot runs in high frequency with constant sampling time, the data transmission actually happens in discrete-time values, $\dot{\mathbf{q}}_{ld}(k) = \dot{\mathbf{q}}_l(k - d_f(k))$, $\boldsymbol{\tau}_{extd}(k) = \boldsymbol{\tau}_{ext}(k - d_f(k))$, $\boldsymbol{\tau}_{rd}(k) = \boldsymbol{\tau}_r(k - d_b(k))$, $k \in \mathbb{N}$.

Furthermore, the dynamical model of the robot is available in the cloud, in order to acquire the parameters like Cartesian position \mathbf{x} , inertia matrix $\mathbf{M}_C(\mathbf{q})$ and Coriolis/centrifugal matrix $\mathbf{C}_C(\mathbf{q}, \dot{\mathbf{q}})$ in Cartesian space. To obtain high accuracy in robot control, the dynamic model of the robot is identified a priori using the approach presented in [17].

B. Passivity Framework for Analysis of the System

The whole system comprising robot/environment, remote controller and the communication channel is arranged in four subsystems in feedback loop interconnection as illustrated in Fig. 4. This preserves the passivity of the entire structure while considering the passivity of each subsystem. It is assumed that the robot is in interaction with a passive environment meaning that the environment block is passive w.r.t. input-output pair $(\dot{\mathbf{q}}_l, -\boldsymbol{\tau}_{ext})$. Additionally, it is easy to show that the robot dynamics (2) is passive w.r.t. $(\boldsymbol{\tau}_l + \boldsymbol{\tau}_{ext}, \dot{\mathbf{q}}_l)$. Therefore, in this section, the passivity of the UFIC with augmented energy tank w.r.t. $(-\dot{\mathbf{q}}_r, -\boldsymbol{\tau}_r)$ is established. Here, the $\dot{\mathbf{q}}_r$ is the modified $\dot{\mathbf{q}}_{ld}$, which will be presented in next section. Furthermore, the passivity of Communication Channel combined with time domain PO/PC will also be shown.

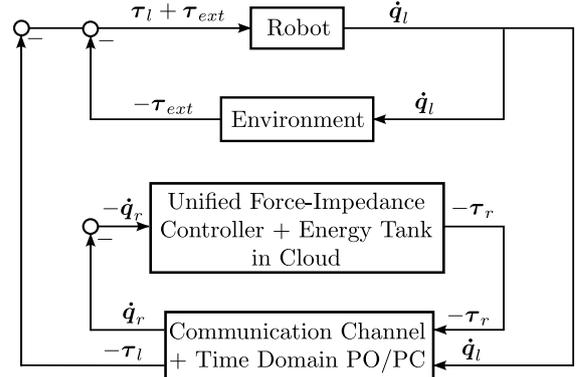


Fig. 4: The block diagram of the system in feedback interconnection of passive elements. The robot in interaction with the environment, as well as the controller, are depicted as one port passive system. The communication channel is considered as two-port passive system.

First, consider the UFIC that is relocated in the cloud, with the storage function defined as $S = \frac{1}{2} \tilde{\mathbf{x}}^T \mathbf{K}_x \tilde{\mathbf{x}}$. To show the passivity of the controller, we need to show that its derivative $\dot{S} \leq \dot{\mathbf{q}}_r^T \boldsymbol{\tau}_r$. By exploiting (3) and considering the regulation case, i.e. $\dot{\mathbf{x}}_{des} = \ddot{\mathbf{x}}_{des} = 0$, the following is given

$$\begin{aligned} \dot{S} &= \dot{\mathbf{x}}^T \mathbf{K}_x \tilde{\mathbf{x}} \\ &= \dot{\mathbf{q}}_r^T \mathbf{J}^T \mathbf{K}_x \tilde{\mathbf{x}} \\ &= \dot{\mathbf{q}}_r^T (\boldsymbol{\tau}_r - \mathbf{J}^T \mathbf{D}_x \dot{\mathbf{x}} - \mathbf{J}^T \mathbf{K}_p (\mathbf{f}_{des} - \mathbf{f}_{extd}) \\ &\quad - \mathbf{J}^T \mathbf{K}_d (\dot{\mathbf{f}}_{des} - \dot{\mathbf{f}}_{extd}) - \mathbf{J}^T \mathbf{h}_i(\mathbf{f}_{extd}, t)) \\ &= (-\dot{\mathbf{q}}_r^T) (-\boldsymbol{\tau}_r) - \dot{\mathbf{x}}^T \mathbf{D}_x \dot{\mathbf{x}} - \dot{\mathbf{x}}^T (\mathbf{K}_p (\mathbf{f}_{des} - \mathbf{f}_{extd}) \\ &\quad + \mathbf{K}_d (\dot{\mathbf{f}}_{des} - \dot{\mathbf{f}}_{extd}) + \mathbf{K}_i \mathbf{h}_i(\mathbf{f}_{extd}, t)). \end{aligned} \quad (10)$$

Where $\dot{\mathbf{x}} = \mathbf{J}^T \dot{\mathbf{q}}_r$. Since the sign of the last term is not clear, the passivity of Unified Force-Impedance Controller

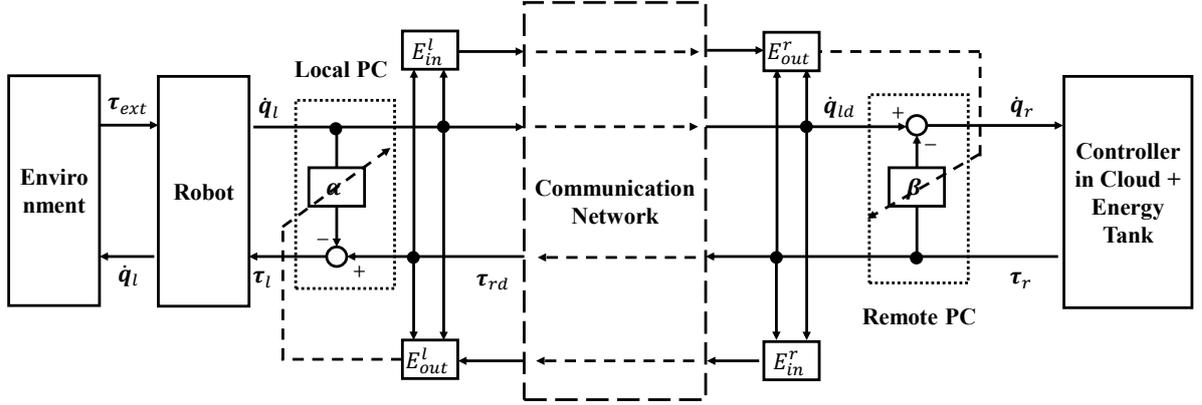


Fig. 5: Diagram of proposed framework with two-port time domain PO/PC

w.r.t $(-\dot{q}_r, -\tau_r)$ is not guaranteed.

