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:	D3.3
Deliverable Title:	Error handling and learning of disassembly primitives
Type:	Report
Dissemination Level:	Public
Authors:	Kübra Karacan, Fan Wu, Hamid Sadeghian, Mihael Simonič, Bojan Nemec, Aleš Ude, and Sami Haddadin
Contributing Partners:	TUM, JSI

Estimated Date of Delivery to the EC:	30-04-2024
Actual Date of Delivery to the EC:	06-05-2024

Contents

Executive summary

In this deliverable, we present the results of Task 3.5 "Error handling and tactile map construction" and learning of the disassembly task energy. Our results on learning disassembly primitives are presented in the deliverable **D3.2**.

We review our contributions to error handling and task energy 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 recycling electronic waste, face inherent limitations, especially when dealing with devices containing hazardous components like batteries. The presence of batteries introduces a fire hazard, necessitating their removal before further recycling steps can proceed. However, successfully removing batteries relies on disassembling the electronic devices. Automating this disassembly process for a wide array of electronic devices presents a challenge, primarily due to the diverse nature of these devices and their varying physical conditions upon disposal. There is an urgent need for efficient and adaptable solutions to enhance the automation of the disassembly process.

Within the scope of the ReconCycle project, we tackle this challenge under three objectives. Firstly, we develop an archetypical disassembly solution tailored to a specific device exemplar, detailing the required steps/actions (e.g., levering, cutting, unscrewing) using a modular and reconfigurable hardware and software architecture (refer to **Objective 1** in the Description of Action (DoA)). Concurrently, sensory information is integrated with the execution steps to establish a semantic representation incorporating variables that capture action-relevant details (refer to **Objective 2** in DoA). This semantic representation is crucial for enabling the robot to autonomously discern the necessary actions for disassembly. Lastly, the robot's actions undergo adaptation and learning for each specific device model and the desired disassembly sequence (refer to **Objective 3** in DoA).

Previous reports have detailed how the modularity and reconfigurability of the developed work cell enable rapid and efficient layout alterations (refer to deliverables **D1.1** and **D1.2**). This flexibility, combined with adaptable soft end-effectors (refer to deliverables **D4.1**, **D4.2**, and **D4.3**), facilitates the handling of various device types within the same work cell.

The reconfigurable hardware is complemented by a modular and hierarchical software architecture, divided into three levels: task-level programming (sequencing robotic skills), programming and acquisition of robotic skills (e.g., levering, unscrewing, pushing, pulling), and low-level control, including skill adaptation. To accommodate the variability of devices within the same device family, we also employ vision-based scene analysis and action prediction (refer to deliverables **D2.1** and **D2.2**), as well as learning disassembly primitives and adapting control parameters (refer to deliverable **D3.2**). Additionally, error-handling mechanisms are implemented to sustain high-quality performance in the work cell, as described in this report.

These capabilities have been successfully demonstrated in use-case-related reports. In deliverable **D5.2**, we presented an archetypical solution for the disassembly of a specific heat cost allocator (HCA) device, laying the foundation for a more generalized pipeline for disassembling different models of HCAs, as outlined in **D5.4**. The proposed pipeline leverages work cell reconfiguration, action prediction, and skill adaptation. Furthermore, in deliverable **D5.5**, we showcased the application of this process to another device type—smoke detectors.

This report summarizes our findings on error handling and learning of the disassembly skill parameters. Section 2 describes how we extend our previous skill definition framework by integrating a fast behavior tree to enable error recovery while executing disassembly skill primitives. Later, in Section 3, we present our flexible control method, which allows the robot to have the option of error recovery from unstable behavior. Additionally, we introduce learning of task energy to sustain a high-quality performance in the low-level control. Finally, the corresponding publications are given in the appendix.

2 1 kHz behavior tree for self-adaptable tactile skills

In the context of recycling, 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. Each step in this protocol requires establishing and aligning the contact between the robotic end-effector and the piece. Nevertheless, contact state estimation and establishment are prone to errors due to perception uncertainties in vision, proprioceptive sensors of the robots, or even undesired contacts. Central to this problem is developing versatile robot skills that are adaptable to new task requirements with minimal human intervention and reprogramming.

The real-time adaptability sequence of the disassembly primitives is an option for error recovery, aiding the robot know when and how to adjust motion strategies to adapt to unknown physical constraints rather than indiscriminately applying force. To address this problem, we propose to extend our skill definition framework with a Behavior Tree (BT) (see Fig. 1) based primitive switching mechanism, which uses high-frequency tactile information for contact state estimation.

The contributions of real-time error handling for the skill execution using a 1 kHz behavior tree framework in the skill execution level can be summarized as follows:

- Real-time contact state estimator: We introduce a real-time contact state estimator for insertion tasks, leveraging time series anomaly detection and tactile information.

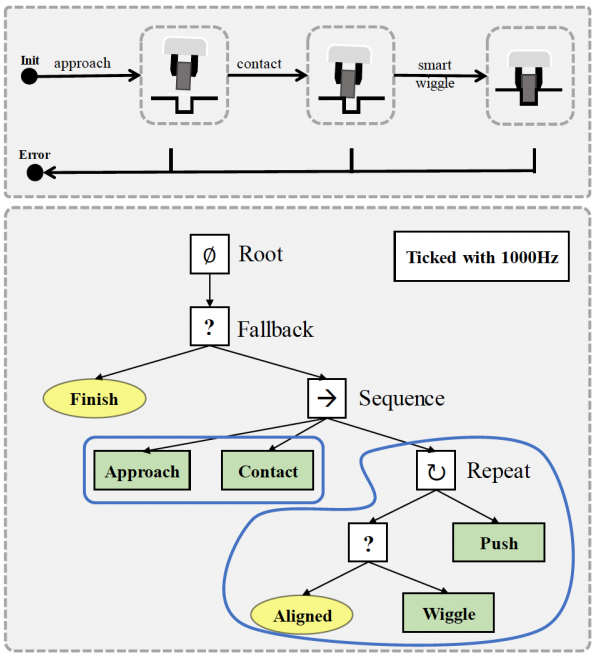


Figure 1: Skill Overview. The upper block depicts our previous skill formalism in deliverable **D3.1** as a Finite State Machine, while the lower one shows the new proposed skill with a Behavior Tree structure. The yellow nodes represent condition nodes, while the green ones indicate action nodes.

- Real-time behavior tree: We incorporate this contact state estimator into our existing insertion skill, revamping it using a behavior tree framework operating at a 1 kHz frequency.
- Experimental validation: We assess the performance of the proposed method by comparing it to our previous approach across various tool alignments and contact initiation, demonstrating its strong efficacy and showing evidence that it can improve learning efficiency in robustness and skill performance. With the new skill framework, the execution time of the final learned skill on our tested objects is almost halved (roughly 50%).
- Transferability test: We showcase that the proposed method surpasses our previous work with a significantly enhanced transferability, i.e., a higher success rate in zero-shot transfers and a more rapid, robust convergence during fine-tuning.

Our 1 kHz Behaviour Tree for Tactile Skills will be presented at the upcoming ICRA 2024 [6].