To show the passivity of the controller block, an energy tank based method is introduced to modify the control law. Let x_t and $T = \frac{1}{2}x_t^2$ be the tank state and energy, respectively. Also define the tank dynamics as

$$\dot{x}_t = \frac{\zeta}{x_t} (\dot{x}^T D_x \dot{x} + \gamma \dot{x}^T K_d (\dot{f}_{des} - \dot{f}_{extd})) + u_t, \quad (11)$$

where ζ is defined as

$$\zeta = \begin{cases} 1 & \text{if } T \leq T_{max} \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

This indicates that when tank energy exceeds the allowed maximum energy T_{max} , no further energy will be loaded into the tank. The control input of the tank is written as $u_t = -\omega^T \dot{x}$, with

$$\omega = \frac{\eta}{x_t} (K_p (\dot{f}_{des} - \dot{f}_{extd}) + (1 - \gamma) K_d (\dot{f}_{des} - \dot{f}_{extd}) - K_i h_i (\dot{f}_{extd}, t)). \quad (13)$$

By defining η as

$$\eta = \begin{cases} 1 & \text{if } T \geq T_{min} \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

we can ensure that when the tank energy reaches the allowed minimum energy T_{min} , the force-impedance controller is detached from the energy tank.

The parameter γ is defined to eliminate the negative value and ensure the system passivity,

$$\gamma = \begin{cases} 1 & \text{if } \dot{x}^T K_d (\dot{f}_{des} - \dot{f}_{extd}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

The original force-impedance controller (3) is now updated with the energy tank as,

$$\tau_r = J^T(q_r) (K_x \tilde{x} + D_x \dot{x} + M_C(q_r) \ddot{x}_{des} + C_C(q_r, \dot{q}_r) \dot{x}_{des} + \gamma K_d (\dot{f}_{des} - \dot{f}_{extd}) - \omega x_t) \quad (16)$$

The following theorem is given.

Theorem 1: The controller given in (16) realizes a passive mapping with respect to input-output pair $(-\dot{q}_r, -\tau_r)$.

Proof: Consider the storage function defined as $S = \frac{1}{2} \tilde{x}^T K_x \tilde{x} + \frac{1}{2} x_t^2$. The derivative of this function is derived as follows,

$$\begin{aligned} \dot{S} &= \dot{x}^T K_x \tilde{x} + x_t \dot{x}_t \\ &= \dot{q}_r^T J^T K_x \tilde{x} + x_t \dot{x}_t \\ &= \dot{q}_r^T (\tau_r - J^T D_x \dot{x} - J^T \gamma K_d (\dot{f}_{des} - \dot{f}_{extd}) + J^T \omega x_t) + x_t \dot{x}_t \\ &= \dot{q}_r^T \tau_r - \dot{x}^T (D_x \dot{x} + \gamma K_d (\dot{f}_{des} - \dot{f}_{extd}) - \omega x_t) + x_t \dot{x}_t \\ &= \dot{q}_r^T \tau_r - \dot{x}^T (D_x \dot{x} + \gamma K_d (\dot{f}_{des} - \dot{f}_{extd}) - \omega x_t) + \zeta (\dot{x}^T D_x \dot{x} + \gamma \dot{x}^T K_d (\dot{f}_{des} - \dot{f}_{extd})) - x_t \omega^T \dot{x} \\ &= \dot{q}_r^T \tau_r - (1 - \zeta) (\dot{x}^T D_x \dot{x} + \gamma \dot{x}^T K_d (\dot{f}_{des} - \dot{f}_{extd})) \\ &\leq \dot{q}_r^T \tau_r \\ &= (-\dot{q}_r^T) (-\tau_r) \end{aligned} \quad (17)$$

Hence, the block is passive with respect to $(-\dot{q}_r, -\tau_r)$. ■

C. Two-Port TDPA for Communication Channel

The communication channel can be seen as a two-port network. The energy of the communication channel is observed by PO as

$$E_{CC}(k) = E_{in}^l(k) + E_{in}^r(k) - E_{out}^l(k) - E_{out}^r(k) \quad (18)$$

where superscript l indicates the *local side* and r indicates the *remote side*.

In real-world communication network (wired or wireless), problems such as time-varying delay and package loss are inevitable. Therefore, it is impossible to observe the energy at both side at the same time instance. The above issues are the source of instability that is imposed by communication channel and makes the coupled robot system unstable, especially when the robot is interacting with an environment.

Considering the network delay and package loss, and noting that each input energy is monotonous, the sufficient

condition of the system passivity (6) can be written as,

$$\begin{aligned} E_{in}^l(k - d_f(k) - l_f(k)) &\geq E_{in}^l(k) \geq E_{out}^r(k) \\ E_{in}^r(k - d_b(k) - l_b(k)) &\geq E_{in}^r(k) \geq E_{out}^l(k) \end{aligned} \quad (19)$$

where l_f and l_b are the package loss in forward and backward communication. The PC modifies the torque or velocity to dissipate extra energy. The energy needs to be dissipated at sample time k is calculated as follows,

$$\begin{aligned} E_{PC}^l(k) &= E_{in}^l(k - d_f(k) - l_f(k)) - E_{out}^r(k) - E_{diss}^l(k-1) \\ E_{PC}^r(k) &= E_{in}^r(k - d_b(k) - l_b(k)) - E_{out}^l(k) - E_{diss}^r(k-1) \end{aligned} \quad (20)$$

The already dissipated energy by the PCs, i.e., $E_{diss}^l(k)$ and $E_{diss}^r(k)$ are given by

$$\begin{aligned} E_{diss}^l(k) &= \Delta T \sum_{i=1}^{k-1} \|\tau_r(i)\|^2 \beta(i), \\ E_{diss}^r(k) &= \Delta T \sum_{i=1}^{k-1} \|\dot{q}_i(i)\|^2 \alpha(i). \end{aligned} \quad (21)$$

In which the damping element α and β are calculated such that, the condition (19) is satisfied, i.e.,

$$\beta(k) = \begin{cases} \frac{-E_{PC}^r(k)}{\Delta T \|\tau_r(k)\|^2} & \text{if } E_{PC}^r(k) < 0 \\ & \text{and } \tau_r(k) \neq 0 \\ 0 & \text{else} \end{cases} \quad (22)$$

$$\alpha(k) = \begin{cases} \frac{-E_{PC}^l(k)}{\Delta T \|\dot{q}_i(k)\|^2} & \text{if } E_{PC}^l(k) < 0 \\ & \text{and } \dot{q}_i(k) \neq 0 \\ 0 & \text{else} \end{cases} \quad (23)$$

The actual joint velocity $\dot{q}_r(k)$ received by the cloud and the command torque $\tau_i(k)$ received by the robot are calculated based on following PC laws,

$$\begin{aligned} \dot{q}_r(k) &= \dot{q}_{id}(k) - \beta(k)\tau_r(k) \\ \tau_i(k) &= \tau_{rd}(k) - \alpha(k)\dot{q}_i(k) \end{aligned} \quad (24)$$

The overall structure of the system including TDPA implementation is depicted in Fig. 5.

The above analysis shows that the Communication Channel with Time Domain PC/PO as illustrated in Fig. 4 is passive with respect to its input and output port. Thus, based on the results from Section III-B, all the subsystems are passive, and thus the entire system is passive.

IV. EXPERIMENTAL EVALUATION

A. Experiment Setup

The following experiments serve as the performance evaluation of the proposed control framework. The experiment setup is shown in Fig. 6. Experiments are conducted with a 7-DoF Franka Emika robot [18]. The local and cloud controller run on separate PCs with the same hardware configuration (Intel Core i7-10700 CPU @ 2.90GHz) and operating system (Ubuntu 20.04 LTS with real-time kernel). The PCs are connected via Ethernet and located in the same subnet.

An open source tool called tcgui [19] which utilizes the Linux kernel's network traffic control and shaping features

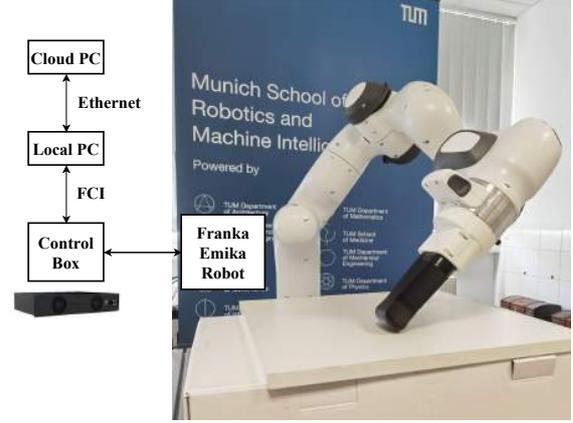


Fig. 6: Experiment setup with Franka Emika Robot; The robot is following a trajectory on the surface plane while regulating the force in the normal direction.

is used to simulate the real-world network condition with variable delay and package loss.

The local PC controls the robot via Franka Control Interface (FCI) at 1 kHz rate. The proposed control framework offloads the UFIC which also runs at 1 kHz rate into the cloud PC. The local PC controls the robot by sending joint torque commands to FCI based on the delayed desired joint torque commands received from the cloud PC. Meanwhile, the robot state acquired from FCI is sent back to the cloud PC. The bi-directional network communication channel between the local and cloud PCs is established using User Datagram Protocol (UDP).

The target of the force-impedance controller is set to follow a circular trajectory with a diameter of 10cm and a period of 5s in x-y plane and apply a force of 10N on the z-axis. The stiffness K_x and damping D_x parameters of the impedance control are set to $diag[400, 400, 400, 1, 1, 1]N/m$ and $diag[10, 10, 10, 0.5, 0.5, 0.5]N \cdot m/s$ respectively. The K_p and K_i for the force controller are set to $0.5I$ and $0.02I$, where I is identity matrix. The setup is tuned at the verge of stability at $50 \pm 10ms$ roundtrip-delay, meanwhile guarantee a good force tracking performance. The network delay parameters from 0 to 50ms with a 10ms interval and 20% variance are set through tcgui. The force measurement is derived from robot joint torque sensors based on the external force estimation method [20].

B. Results

The energy plot in Fig. 7 illustrate the passivity of the communication channel. E_{CC} given in (18) is always greater than zero in all delay scenarios. The accumulated energy also indicates the energy conservatism of the TDPA, as well as the energy tank.

The trajectory of the end-effector and the tracking error are shown in Fig. 8. It is clear that the proposed control framework is capable of controlling the robot to follow the desired trajectory p_{des} . Tracking performance deteriorates with increase of latency. It is obvious when comparing with the error without any delay ep_{std} .

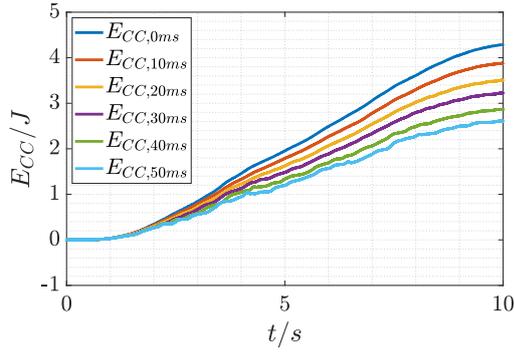


Fig. 7: The observed energy (18) in the communication channel, the positive energy indicates the communication channel is always passive.