3 Learning of the task energy for stable and high-performance disassembly skill execution

Robotic manipulation presents many challenges, particularly in adapting predefined contact-rich skills to diverse contexts, as encountered in real-world operations. Therefore, the robot controllers should enable flexibility and adaptability to the undesired contacts for fast error recovery at the low level. To control the skills in ReconCycle, we use the Unified Force-Impedance Control approach, as introduced in the deliverables **D3.1** and **D3.2** [1, 2]. Identifying potential instabilities arising from stiffness variations and force regulations to ensure stability even amidst dynamic changes, virtual energy tanks are integrated to guarantee stability. The stability analysis and virtual energy tank installation for the robotic tactile skills is submitted as a publication to IROS2024 and is currently under review [3]. Even though stability is proven, this does not necessarily mean the task can be fulfilled as desired. If energy tanks are not loaded with sufficient energy, the force controller will be deactivated, or the impedance control will be transited to compliance control. If this happens during manipulation, the intended task goal will not be achieved since the force profile cannot be regulated accordingly or the desired trajectory is not followed correctly. For solving this problem, the concept of initial task energy $E_{\text{tank}}(0)$ is used. $E_{\text{tank}}(0)$ is defined as the minimal energy to be initially stored in the tanks for fulfilling all requirements of a disassembly skill. This task energy or an estimated lower bound needs to be known before execution to determine if stability and correct task execution and performance are to be achieved. The most straightforward strategy would be using energy that is practically high enough. However, for improved safety or process monitoring, it would be beneficial to leverage i.) model-based, ii.) data-driven, or iii.) model-informed hybrid approaches.

Here, one may set $E_{\text{tank}}(0)$ to a constant to fulfill the desired task. However, this leads us to fine-tune it for each task and the corresponding working surface material, geometry, etc. Additionally, when, by default, we set it to a large number, the robot may be loaded with an unnecessarily large amount of energy, leading to a waste of energy in case of instability. So, ideally, after finishing the task at $t = t_{\text{final}}$, the tank should end up with ϵ amount of energy.

The robot controller $\boldsymbol{f}_{\text{cntr}}$ spends the energy of $\int_0^{t_{\text{final}}} \dot{\boldsymbol{x}}^T \boldsymbol{f}_{\text{cntr}}\, d\tau$ when moving with $\dot{\boldsymbol{x}}$.

$$E_{\text{tank}}(t_{\text{final}}) = E_{\text{tank}}(0) - \int_0^{t_{\text{final}}} \dot{\boldsymbol{x}}^T \boldsymbol{f}_{\text{cntr}}\, d\tau \; = \epsilon\,, \tag{1}$$

$$E_{\text{tank}}(0) = \int_0^{t_{\text{final}}} \dot{\boldsymbol{x}}^T \boldsymbol{f}_{\text{cntr}}\, d\tau + \epsilon\,. \tag{2}$$

One option to overcome this issue is to learn the initial tank energy or, in other words, the task energy. Regarding learning, energy budgets are adjusted to accommodate specific contact behavior, as shown in Alg.1. Learning algorithms are implemented to adapt energy budgets for the electronic screwdriver and levering based on observed contact behavior during the execution.

Algorithm 1: Data-driven learning by CMA-ES-based RL

Input : $\boldsymbol{\pi}_{\text{init}} = E_{\text{tank}}(0)$
Output : $\boldsymbol{\pi}$
Initialize: generate initial multi-dimensional Gaussian distribution $X_{\text{init}} \sim \mathcal{N}(\mu,\, \sigma^2)$ based on initial policy $\boldsymbol{\pi}_{\text{init}}$, k = 1
while k $<$ k_{max} or $\sigma^2 > \sigma^2_{\text{min}}$ **do**
  if *sampling* **then**
    if k = 1 **then**
      generate $\boldsymbol{\pi}_{new}$ based on X_{init};
      else
        read $\boldsymbol{\pi}_{old}$, J_c;
        generate X_{new} based on X_{old};
        $\boldsymbol{\pi}_{old}$, J_c;
        generate $\boldsymbol{\pi}_{new}$ with updated X_{new};
  roll-out generated $\boldsymbol{\pi}_{new}$;
  evaluate $\boldsymbol{\pi}_{new}$ and generate costs J based on cost function;
  $\boldsymbol{\pi}_{old} \leftarrow \boldsymbol{\pi}_{new}$;
  $X_{\text{old}} \leftarrow X_{\text{new}}$;
  k = k + 1;

An observation node provides initial motion policy π_{init} on the working surface for the learning module to initialize Gaussian distribution X_{init}. And for the learning algorithm, CMA-ES is implemented under the hood. One significant advantage of a random sampling algorithm, i.e., CMA-ES, is fast convergence, which makes it well-suited for policy search of such tactile manipulation skills. The working principle of CMA-ES can be found in Algorithm 1. With proper initial condition and stopping conditions, i.e., max episode k_{max} and minimum covariance σ^2_{min}, the convergence of samples can be achieved within a short amount of time.

When the sampling request is received, the CMA-ES server node produces one episode of samples based on the current distribution and feeds samples into the robot for execution. During the execution of the samples, the corresponding cost is evaluated based on the cost function and used as feedback. After executing a whole episode, the distribution is updated with the old samples and costs of the current episode. Our cost function J_c is the energy left in the tank at the end of the episode $E_{\text{tank}}(t_{\text{final}})$.

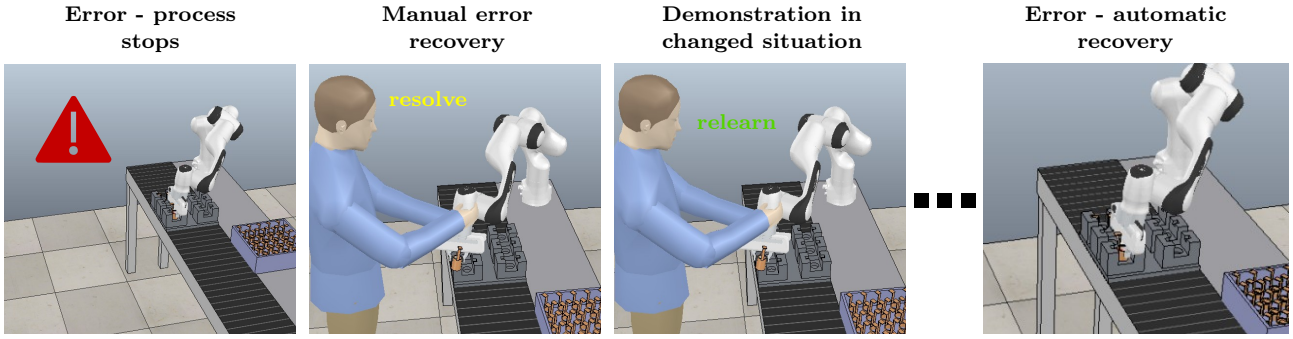


Figure 2: A simplified exception-handling workflow.

4 Learning of exception strategies for error-handling

Many unexpected events may arise during robot disassembly, for which the robot system may not have a predefined scenario. Possible causes include deviations in the geometry of work-pieces or the model, imprecise grasping, discrepancies in positioning, etc. Given the sheer number of potential causes for unintended behavior, it is generally not feasible to anticipate them all. Moreover, since individual unforeseen events occur relatively rarely, it would not be efficient to do so. A well-defined and flexible exception-handling process is crucial to address these unexpected events. While integrating fixed search-and-rescue patterns into existing control policies or policy adaptation can address some of these cases, human-operator intervention is often needed to resolve and resume the operation. However, in most exception-handling setups, the system does not learn from these interventions, and if a similar situation occurs again, it requires human intervention.

We developed a novel framework where the robot learns from the intervention of the human operator and becomes capable of autonomously resolving errors in the future. Our framework is based on determining the context of an exception, kinesthetic guidance, and statistical learning to enable learning of exception strategies and later autonomous resolution of similar exceptions, as illustrated in Fig. 2. In general, it is crucial to understand or at least classify the cause of an error to choose an appropriate strategy to correct it.

Understanding the causes of errors is a complex process that robotic systems cannot fully perform. Therefore, in our system, we use implicit classification based on sensor data describing the circumstances of the failure. This process is called context determination, described in a journal paper [5].

The evaluation was done on two examples of assembly operations. However, the context determination was based on input data from force-torque and vision sensors, typically found in disassembly setups. We linked the context description to the human operator's actions to resolve the situation and the demonstrations to proceed in the changed situation. Over time, a database of demonstrated actions and the associated context is built, and using statistical learning [4], the system was able to generate appropriate actions in unexpected situations.

References

[1] 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)*. 2024.

[2] 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.

[3] K. Karacan, A. Zhang, H. Sadeghian, F. Wu, and S. Haddadin. "Visuo-Tactile Exploration of Unknown Rigid 3D Curvatures by Vision-Augmented Unified Force-Impedance Control". In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. under review. 2024.

[4] T. Petrič, A. Gams, L. Colasanto, A. J. Ijspeert, and A. Ude. "Accelerated Sensorimotor Learning of Compliant Movement Primitives". In: *IEEE Transactions on Robotics* 34.6 (2018), pp. 1636–1642.

[5] M. Simonič, M. Majcen Hrovat, S. Džeroski, A. Ude, and B. Nemec. "Determining Exception Context in Assembly Operations from Multimodal Data". In: *Sensors* 22.20 (2022). DOI: 10.3390/s22207962.

[6] Y. Wu, F. Wu, L. Chen, K. Chen, S. Schneider, L. Johannsmeier, Z. Bing, F. Abu-Dakka, A. Knoll, and S. Haddadin. "1 kHz Behavior Tree for Self-adaptable Tactile Insertion". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2024.

A Copies of scientific publications

A.1 "1 kHz Behavior Tree for Self-adaptable Tactile Insertion"

This is the pre-print version (author accepted manuscript) of the following publication: Y. Wu, F. Wu, L. Chen, K. Chen, S. Schneider, L. Johannsmeier, Z. Bing, F. Abu-Dakka, A. Knoll, and S. Haddadin. "1 kHz Behavior Tree for Self-adaptable Tactile Insertion". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2024.

1 kHz Behavior Tree for Self-adaptable Tactile Insertion

Yansong Wu[1], Fan Wu[1], Lingyun Chen[1], Kejia Chen[1], Samuel Schneider[1], Lars Johannsmeier[2], Zhenshan Bing[1], Fares J. Abu-Dakka[3], Alois Knoll[1], Sami Haddadin[1]

***Abstract*— Insertion is an essential skill for robots in both modern manufacturing and services robotics. In our previous study, we proposed an insertion skill framework based on force-domain wiggle motion. The main limitation of this method lies in the robot's inability to adjust its behavior according to changing contact state during interaction. In this paper, we extend the skill formalism by incorporating a behavior tree-based primitive switching mechanism that leverages high-frequency tactile data for the estimation of contact state. The efficacy of our proposed framework is validated with a series of experiments that involve the execution of tightly constrained peg-in-hole tasks. The experiment results demonstrate a significant improvement in performance, characterized by reduced execution time, heightened robustness, and superior adaptability when confronted with unknown tasks. Moreover, in the context of transfer learning, our paper provides empirical evidence indicating that the proposed skill framework contributes to enhanced transferability across distinct operational contexts and tasks.**

I. Introduction

Since the early stage of automatons and industrial robots, manufacturing has been one of the most important sectors motivating and witnessing the evolution of robotics [1]. Transitioning from Industry 4.0 to Industry 5.0 [2], robotic assembly meets challenges from flexible manufacturing's rise, requiring efficient small batch production management in automated factories. This brings the research focus from implementing robots on repetitive tedious tasks, be it simple pick-and-place, or welding, grinding, assembly, in a structured environment to an unstructured dynamic environment, even with humans in their vicinity to collaborate. Central to this problem is how to develop versatile robot skills that are adaptable to new task requirements with minimal human intervention and reprogramming.

Among many skills for contact-rich manipulation, *Insertion*, also known as *Peg-in-Hole* (as depicted in Fig. 1) is of paramount importance and has received numerous research efforts recently [3]–[19]. Admittedly, a myriad of recent works [3]–[10], [14]–[19] take the *learning-based* approach in contrast to those exploiting human expert knowledge to handcraft solutions [11]–[13]. The learning-based methods span three main categories: (i) end-to-end (deep) reinforcement learning (RL), whether taking force signals [5], [7], [16] or visuo-tactile sensing [9], [15] into model inputs; (ii) imitation learning or learning from demonstration (LfD) [3], [4], [6], [17]; and (iii) parameterized skill learning [8], [10]. In the line of deep RL, training a general model with meta-reinforcement learning [14], [18], [20], [21] seems promising to acquire highly versatile and transferable insertion skills. Nevertheless, the lack of sample efficiency, safety guarantees and interpretability are imperatives to its real-world deployment. Imitation learning as a more sample-efficient approach has been adopted widely in industrial applications, combined with RL to further optimize control policies. However, learning skills constrained by changing environments as well as capable of real-time adaptation based on tactile information is still an open problem.

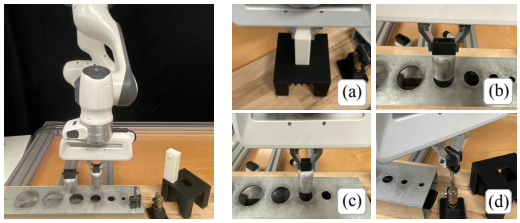


Fig. 1: Experiment setup for Tactile Insertion. The left figure shows the overall setup. Objects used in this work are: (a) **Object A**: a cuboid with the geometry size of $35\,\text{mm} \times 25\,\text{mm} \times 60\,\text{mm}$, clearance is $0.1\,\text{mm}$ in each dimension, (b) **Object B**: a cylinder length of $50\,\text{mm}$ and diameter of $40\,\text{mm}$, clearance is $0.05\,\text{mm}$, (c) **Object C**: a cylinder with length of $50\,\text{mm}$ and diameter of $30\,\text{mm}$, clearance is $0.025\,\text{mm}$, (d) **Object D**: a $37\,\text{mm}$ long key.

Compared to learning-based methods, off-the-shelf solutions [22] programmed by human experts are still more widely used in real manufacturing, which has well-structured environments but requires high precision manipulation. They often outperform learning-based methods in certain aspects [19]. In the context of robotic insertion, most popular approaches [11]–[13] usually feature force-based spiral search strategies and a skill framework consisting of multiple phases or primitives. Multi-phase skill formalism is also used in [8], [10] with a force-based spiral search primitive termed as "Wiggle" motion, which enables learning in reduced parameter space, resulting in much higher sample efficiency compared to deep RL approaches.

The successful use of spiral force search as demonstrated in [13] replies on the use of active compliance, which

[1]The authors are with the Chair of Robotics and Systems Intelligence, MIRMI - Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, Germany. f.wu@tum.de The authors acknowledge the financial support by 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). In addition to the support by euROBIN project under grant agreement No. 101070596, by 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, 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 and by the European Union's Horizon 2020 research and innovation programme as part of the project ReconCycle under grant no. 871352. Note that S. Haddadin had a potential conflict of interest as a shareholder of Franka Emika GmbH.

[2]Franka Robotics GmbH, Germany.

[3]Electronic and Informatics Department, Faculty of Engineering, Mondragon Unibertsitatea, Bilbao, Spain.

resonates with the old idea of *adapting to physical interaction rather than overcoming it*, first implemented by McCallion et al. [23] in a *physical* compliance device for an industrial insertion task. However, most previous works focus only on searching (approximately) optimal solutions, either by learning or human programming, to solve the hole searching problem. The effectiveness, performance and transferability of the insertion skill, in terms of adapting to physical interaction (in the presence of imperfect perception and changing environment constraints) during the process when the peg is being pushed into the hole, remains an under-explored question. This is in part due to the fact that tight-clearance industrial assembly tasks [5] are rarely investigated in the research community. On the contrary, many studies are conducted with "generous" clearance tasks, which inevitably biases on hole searching and mitigates the importance of adaptability and failure recovering during the whole process of insertion.