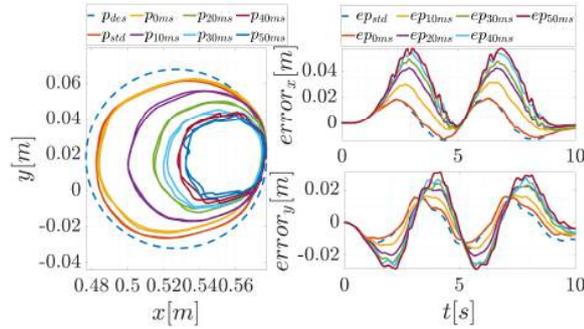


Fig. 8: Position tracking performance. The figure on the left shows the trajectory and figures on the right shows the tracking error in x and y direction. The tracking performance is better with less delay.

The force tracking performance is shown in Fig. 9. With different delay parameters, the proposed framework is able to control the robot to apply desired force profile f_{des} . The force tracking performance also slightly deteriorates with increasing latency (Fig. 10).

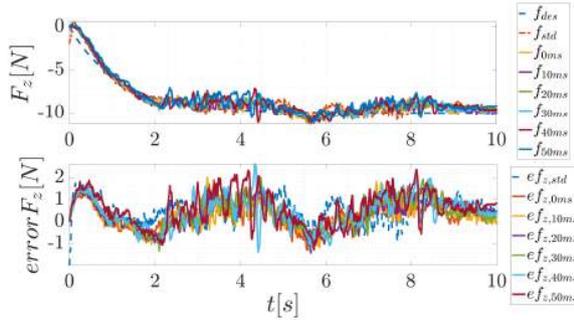


Fig. 9: Force tracking performance. The recorded force is shown in upper plot and the lower plot shows the tracking error.

C. Discussion

Our experiments shows only up to $50 \pm 10ms$ delay, which is at the scale of wired communication at distance of $7000km$ or currently available commercialized 5G network. However, the affordable delay can go higher by tuning the controller parameters.

As the delay increases, the control bandwidth decreases. To increase the total control bandwidth, it is possible to only relocate the computational heavy part of the controller,

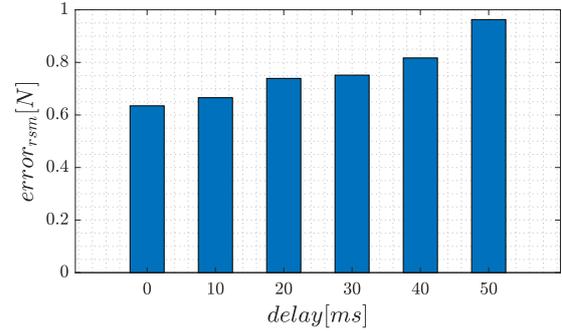


Fig. 10: Root mean square error of force tracking under different delays. The error increases as delay increases.

meanwhile the computationally light weight part of that can stay with the local robot. As an example, in this proposed framework, the gravity compensation, which can be computed in short time, is located on the local controller side. In this way, even without receiving any control command, the robot still maintains its basic function.

V. CONCLUSIONS

This article presents a novel framework for relocating computationally intensive controllers to the cloud in order to alleviate the burden on local robots. Our experimental results demonstrate the feasibility of offloading a state-of-the-art nonlinear controller to the cloud for controlling a tactile robot. The proposed passivity-based framework ensures the stability of the system. To maintain passivity in the presence of time-varying communication delays, methods such as the energy tank and TDPA are implemented. Future research will focus on reducing the conservatism of the TDPA to improve tracking performance, as well as exploring how to allocate different components of the controller to the local and remote sides.

ACKNOWLEDGEMENT

This work was funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (BMBF) within the project VERITAS under the Project Number 01IS18073A and also with the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy EXC 2050/1 Project ID 390696704 Cluster of Excellence Centre for Tactile Internet with Human-in-the-Loop (CeTI) of Technische Universität Dresden. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany (BMBF) in the programme of "Souverän. Digital. Vernetzt." Joint project 6G-life, project identification number 16KISK002. We gratefully acknowledge the funding of the Lighthouse Initiative Geriatrics by StMWi Bayern (Project X, grant no. IUK-1807-0007// IUK582/001) and LongLeif GaPa gGmbH (Project Y). We also gratefully acknowledge the funding by the European Union's Horizon 2020 research and innovation program as part of the project ReconCycle under grant no. 871352. Please note that S. Haddadin has a potential conflict of interest as shareholder of Franka Emika GmbH.

REFERENCES

- [1] J. KUFFNER, "Cloud-enabled humanoid robots," *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on, Nashville TN, United States, Dec.*, 2010.
- [2] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.
- [3] Z. Kamranian, H. Sadeghian, A. R. Naghsh Nilchi, and M. Mehrandezh, "Fast, yet robust end-to-end camera pose estimation for robotic applications," *Applied Intelligence*, vol. 51, no. 6, pp. 3581–3599, 2021.
- [4] A. Vick, V. Vonásek, R. Pěnička, and J. Krüger, "Robot control as a service—towards cloud-based motion planning and control for industrial robots," in *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*. IEEE, 2015, pp. 33–39.
- [5] D. M. Lofaro, A. Asokan, and E. M. Roderick, "Feasibility of cloud enabled humanoid robots: Development of low latency geographically adjacent real-time cloud control," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 519–526.
- [6] D. M. Lofaro and A. Asokan, "Low latency bounty hunting and geographically adjacent server configuration for real-time cloud control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5277–5282.
- [7] H. Zhu, M. Sharma, K. Pfeiffer, M. Mezzavilla, J. Shen, S. Rangan, and L. Righetti, "Enabling remote whole-body control with 5g edge computing," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3553–3560.
- [8] R. Anderson and M. Spong, "Bilateral control of teleoperators with time delay," *IEEE Transactions on Automatic Control*, vol. 34, no. 5, pp. 494–501, 1989.
- [9] L. Chen, A. Swikir, and S. Haddadin, "Drawing elon musk: A robot avatar for remote manipulation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4244–4251.
- [10] X. Chen, L. Johannsmeier, H. Sadeghian, E. Shahriari, M. Danneberg, A. Nicklas, F. Wu, G. Fettweis, and S. Haddadin, "On the communication channel in bilateral teleoperation: An experimental study for ethernet, wifi, lte and 5g," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, p. in progress.
- [11] B. Hannaford and J.-H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002.
- [12] J.-H. Ryu, J. Artigas, and C. Preusche, "A passive bilateral control scheme for a teleoperator with time-varying communication delay," *Mechatronics*, vol. 20, no. 7, pp. 812–823, 2010, special Issue on Design and Control Methodologies in Telerobotics.
- [13] M. Panzirsch, J.-H. Ryu, and M. Ferre, "Reducing the conservatism of the time domain passivity approach through consideration of energy reflection in delayed coupled network systems," *Mechatronics*, vol. 58, pp. 58–69, 2019.
- [14] H. Singh, A. Jafari, and J.-H. Ryu, "Enhancing the force transparency of time domain passivity approach: Observer-based gradient controller," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1583–1589.
- [15] C. Schindlbeck and S. Haddadin, "Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 440–447.
- [16] E. Shahriari, L. Johannsmeier, E. Jensen, and S. Haddadin, "Power flow regulation, adaptation, and learning for intrinsically robust virtual energy tanks," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 211–218, 2019.
- [17] A. Jubien, M. Gautier, and A. Janot, "Dynamic identification of the kuka lwr robot using motor torques and joint torque sensors data," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 8391–8396, 2014.
- [18] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jaehne, L. Hausperger, and S. Haddadin, "The franka emika robot: A reference platform for robotics research and education," *IEEE Robotics & Automation Magazine*, 2022.
- [19] B. Heß, C. Sieber, M. Hofbauer, and TUM LKN team, "tcgui: A lightweight Python-based Web-GUI for Linux traffic control," 7 2016.
- [20] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 1623–1630.

A.7 “Adaptive Robotic Levering for Recycling Tasks”

This is the author’s manuscript of the following publication: B. Kuster, M. Simonič, and A. Ude. “Adaptive Robotic Levering for Recycling Tasks”. In: *Advances in Service and Industrial Robotics: Proceedings of the 32nd International Conference on Robotics in Alpe-Adria Danube Region (RAAD 2023)*. Springer, 2023, pp. 417–425

Adaptive Robotic Levering for Recycling Tasks

Boris Kuster^{1,2}, Mihael Simonič¹, and Aleš Ude¹

¹ Humanoid and Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia

² Jožef Stefan International Postgraduate School, Ljubljana, Slovenia

Abstract. A common step in autonomous robotic disassembly (recycling) of electronics is levering, which allows the robot to apply greater forces when removing parts of the devices. In practical applications, the robot should be able to adapt a levering action to different device types without an operator specifically recording a trajectory for each device. A method to generalize the existing levering actions to new devices is thus needed. In this paper we present a parameterized algorithm for performing robotic levering using feedback-based control to determine contact points and a sinusoidal pattern to realize adaptive levering motion. The algorithm can deal with devices of different shapes. After the initial adaptation process, the subsequent executions of the learnt levering action can be sped up to improve performance.

Keywords: Robotic disassembly · Levering · Force control.

1 Introduction

Recycling faces the problem of small batch sizes and large variety of recycled items [1,2]. In such circumstances, the effort of robot programming to perform autonomous disassembly of generic electronics is one of the main reasons for the slow deployment of robotic-based solutions.

During the disassembly of electronics and other items, various robotic skills are needed, one of which is levering. Levering is a process whereby mechanical advantage can be gained using a rigid beam (lever) and a fixed hinge (fulcrum), which allows a greater force to be exerted on the load (the levered object). Some operations where levering is needed include removing pins and nails or separating different device parts. While this is often an easy task for humans as they can rely on vision, force, and pressure sensing supported by previous experience and generalisation capabilities, robotic disassembly applications still commonly use pre-recorded trajectories obtained by learning from demonstration (LfD) [3]. These trajectories, however, cannot be applied to different different electronic devices without a properly implemented adaptation process.

To perform a generalized levering action, we encode it with a sinusoidal pattern. The execution is controlled by monitoring the external forces and torques acting on the end-effector. One of the benefits of the proposed framework is that adaptation can be performed even with noisy vision information, which can

occur due to the poor lighting conditions and occlusions often encountered in recycling processes [4]. The proposed algorithm for adaptive levering was implemented on a collaborative Franka Emika Panda robot within a modular robotic workcell [5] and applied to disassembling heat cost allocators (electronic devices shown in Fig. 2).

The paper is organized as follows: in Section 2.1, the levering setup is presented. Section 2.2 describes a search algorithm for automatically detecting contact points which must be known in order to perform a levering action. In Section 2.3 we discuss the application of the sinusoidal pattern to encode a levering action. An algorithm for detecting the completion of the levering task based on force-torque measurements is also presented. In Section 2.4, the adaptation of the levering movement is explained. The experimental evaluation of our approach is presented in Section 3. We conclude with a critical discussion and plans to improve the proposed algorithm.

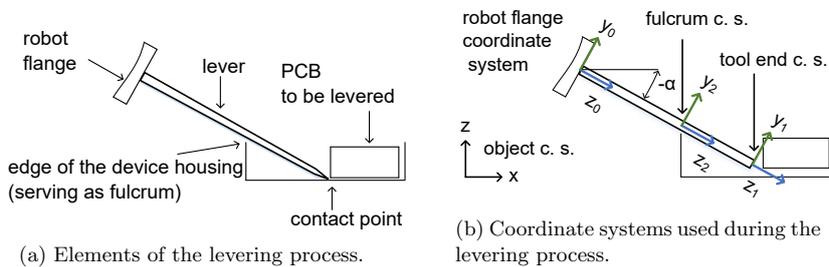
2 Methodology

The proposed adaptive levering procedure is based on our knowledge about the geometry of the task.