In our previous works [8], [19], we demonstrated the feasibility of replicating human-like wiggling with feed-forward force in robotic insertion tasks. However, this approach is still far from achieving human performance in terms of real-time adaptability when conducting new tasks. This is due to the fact that humans know when and how to adjust motion strategies to adapt to unknown physical constraints, rather than indiscriminately applying force. To address this problem, in this paper, we propose to extend the skill framework with a Behavior Tree (BT) [24] based primitive switching mechanism, which uses high-frequency tactile information for contact state estimation.

The contributions of this work can be summarized as follows:

1) Real-time contact state estimator: We introduce a real-time contact state estimator for insertion tasks, leveraging time series anomaly detection and tactile information.
2) Real-time behavior tree: We incorporate this contact state estimator into our existing insertion skill, revamping it using a behavior tree framework operating at a 1 kHz frequency.
3) Experimental validation: We assess the performance of the proposed method by comparing it to our previous approach across various insertion tasks, demonstrating its strong efficacy and showing evidence that it can improve learning efficiency in terms of robustness and skill performance. With the new skill framework, execution time of final learned skill on our tested objects is almost halved (roughly 50% reduction).
4) Transferability test: We showcase that the proposed method surpasses our previous work with a significantly enhanced transferability, *i.e.,* a clearly higher success rate in zero-shot transfers and a more rapid, robust convergence during fine-tuning.

II. Methods

A. Adaptive Impedance control with Feed-forward Force

Consider a torque-controlled robot with n-Degrees of Freedom, the second-order rigid body dynamics is written as:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_m + \boldsymbol{\tau}_{\text{ext}} \tag{1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the joint position. $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ corresponds to the mass matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the Coriolis matrix and $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravity vector. The motor torque (control input) and external torque are denoted by $\boldsymbol{\tau}_m \in \mathbb{R}^n$ and $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^n$, respectively. The adaptive impedance control law with feed-forward force profile is defined as [25]:

$$\begin{aligned}\boldsymbol{\tau}_m(t) =& \boldsymbol{J}(\boldsymbol{q})^{\mathsf{T}}[\boldsymbol{F}_{ff}(t) + \boldsymbol{K}(t)\boldsymbol{e} + \boldsymbol{D}\dot{\boldsymbol{e}} \\ &+ \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{x}}_d + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{x}}_d] + \boldsymbol{g}(\boldsymbol{q}),\end{aligned} \tag{2}$$

where $\boldsymbol{F}_{ff}(t)$ compensates the feed-forward wrench, while $\boldsymbol{x}_d$ is the desired trajectory. $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}$ and $\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}$ are the position and velocity error, respectively. $\boldsymbol{K}(t)$ and $\boldsymbol{D}$ are stiffness and damping matrices in Cartesian space. $\boldsymbol{J}(\boldsymbol{q})$ represents the robot Jacobian matrix. This control law is used in all motion primitives in the skill framework, which will be introduced below.

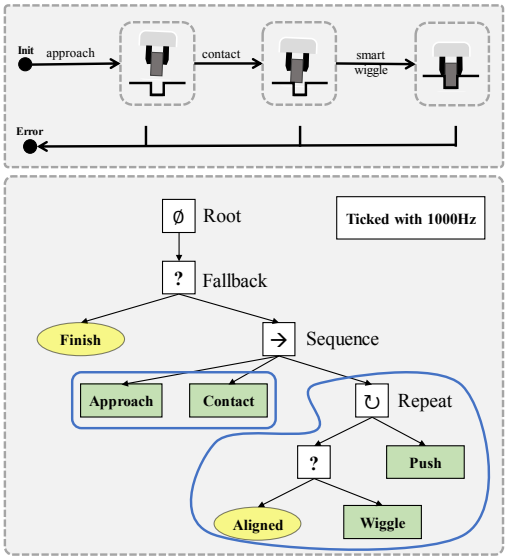


Fig. 2: Skill Overview. The upper block depicts our previous skill [8] formalism structured as a Finite State Machine, while the lower one shows the new proposed skill with a Behavior Tree structure. The yellow nodes represent condition nodes, while the green ones indicate action nodes.

B. Insertion Skill Design

In contrast to [8], as shown in Fig. 2, the architecture of the skill framework proposed in this paper shifts from a sequential Finite State Machine to a Behavior Tree of depth 5, which iterates at 1 kHz frequency to decide which type of actions is executed based on real-time contact state estimation. Every 1 ms an enabling signal is fired out from the Root node to its leaf nodes. These triggering signals, also called "ticks", traverse recursively in the tree following the *Depth First Search* rule. The continual generation of ticks and their tree traversal result in a closed loop execution. Actions are executed and aborted according to the ticks' traversal, which depends on the leaf nodes' return statuses [24].

From the beginning of the task, the robot gripper moves from its initial position towards the hole until contact established. In this Pre-insertion phase only **Approach** and **Contact** primitives are used, and their action nodes are almost surely executed in sequence by applying the control law Eq. (2) with $\boldsymbol{F}_{ff}(t) = 0$ to follow the desired trajectory $\boldsymbol{x}_d$.

After establishing contact, the tick in BT traverses through the Repeat node at Depth 3, which triggers its left child node to estimate contact state (see Sec. II-C.3 below) and evaluate how the peg is aligned with the hole. If the condition "Aligned == Yes" is fulfilled, the fallback node returns success and its sibling node, the **Push** action node, is executed, whereas **Wiggle** is executed whenever the estimation of the alignment returns false. **Wiggle** and **Push** are implemented based on the control law Eq. (2). In **Wiggle**, $\boldsymbol{F}_{ff}(t)$ follows a designed trajectory from motion generator; In **Push**, $\boldsymbol{F}_{ff}(t)$ maintains the last updated value.

In our previous works [8], [19], the Lissajous curve-shaped feed-forward force $\boldsymbol{F}_{ff}(t)$ is leveraged to mimic the human's periodic wiggle motion. The desired force trajectory in direction i is formulated as:

$$F_{ff,i}(t) = a_i \cdot \sin(2\pi f_i t + \varphi_i) \tag{3}$$

where a_i, f_i and φ_i refer to the amplitude, frequency and phase, respectively. The subscript i refers to the direction in the range x, y, rx, ry, rz of the End-Effector (EE) frame. The applied force in the z direction (main assembly direction) maintains a constant value a_z.

The feasibility and efficiency of applying feed-forward force to mimic the human's wiggle motion in a robotic insertion task have been demonstrated in [8], [19]. The advantages are twofold: On the one hand, it can search and align the hole before inserting the peg; On the other hand, during the Insertion phase, wiggling effectively help the peg get out of a stuck state.

However, by further observing how humans perform insertion tasks on tight-clearance objects, humans tend to employ wiggling motions only when necessary, which coincides to the minimum intervention principle (in a loose sense). From the perspective of energy, the optimal solution while achieving task goal during tight-tolerance insertion, should exert minimal energy to overcome friction and recover from anomaly, *i.e.,* the peg getting stuck due to misalignment with physical constraints. Moreover, humans have remarkable ability to generalize their manipulation skills to unseen new tasks without new training. For instance, given a new difficult tight-clearance peg-in-hole task, many people would naturally utilize force spiral search or wiggle motion for contact alignment and failure recovering during insertion. In other words, humans can intentionally self-adapt tactile skill to tackle complex novel tasks. In philosophical terminology, this intentional self-adaptability exemplifies a meta-agentive capability, that intervenes in and influences other agentive processes.

Based on the above observation and reasoning, we postulate that "mimicking" such meta-agentive ability is the key to realize robot skills that are highly transferable to new tasks with various environment constraints. Without over-complicating the problem by taking less interpretable meta-RL approach, in this paper we propose to incorporate human knowledge into skill framework by designing a simple yet effective behavior tree-based skill formalism to achieve dynamic and reactive self-adaptable behavior in insertion skills.

Algorithm 1 Real-time Contact State Estimation

```
z ← 0, s ← Searching                                   ▷ initial
record current x_z as x_{z_0}
for any new data x_z do
    if s == Searching then
        if x_z − x_{z_0} > ϵ then
            s ← Stuck                                  ▷ searching success
            z ← z(x_z)
        end if
    else
        z ← z(x_z)
        if s == Stuck & z > 3 then
            s ← Unstuck                                ▷ Stuck to Unstuck
            if f_{res_z} is local maximum then
                s ← Aligned & v_ref ← αv               ▷ alignment
            end if
        else if s ≠ Stuck & v < v_ref then
            s ← Stuck                                  ▷ get stuck
        end if
    end if
    add x_z into z-score detection buffer
end for
```

C. Real-time Contact State Estimation

1) data pre-processing: To mitigate the impact of high-frequency noises, the robot states series $\boldsymbol{X}$ is filtered by convolution with a Blackman window [26], [27]:

$$\boldsymbol{w}[n] = 0.42 - 0.5 \cdot \cos\left(2\pi \frac{n}{N}\right) + 0.08 \cdot \cos\left(4\pi \frac{n}{N}\right) \tag{4}$$

$$\boldsymbol{w}[n] = \frac{\boldsymbol{w}[n]}{\sum_{i=1}^{N} \boldsymbol{w}[i]} \tag{5}$$

$$\tilde{\boldsymbol{X}} = \boldsymbol{X} * \boldsymbol{w} \tag{6}$$

where $\boldsymbol{w}[n]$ is the n-th element in a Blackman window of length $N = 50$. $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}$ refer to the measured and filtered time series, respectively.

2) moving z-score based "Unstuck" state detection: The moving z-score is a commonly employed methodology for quantifying the degree of anomaly exhibited by individual data points within a time series [28]. Applying it to x_z (the z-position of the EE's frame w.r.t. the task frame), the z-score value of the new coming measured point is:

$$z = \frac{x_z - \mu}{\sigma} \tag{7}$$

where the mean μ and standard deviation σ are calculated over the previous observations.[1] Grounded in the concept of statistical dispersion, if the z-score associated with a newly acquired sample surpasses three, it warrants classification as an anomalous data point, with a confidence level of 97.7%. As illustrated in the third row of Fig. 3, the EE's

[1]In this work, measurements from the last 1 second are used as reference.

z-position in the initial searching phase and when the object is stuck closely approximates a horizontal line. As the object transitions from a stuck state to becoming unstuck, it undergoes a rapid upward elevation. This turning point can be effectively captured via anomaly detection.

3) contact state estimation: As depicted in algorithm 1, the contact estimation may output different candidate states, *i.e.,* "Searching" indicates the robot is in the process of locating the hole; "Stuck" means the insertion object gets stuck; "Unstuck" represents the object is moving along the insertion direction and "Aligned" signs that the object is currently aligned with the insertion hole. Due to the existence of clearance, a misaligned object may also move in the insertion direction with a pressing force. For such kind of object, sequenced wiggle motion helps it get closer to the perfect aligned pose. Compared to a misaligned object, an aligned object experiences less resistance under the same conditions. Therefore, our contact detector estimates the alignment moment by identifying the local maximum of f_{res_z}, namely the resistance force $\boldsymbol{F}_{\text{res}}$ in the z-direction.

$$[\boldsymbol{F}_r^{\mathsf{T}}, \boldsymbol{\tau}_r^{\mathsf{T}}]^{\mathsf{T}} = \boldsymbol{J}_{\text{body}}^{-\mathsf{T}}(\boldsymbol{\tau}_m - \boldsymbol{C}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}} - \boldsymbol{g}\left(\boldsymbol{q}\right)) \tag{8}$$

$$\boldsymbol{F}_{\text{res}} = \boldsymbol{F}_r - \boldsymbol{F}_{\text{ext}} \tag{9}$$

where $\boldsymbol{F}_r$ and $\boldsymbol{\tau}_r$ refer to the force and torque exerted by the robot on the insertion object. $\boldsymbol{J}_{\text{body}}$ represents the body Jacobian, relating joint velocities to the EE twist expressed in the body frame (a frame at the EE). $\boldsymbol{F}_{\text{ext}}$ indicates the estimated external force based on the joint torques.

As the f_{res_z} reaches a local minimum, the corresponding velocity in the z-direction is multiplied by a discount factor ($\alpha = 0.1$) to generate a reference speed. When the object's velocity drops below this threshold, the system state is re-evaluated as "Stuck".

D. Evolution Strategy based Learning Algorithm

1) Exploration and evaluation: The exploration phase generates K unconstrained perturbations in skill parameter space for K roll-outs. These perturbations are assumed to obey the multi-variate Gaussian distribution $\tilde{\boldsymbol{\xi}}_k \sim \mathcal{N}(\boldsymbol{\xi}, \boldsymbol{\Sigma}_\epsilon)$, where $k = 1, 2, ..., K$ and $\boldsymbol{\xi}$ indicates the centre of the distribution and $\boldsymbol{\Sigma}_\epsilon$ indicates the covariance matrix. Then, the box constraints $\boldsymbol{\xi}_{\max}$ and $\boldsymbol{\xi}_{\min}$ are applied while mapping the perturbation $\tilde{\boldsymbol{\xi}}_k$ to the parameter vector $\boldsymbol{\xi}_k$ (detailed in [8]), which represents the whole policy of the k-th roll-out.

$$\boldsymbol{\xi}_k = \min(\max(\tilde{\boldsymbol{\xi}}_k, \boldsymbol{\xi}_{\min}), \boldsymbol{\xi}_{\max}) \tag{10}$$

where min and max are evaluated element-wise. The performance of each roll-out is evaluated with the cost function:

$$J = \frac{t_{\text{exe}}}{t_{\max}} + \Phi \cdot \mathrm{e}^d \tag{11}$$

It includes the following aspects: (i) **Execution time**: t_{exe} and $t_{\max}$ represent the execution time and time limitation; (ii) **Task accomplishment**: The Boolean value Φ equals to 0 for a completed handover and 1 for an unsuccessful trial; and (iii) **Average distance**: The average distance d between the EE and insertion hole indicates the quality of an unsuccessful sample. The larger the value, the further it deviates from a successful trial, vice versa.

2) Policy update: The policy update steps (12)-(16) are based on the PI^{BB} algorithm introduced by [29].

$$\tilde{J}_k = \frac{J_k - \min(\{J_k\})}{\max(\{J_k\}) - \min(\{J_k\})} \tag{12}$$

$$P_k = \frac{\exp\left(-c\tilde{J}_k\right)}{\sum_{i=1}^{K}\exp\left(-c\tilde{J}_i\right)} \tag{13}$$

$$\boldsymbol{\xi} \leftarrow \sum_{k=1}^{K} P_k \boldsymbol{\xi}_k \tag{14}$$

$$\boldsymbol{\Sigma}_\epsilon^{\text{temp}} = \sum_{k=1}^{K} P_k(\boldsymbol{\xi}_k - \boldsymbol{\xi})(\boldsymbol{\xi}_k - \boldsymbol{\xi})^{\mathsf{T}} \tag{15}$$

$$\boldsymbol{\Sigma}_\epsilon \leftarrow \boldsymbol{\Sigma}_\epsilon + \gamma(\boldsymbol{\Sigma}_\epsilon^{\text{temp}} - \boldsymbol{\Sigma}_\epsilon) \tag{16}$$

First, the cost J_k is normalized according to their maximum and minimum by (12). The normalized cost $\tilde{J}_k$ is used to calculate probability P_k for k-th roll-out according to (13), where $c > 0$ is a constant. Then, the distribution is updated, according to the weighted averaging rule (14)-(16), where $\gamma \in (0, 1]$ is the applied decate factor while updating the covariance matrix.[2]

III. Experiment

To evaluate our proposed method, we designed three experiments to: (i) demonstrate the performance improvement of our proposed skill framework with behavior tree and contact state estimation over that without them, (ii) validate the learning performance, and (iii) investigate the transferability. The original skill without behavior tree and state estimation is utilized as our comparing baseline. The experiments are implemented with a 7-DoF franka robot [30] and 4 tight-clearance insertion objects, as illustrated in Fig. 1.