2.1 Levering setup

Fig. 1a shows a typical levering setup, where the lever is attached to the robot's flange. In the following the terms lever and tool are used interchangeably. In our system, the tool is integrated into qbRobotics Variable Stiffness Gripper. Subsequently, we refer to the levered object (in our experiments a printed circuit board) as part. To increase mechanical advantage, the lever is positioned against the fulcrum.

In Fig. 1b, the object coordinate system (c.s.) (x, y, z axes), the robot flange c.s. (x_0, y_0, z_0), the tool end c.s. (x_1, y_1, z_1) and the fulcrum c.s. (x_2, y_2, z_2) are shown. The Tool Center Point (TCP) coincides with the origin of tool end c.s. The fulcrum c.s. is estimated once the lever establishes a contact with the edge of the device housing. In our work, we make use of force-torque (FT) measurements ($F_{x0}, F_{y0}, F_{z0}, M_{x0}, M_{y0}, M_{z0}$), which are calculated in the robot flange c.s.



To transform the forces from the flange to the object coordinate system, we define the rotational matrix between flange and object c.s. as seen in Eq. (24). The vector of forces $\mathbf{f}_{obj} = [F_x, F_y, F_z]$ in object c.s. can be estimated as follows

$$\mathbf{f}_{obj} = \mathbf{R}_{flange_to_obj} \mathbf{f}_{flange}, \quad (1)$$

$$\mathbf{R}_{flange_to_obj} = \mathbf{R}_{flange}^\top \mathbf{R}_{obj}, \quad (2)$$

where $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ is a rotation matrix. The torques are not transformed to the object c.s. since the $M_{x,0}$ measurements are used, which are calculated in the robot flange c.s. To map the robot movements from the robot base c.s. to the object c.s., we define the rotational transformation between them as

$$\dot{\mathbf{p}}_{base} = \mathbf{R}_{base_to_obj} \dot{\mathbf{p}}_{obj}, \quad (3)$$

$$\mathbf{R}_{base_to_obj} = \mathbf{R}_{base}^\top \mathbf{R}_{obj}. \quad (4)$$

$$\mathbf{p}^{base} = \mathbf{p}_{base_to_obj} + \mathbf{R}_{base_to_obj} base_{obj}^P. \quad (5)$$

$$\mathbf{T}_{obj}^{base}. \quad (6)$$

$$\tau_{desired} = \tau_{task} + \tau_{nullspace} + \tau_{coriolis} + \tau_{added_FT} \quad (7)$$

$$\tau_{task} = \mathbf{J}^\top (-\mathbf{K}\mathbf{e}_{pos} - \mathbf{D}\mathbf{e}_{velocity}) \quad (8)$$

$$\tau_{added_FT} = \mathbf{J}^\top \mathbf{f}_{added} \quad (9)$$

$$(10)$$

$$\mathbf{T}_{pcb}^{obj} = \mathbf{T}_{obj}^{-1} \mathbf{T}_{pcb} \quad (11)$$

$$\mathbf{T}_{final}^{pcb} = \mathbf{T}_{pcb}^{-1} \mathbf{T}_{final} \quad (12)$$

$$(13)$$

$$\mathbf{T}_{pcb}^{base} \quad (14)$$

$$(15)$$

$$\mathbf{T}_{final}^{base} \quad (16)$$

$$(17)$$

$$(18)$$

$$\mathbf{f}_{obj} = \mathbf{R}_{obj}^{flange} \mathbf{f}_{flange}, \quad (19)$$

$$\mathbf{R}_{obj}^{flange} = \mathbf{R}_{flange}^\top \mathbf{R}_{obj}, \quad \dot{\mathbf{p}}_{base} = \mathbf{R}_{base_to_obj} \dot{\mathbf{p}}_{obj} \quad (20)$$

$$\mathbf{R}_{base_to_obj} = \mathbf{R}_{base}^\top \mathbf{R}_{obj}. \quad (21)$$

$$\mathbf{p}_{base} = \mathbf{p}_{base_to_obj} + \mathbf{R}_{base_to_obj} \mathbf{p}_{obj}, \quad (22)$$

$$\mathbf{T}_{obj}^{base}. \quad (23)$$

$$\mathbf{p}(t + t_{samp}) = \mathbf{p}(t) + v t_{samp} \mathbf{R}_{base_to_obj} \mathbf{d}\mathbf{p}^T \quad (24)$$

2.2 Fulcrum and part contact point search algorithm

To perform a levering action, the robot must first place the lever into the gap between the object and the fulcrum. A safe initial position is in the middle of the

gap. We detect the gap by using 3-D vision. Examples of devices and the gaps for levering are shown in Fig. 2, where the gaps are marked with a red line. For some devices, the gaps are quite large, while for others, the gap is very narrow and challenging to detect.

While vision is sufficiently accurate to approximately position the tool in the middle of the gap, it is not possible to place the lever at the object based on vision results only. To establish contact, the robot first moves to a fixed height above the middle of the gap and then starts moving in the negative z direction until it hits the bottom of the device housing. The motion is stopped once the measured force F_z at the tool center point in vertical direction exceeds a predefined threshold F_{max} . The force-torque estimation is performed by the Franka Emika robot control system using internal joint torque sensors. Based on the half length of the gap, we determine also the initial inclination of the lever. A minimal threshold distance $d_{z,min}$ is set beforehand. If it is not exceeded before detecting contact, we consider that the initial position for levering (the gap) was incorrectly determined. In this case, we select another initial position with stochastic search. It is implemented by adding a small random vector ε to the previous starting point

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix}, \quad (25)$$

where the mean of the random vector is set to zero, while the standard deviation is determined as a fraction of the half gap width of a particular device type (the red line in Fig. 2).