A. Skill Performance

To validate the efficiency of our proposed method, we conducted a series of insertion tasks with our proposed methods and compared them against the baseline. These tasks were carried out using Object A, and the process was repeated 100 times with different parameters. These parameters are sampled from a Gaussian distribution, generated based on successful samples obtained when using the baseline executing various insertion tasks. The results indicate: (i) Our proposed method achieves a significantly improved success rate of 30%, whereas the original skill yields a success rate of 21%. (ii) For completed trials, an 8.9% reduction in overall execution time is observed.

TABLE I: Parameters value

Parameter	Value
K_{xyz} [N/m]	523.907
K_r [N/rad]	24.984
$[a_x, a_y, a_z]$ [N]	[1.792, 2.360, 4.931]
$[a_{rx}, a_{ry}, a_{rz}]$ [N/rad]	[0.766, 0.906, 3.228]
$[\varphi_x, \varphi_y]$	[-0.078, 0.776]
$[\varphi_{rx}, \varphi_{ry}, \varphi_{rz}]$	[-1.562, 0.610, -0.119]
$[f_x, f_y]$	[2.179, 1.561]
$[f_{rx}, f_{ry}, f_{rz}]$	[0.718, 0.720, 0.143]

[2]In this work, $c = 10$ and $\gamma = 0.9$.

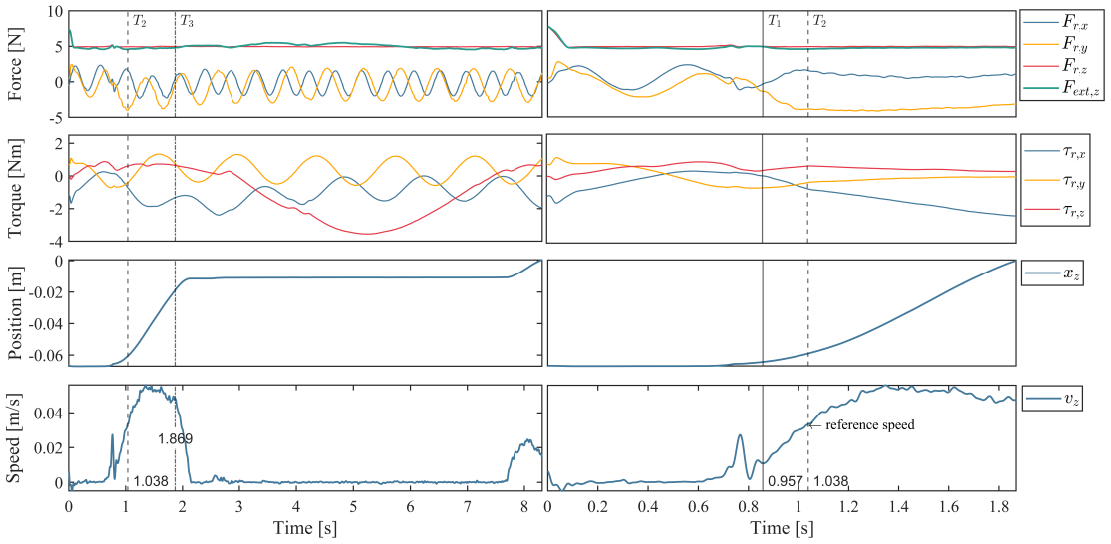


Fig. 3: Skill performance. The figures on the left correspond to the insertion with baseline, whereas the right figures demonstrate the insertion with our proposed method. Both of them are conducted with identical parameters (detailed in Table I). In these figures, specific time points are marked for reference: T_1 signifies the moment when the object transitions from a Stuck to an Unstuck state; T_2 represents the time point when the object is estimated in an Align state; T_3 denotes the complication time of our proposed method (the end time in the right subgroups).

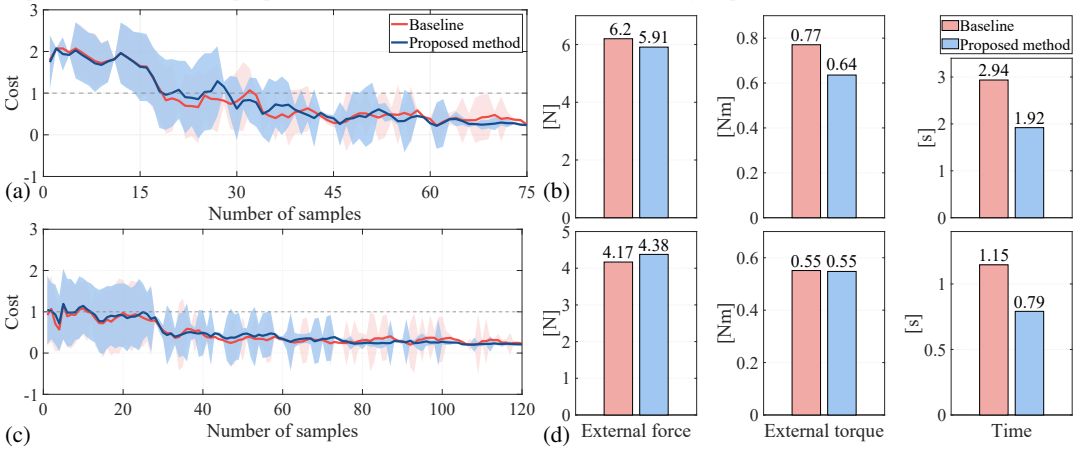


Fig. 4: Learning performance. (a) learning curve of Object A, (b) measured external force, torque and execution time of final result on Object A, (c) learning curve of Object C, (d) measured external force, torque and execution time of final result on Object C.

To gain a comprehensive understanding of the influence of the behavior tree and contact state detection on the Insertion phase, the results executed with the parameters in Table I are visualized in Fig. 3 (The parameters' meaning is detailed in [8]). The figures in the first two rows depict the estimated wrench $\boldsymbol{F}_r$ and $\boldsymbol{\tau}_r$ exerted on the object by the robot. Additionally, the green line represents the external force exerted on the object by the environment. The corresponding position and speed of the EE in z-axis are illustrated in the last two rows, respectively. Note that, all the measurements in this figure are the pre-processed with Eq. (6).

In the initial stage (before T_2), the performance of both skills exhibits significant similarities. At the moment T_1, our proposed contact estimator detects a critical event: The object successfully transitions from a Stuck to an Unstuck state after locating the insertion hole. Following this, at T_2, our proposed method stops its wiggle motion when it meets an optimally aligned gesture, identified as a local maximum of the resultant force f_{res_z}. Subsequently, the robot transitions its action mode to pushing with a constant feed-forward force and accomplishes the task at time T_3; In contrast, the baseline keeps wiggling naively after T_2 and misses the achieved

aligned position, resulting in a prolonged execution time.