In the next step, the robot moves along the positive x axis in the object c.s. until the horizontal force F_x exceeds the force threshold F_{max} . At this point the lever (tool) is in contact with the PCB to be levered out of the device. Next the robot determines the fulcrum position by performing a rotational motion around the x axis of the tool end c.s. (positioned at the tip of the lever). The motion is stopped once the torque $M_{x,0}$ exceeds the threshold M_{max} , which signifies that the contact between the lever and the fulcrum has been established.



Fig. 2: Example electronic devices that need to be dismantled. Levering is used to remove the PCB from the housing.

2.3 Adaptive levering algorithm

The robot can now proceed with levering out the PCB. However, it is not easy to manually program a general levering action because the required forces and amplitude of motion that need to be applied to pry out the PCB cannot be analytically determined in advance because they depend on the geometry of the device and its current state of damage. We have therefore developed an adaptation algorithm that modifies the generic levering action so that it becomes suitable for the current device that needs to be dismantled.

Humans often use periodic movements when levering, especially when they do not know the force required to dislodge an object with the lever. In doing so, they slightly increase the force on the lever in each period.

We generate a single degree-of-freedom (DOF) sinusoidal cycle with an amplitude A and cycle time t_c , which represents the angle of the end-effector relative to the initial angle α at which the robot is in contact with the fulcrum and part. This results in the following trajectory around the x -axis of the fulcrum c.s.

$$\varphi_{x,2}(t) = g(t) + A \sin\left(\frac{2\pi t}{t_c}\right). \quad (26)$$

The other two angles around fulcrum coordinate axes are set to $\varphi_{y,2}(t) = \varphi_{z,2}(t) = 0$. The levering algorithm is parameterized with the following parameters:

- initial amplitude of the sinusoidal cycle, set to $A = 10^\circ$ in our experiments.
- duration of the sinusoidal cycle, set to $t_c = 3$ s.
- offset increment, set to $g(t) = 0.1At/t_c$.

When the lever is in the initial position, touching both the PCB and the fulcrum, the execution of the sinusoidal pattern begins. An example of a three period sinusoidal pattern with a constant amplitude is shown in Fig. 3.

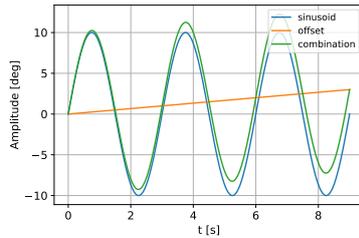


Fig. 3: Three periods of a sinusoid with an increasing offset g .

During levering, the lever must remain in contact with the part. Velocity-resolved control is used to ensure that a desired constant force on the part is maintained [6]. While the angle of the tool is determined by the sinusoidal pattern, the commanded velocity in the x_{obj} direction is determined based on a desired preset contact force F_d , the actual current contact force $F_x(t)$ and the

coefficient k_F . The commanded position in the x direction of the object c.s. is computed by integration

$$\dot{x}(t) = k_F(F_d - F_x(t)), \quad (27)$$

$$x(t + \Delta t) = x_0 + \dot{x}(t)\Delta t. \quad (28)$$

The other two coordinates remain constant during the execution of the leveringing movement, i.e. $y = y_0$, $z = z_0$.

The goal of leveringing is to pry out the PCB out of the device housing. We therefore let the leveringing out trajectory (3) running until the success signal (29) reaches the threshold. Given the time t at which the leveringing action was successful, we record the amplitude as $g(t) + A$. Thus next time we can start the leveringing action using this amplitude instead of starting from A and gradually increasing the amplitude. Note that we cannot just start with a very large amplitude as this could cause unsafe robot movements or uncontrolled motion of the PCB being levered out. The above adaptation procedure has turned out to be sufficient for our application. If needed more advanced learning algorithms [7] could be used to optimize the learnt leveringing out behavior.

We can use force-torque measurements $[F_{x,0}, F_{y,0}, F_{z,0}, M_{x,0}, M_{y,0}, M_{z,0}]^T$ at the robot's flange to detect when the leveringing action succeeded at prying out the PCB. To determine when the PCB (part) has been pried out, we monitor the torque $M_{x,0}$ measured at the robot's flange. During leveringing the torque increases, while a sharp torque drop-off is observed at the moment when the PCB is dislodged. The leveringing is successful if the difference between the maximum and minimum torque value within the width of the signal observation window is greater than a prespecified threshold M_{max}

$$\max_{t \in W} \{M_{x,0}(t)\} - \min_{t \in W} \{M_{x,0}(t)\} > M_{max}, \quad (29)$$

where $W = \{t_k - t_w, \dots, t_k\}$, t_k denotes the current sample time, and $t_w = 1.5s$ is the size of the sliding window. This condition can only be triggered while the lever angle α is decreasing, meaning it's applying a force to the part. The size of the sliding window is constant for all device types. The value of the threshold M_{max} is constant for all devices on which we tested the algorithm, however for novel device types it might require tuning.

When recycling old electronics, devices can be in various states of damage. It can sometimes happen that the part that needs to be levered out is very loose and does not provide a large resistance force, so a drop in $M_{x,0}$ will not be detected. Therefore, the secondary condition for leveringing success is when the lever angle α becomes higher than a prespecified angle α_{max} . In our case, shown in Fig. 1a, the contact point with the part is always lower than the fulcrum in the object z axis. Thus the leveringing action is stopped if the lever angle is greater than the horizontal tool placement ($\alpha_{max} = 0^\circ$).

2.4 Levering after adaptation

The time required to perform an operation is particularly important in industrial robotics, where fast cycle times are required to optimize the productivity

of the robotic workcell. The knowledge about the performed operations can be used to accelerate the performance. To improve the execution time when the levering operation is performed several times, we record the robot's joint trajectory performed during the initial adaptation of the generic levering action. If the device of the same type is encountered for the second time, we can achieve the necessary contacts without searching. In addition, the levering trajectory can be sped up.

To speed up the levering process, we record the initial position at the start of the levering action and the robot pose reached after establishing the contact of the tool with the levered part (in our case, the PCB) and the fulcrum (edge of the device housing), both in object c.s. The highest amplitude of the adapted levering operation is also recorded. During execution of the adapted levering operation, the robot can directly move from the initial position to the posture where the tool establishes contact with the PCB and the fulcrum. Thus we perform only one movement at a higher speed instead of two. Next the recorded levering operation defined by Eq. (26) is executed at the maximum recorded amplitude and at a higher speed, i.e. by decreasing t_c . The position trajectory defined by Eq. (28) is sped up in the same way. The success signal is monitored as per Section 2.3. When success is not detected, the adaptive levering procedure defined in Section 2.3 is performed. However, here the search process can start at the highest previously recorded amplitude.

3 Experimental evaluation

To test the robustness of the algorithm, we tested altogether 5 exemplars of two different device types and performed the levering for each of them. Some selected devices are in good condition and require a higher levering force, while some are already worn out and require less force.

Figs. 4a and 4b show the average $M_{x,0}$ torques (which we use as a feedback signal) observed during the trials of each device type in various states. Initially the robot is only touching the fulcrum, so the torque values are negative. Upon contacting the PCB, the torque values rise. It can be seen that particularly for devices of type 1, the required levering torque differs significantly depending on the particular device. A torque drop-off is observed at around normalized time $t = 0.8$ in Fig. 4a and around $t = 0.95$ in Fig. 4b, which indicates successful levering completion.

A comparison of the adaptation (initial) trajectory and the adapted trajectory is shown in Figs. 5a and 5b. Instead of searching for contact, the robot immediately moves to the previously learned contact point. Fig. 5c shows the comparison of average execution duration, as well as the standard deviation of this duration, both for the case of the initial search and after adaptation, for each of the two device types. It can be seen that after adaptation, the levering is faster and the duration is deterministic.

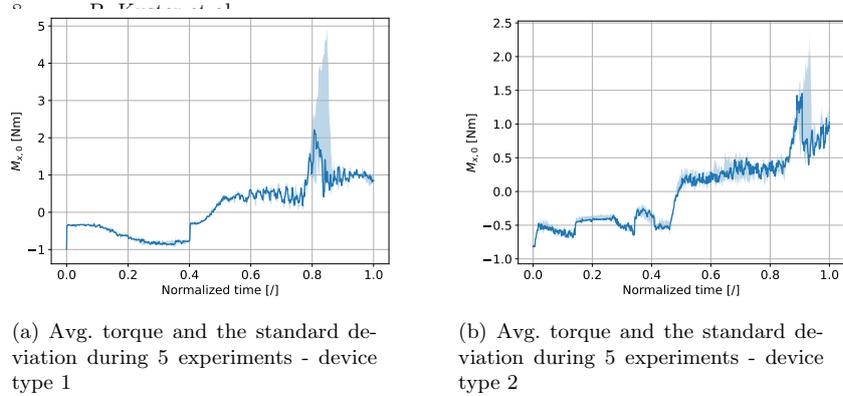


Fig. 4: The torques measured while performing the levering operations.

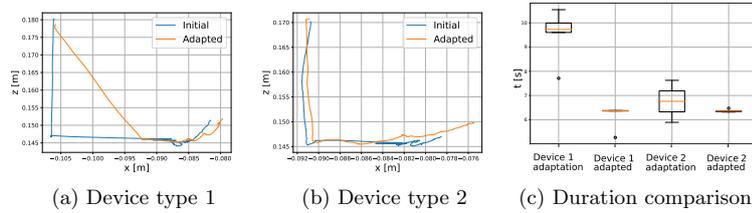


Fig. 5: Comparison of initial and adapted trajectories for two device types (a,b) and duration comparison of initial and adapted levering for both types (c).

4 Conclusion and further work

We presented a parameterized levering algorithm, composed of two sub-tasks, searching and levering. The search algorithm automatically detects contact points after the device pose is estimated by vision. Force control is then used to first establish contact between the tool and the device housing and then position the tool so that it establishes contact with the part to be pried out and the fulcrum. Levering is performed using a sinusoidal motion pattern and force-torque feedback. After the initial learning step, subsequent executions can be sped-up. We have demonstrated the algorithm's robustness to different device types.

With the applied robot, end-effector forces and torques acting are calculated from internal joint torque measurements, which can be noisy, particularly in or near singular joint configurations. This can be solved by using a dedicated force-torque sensor mounted on the end-effector. However, even precise force measurement cannot assure totally reliable classification of the levering process outcome. Additional modalities, such as the gripper encoder feedback signal, could be used to more reliably determine the outcome.

References

1. G. Foo, S. Kara, and M. Pagnucco, "Challenges of robotic disassembly in practice," *Procedia CIRP*, vol. 105, pp. 513–518, 2022.
2. T. Gašpar, M. Deniša, P. Radanovič, B. Ridge, T. R. Savarimuthu, A. Kramberger, M. Priggemeyer, J. Roßmann, F. Wörgötter, T. Ivanovska, S. Parizi, Ž. Gosar, I. Kovač, and A. Ude, "Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells," *Robotics and Computer-Integrated Manufacturing*, vol. 66, 101979, 2020.
3. M. Simonič, T. Petrič, A. Ude, and B. Nemeč, "Analysis of methods for incremental policy refinement by kinesthetic guidance," *Journal of Intelligent & Robotic Systems*, vol. 102, 2021.
4. Y. Laili, Y. Wang, Y. Fang, and D. Pham, *Optimisation of Robotic Disassembly for Remanufacturing*. 2022.
5. P. Radanovič, J. Jereb, I. Kovač, and A. Ude, "Design of a modular robotic workcell platform enabled by plug & produce connectors," in *2021 20th International Conference on Advanced Robotics (ICAR)*, pp. 304–309, 2021.
6. A. Gams, M. Do, A. Ude, T. Asfour, and R. Dillmann, "On-line periodic movement and force-profile learning for adaptation to new surfaces," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pp. 560–565, 2010.
7. O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: challenges, representations, and algorithms," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1395–1476, 2021.