B. Learning Performance

In this section, we employ the evolutionary strategy detailed in Section II to train the robot solving insertion task, utilizing the Object A and Object C, as depicted in Fig. 1, respectively. Each training process is repeated 10 times. The costs during the training process are presented in the left part of Fig. 4. The red line represents the mean of the training processes based on the baseline, while the blue line indicates that of our proposed method. The shadow area represents the corresponding variance. Additionally, a horizontal dashed line shows the boundary to distinguish between successful and unsuccessful trials. Evaluating the overall performance of the Pre-Insertion and Insertion phases with Eq. (11), the proposed method demonstrates a modest improvement, characterized by reduced cost and variance. However, it is worth highlighting that the Pre-Insertion phase is identical for both methods, with the sole distinction arising during the Insertion phase. Therefore, the right-hand figures provide a detailed analysis of the Insertion phase, demonstrating a marked improvement in execution speed while ensuring effective limiting of the contact force. Specifically, the average execution speeds for the tasks improved by 52.9% and 45.6%, respectively.

C. Transferability

In this section, we assess the transferability of our method by examining its zero-shot transfer and fine-tuning performances.

1) zero-shot transfer: We apply the policies (skills with optimal parameters) learned from Object A to tasks for which it was not explicitly trained, *i.e.,* the insertion of Objects B, C, and D. This procedure is executed 100 times using the policies derived from both methods (in Sec. III-B). The results are depicted in Fig. 5. For each object, the policy derived from our proposed method, represented in blue, consistently demonstrates significantly higher success rates in comparison to the baseline method, depicted in red, resulting in an overall enhancement in the success rate by 22.7%.

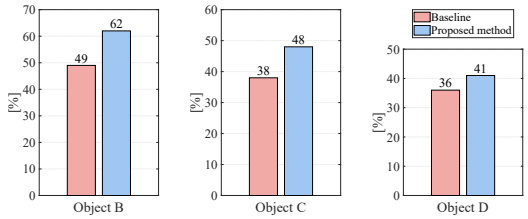


Fig. 5: Success rate while transferring the models learned with Object A to the other objects shown in Fig.1.

2) fine-tuning: Subsequently, we utilize the policies developed for Object A as pre-trained models and proceed to fine-tune them for the insertion tasks involving Objects B, C, and D. As depicted in Fig. 6, our method demonstrates notable efficiency and robustness improvements. Specifically, for Object B, our approach not only converges 33.3% faster than the baseline but also exhibits a 49.4% reduction in performance's variance. Regarding Object C, our approach consistently outperforms the baseline throughout the learning process. Notably, for Object D (characterized by its unique type and complex geometry), our method reaches convergence 1.7 times quicker than the baseline, with a notable 66.2% reduction in outcome variance. These experimental outcomes affirm the superior transferability of our newly proposed skill framework, primarily due to the improved self-adaptability by the integration of contact state estimator and the BT structure.

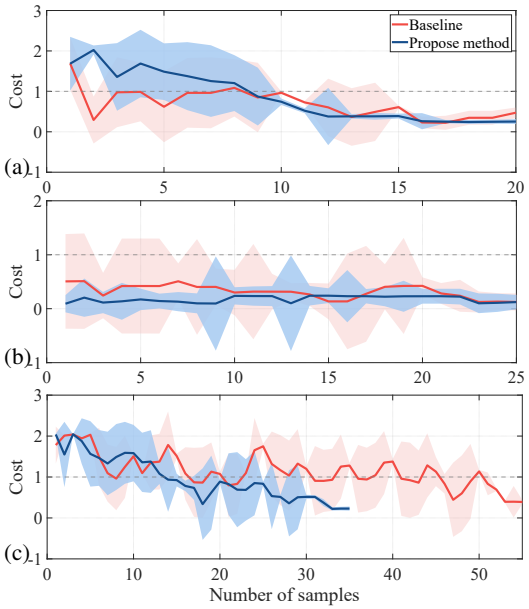


Fig. 6: Fine-tuning performances. Shown in figure are: (a) Object B, (b) Object C, and (c) Object D. The experiment is conducted five times, with the solid line depicting the mean values and the shaded area indicating the variance.

IV. Conclusion

This paper enhanced our precious framework by incorporating behavior tree and contact state estimation. The efficiency of our proposed framework has been validated with various tight-clearance insertion tasks. The experiment results showcased a substantial improvement with reduced execution time while ensuring controlled contact forces. Additionally, it demonstrated enhanced robustness and superior performance when learning unknown tasks. Furthermore, the transfer learning experiment implies that our extended skill framework can effectively enhance the skill transferability, by improving the model's self-adaptability through the proposed contact state estimator and 1 kHz BT structure. In future works, we will conduct extensive empirical research on investigating skill transfer learning involving a wider range of objects.

References

[1] J. Wallen, "The history of the industrial robot," 2008.
[2] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, "Industry 4.0 and industry 5.0—inception, conception and perception," *Journal of Manufacturing Systems*, vol. 61, pp. 530–535, 2021.

[3] Y. Mollard, T. Munzer, A. Baisero, M. Toussaint, and M. Lopes, "Robot programming from demonstration, feedback and transfer," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 1825–1831.

[4] T. Tang, H.-C. Lin, Y. Zhao, Y. Fan, W. Chen, and M. Tomizuka, "Teach industrial robots peg-hole-insertion by human demonstration," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2016, pp. 488–494.

[5] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana, "Deep reinforcement learning for high precision assembly tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 819–825.

[6] Z. Zhu and H. Hu, "Robot learning from demonstration in robotic assembly: A survey," *Robotics*, vol. 7, no. 2, p. 17, 2018.

[7] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, A. M. Agogino, A. Tamar, and P. Abbeel, "Reinforcement learning on variable impedance controller for high-precision robotic assembly," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3080–3087.

[8] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5844–5850.

[9] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8943–8950.

[10] F. Voigt, L. Johannsmeier, and S. Haddadin, "Multi-level structure vs. end-to-end-learning in high-performance tactile robotic manipulation," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 2306–2316.

[11] J. Watson, A. Miller, and N. Correll, "Autonomous industrial assembly using force, torque, and RGB-d sensing," *Advanced Robotics*, vol. 34, no. 7, pp. 546–559.

[12] G. Gorjup, G. Gao, A. Dwivedi, and M. Liarokapis, "Combining compliance control, CAD based localization, and a multi-modal gripper for rapid and robust programming of assembly tasks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9064–9071, ISSN: 2153-0866.

[13] H. Park, J. Park, D.-H. Lee, J.-H. Park, and J.-H. Bae, "Compliant peg-in-hole assembly using partial spiral force trajectory with tilted peg posture," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4447–4454, 2020.

[14] G. Schoettler, A. Nair, J. A. Ojea, S. Levine, and E. Solowjow, "Meta-reinforcement learning for robotic industrial insertion tasks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9728–9735.

[15] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020.

[16] Y. Shi, Z. Chen, Y. Wu, D. Henkel, S. Riedel, H. Liu, Q. Feng, and J. Zhang, "Combining learning from demonstration with learning by exploration to facilitate contact-rich tasks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1062–1069.

[17] Y. Li and D. Xu, "Skill learning for robotic insertion based on one-shot demonstration and reinforcement learning," *International Journal of Automation and Computing*, vol. 18, no. 3, pp. 457–467, 2021.

[18] T. Z. Zhao, J. Luo, O. Sushkov, R. Pevceviciute, N. Heess, J. Scholz, S. Schaal, and S. Levine, "Offline meta-reinforcement learning for industrial insertion," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6386–6393.

[19] L. Johannsmeier and S. Haddadin, "Can we reach human expert programming performance? a tactile manipulation case study in learning time and task performance," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12 081–12 088.

[20] Z. Bing, D. Lerch, K. Huang, and A. Knoll, "Meta-reinforcement learning in non-stationary and dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3476–3491, 2022.

[21] X. Yao, Z. Bing, G. Zhuang, K. Chen, H. Zhou, K. Huang, and A. Knoll, "Learning from symmetry: Meta-reinforcement learning with symmetrical behaviors and language instructions," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 5574–5581.

[22] W. Lian, T. Kelch, D. Holz, A. Norton, and S. Schaal, "Benchmarking off-the-shelf solutions to robotic assembly tasks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1046–1053, ISSN: 2153-0866.

[23] H. McCallion, G. Johnson, and D. Pham, "A compliant device for inserting a peg in a hole," *Industrial Robot: An International Journal*, vol. 6, no. 2, pp. 81–87.

[24] M. Colledanchise and P. Ögren, *Behavior trees in robotics and AI: An introduction*. CRC Press, 2018.

[25] 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.

[26] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.

[27] R. B. Blackman and J. W. Tukey, "The measurement of power spectra from the point of view of communications engineering—part i," *Bell System Technical Journal*, vol. 37, no. 1, pp. 185–282, 1958.

[28] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The journal of finance*, vol. 23, no. 4, pp. 589–609, 1968.

[29] F. Stulp and O. Sigaud, "Policy improvement methods: Between black-box optimization and episodic reinforcement learning," 2012.

[30] 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, Jun. 2022.

A.2 "Visuo-Tactile Exploration of Unknown Rigid 3D Curvatures by Vision-Augmented Unified Force-Impedance Control"

This is the version under review of the following publication: K. Karacan, A. Zhang, H. Sadeghian, F. Wu, and S. Haddadin. "Visuo-Tactile Exploration of Unknown Rigid 3D Curvatures by Vision-Augmented Unified Force-Impedance Control". In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. under review. 2024.

Visuo-Tactile Exploration of Unknown Rigid 3D Curvatures by Vision-Augmented Unified Force-Impedance Control

Kübra Karacan, Anran Zhang, Hamid Sadeghian, Fan Wu, and Sami Haddadin

***Abstract*— Despite recent advancements in torque-controlled tactile robots, integrating them into manufacturing settings remains challenging, particularly in complex environments. Simplifying robotic skill programming for non-experts is crucial for increasing robot deployment in manufacturing. This work proposes an innovative approach, Vision-Augmented Unified Force-Impedance Control (VA-UFIC), aimed at intuitive visuo-tactile exploration of unknown 3D curvatures. VA-UFIC stands out by seamlessly integrating vision and tactile data, enabling the exploration of diverse contact shapes in three dimensions, including point contacts, flat contacts with concave and convex curvatures, and scenarios involving contact loss. A pivotal component of our method is a robust online contact alignment monitoring system that considers tactile error, local surface curvature, and orientation, facilitating adaptive adjustments of robot stiffness and force regulation during exploration. We introduce virtual energy tanks within the control framework to ensure safety and stability, effectively addressing inherent safety concerns in visuo-tactile exploration. Evaluation using a Franka Emika research robot demonstrates the efficacy of VA-UFIC in exploring unknown 3D curvatures while adhering to arbitrarily defined force-motion policies. Our evaluation encompasses various metrics, including contact alignment monitoring accuracy, real-time feedback latency, computational efficiency, and control performance. These metrics provide comprehensive insights into the effectiveness and practicality of VA-UFIC in real-world manufacturing scenarios. By seamlessly integrating vision and tactile sensing, VA-UFIC offers a promising avenue for intuitive exploration of complex environments, with potential applications spanning manufacturing, inspection, and beyond.**

I. Introduction

Robotic systems have become indispensable in industrial operations, excelling in tasks demanding repetitive speed and precision. However, challenges persist when these systems confront tasks requiring nuanced force and compliance control, such as polishing car doors or carving metal. Despite advancements in torque-controlled tactile robots, their deployment for tactile and flexible interaction remains limited due to the expertise required in control implementation [1].

To enhance the deployment of tactile robots, the development of straightforward and intuitive robot skill programming methods is essential to alleviate the need for intricate tailoring and adjustment of software programs according to the task specifications of each application. In traditional factory settings, industry experts experienced in standard automation processes program the machines, such as CNC machines,[1] to perform required motion or force to deliver high-quality operations [2]. However, although robotics has made vast progress in force-motion interaction, including impedance, force, and unified controls [3]–[5], in flexible manufacturing where frequent reconfiguration is common, it remains difficult to efficiently program the robots while adhering to desired forces and motions derived from task and process requirements. Moreover, deploying robots in highly variable environments, such as small batch-size production, requires fine-tuning robot controllers to adapt to changing environmental features and constraints [6]–[8].

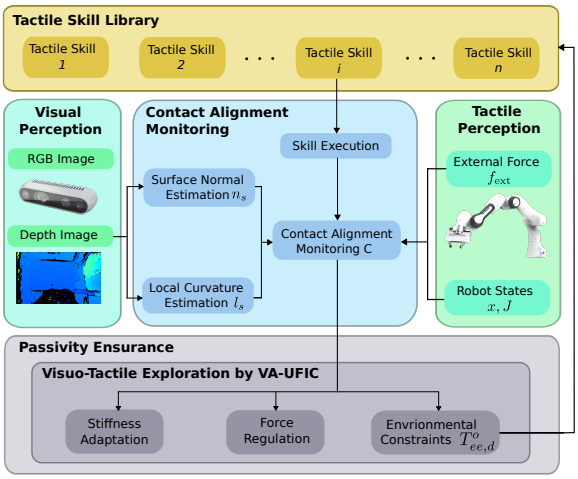


Fig. 1: **Visuo-Tactile Exploration of Unknown Rigid 3D Curvatures by vision-augmented unified force-impedance control (VA-UFIC) for a chosen tactile skill.** Visuo-tactile exploration is the next step to achieving a force-motion planning framework that outputs an object-centric force-motion profile for an arbitrary tactile skill policy. The explored environment is fed back to the library to further plan the force-motion policy.

To achieve more natural and intuitive robot programming to broaden robot deployment in manufacturing, it is desirable to autonomously explore environmental features for a given arbitrary force-motion policy and use the explored environment information to plan the object-centric force-motion policy, as shown in Fig. 1. Methods such as the operational space framework, constrained-based task specifications, and object-centric representations constitute significant steps towards a user-friendly programming paradigm [9]–[12]. However, directly producing or planning the object-centric force-motion policy for a non-control expert, given an arbitrary force-motion policy, requires autonomous investigation of the environmental constraints experienced by the tools, such as surface curvatures or normal, during task execution. This approach would allow non-experts to use controllers, leveraging environment exploration and analysis of current surface constraints.

Integrating visual and tactile sensors for contact alignment monitoring, like an intelligent end-effector, offers a promising solution to enhance robots' environmental awareness, particularly in exploring unknown surface constraints such as curvatures. While visual perception enables robots to perceive environmental details without touching, tactile sensors provide unique insights into force and moments not discernible through vision alone [13], [14]. However, challenges arise when irregularities occur outside the camera's field of view, i.e., the camera's view is blocked in the contact point or when tactile sensors fail to sense forces and moments due to the point contact or even loss of contact, as

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 and the research and development program under grant no. DIK0249). The authors are with the Chair of Robotics and Systems Intelligence, MIRMI - Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, Germany. kuebra.karacan@tum.de

[1]Using CNC milling as an example, a standardized calculation process generates the tool path based on surface geometry, material feed rate, and cutting speed.

presented in Fig. 2. In other words, different contact shapes dictate the sensing modality for perceiving environmental features. Thus, unifying visual and tactile sensors to monitor the contact alignment between the tool and surface presents a more comprehensive solution involving various contact shapes in real-world applications. Approaches in robotics that synergize visual perception and tactile sensing vary, focusing on enhancing grasp stability, evaluating object shapes, or executing manipulation tasks based on predefined structures such as manipulation graphs or computer-aided design models [15]–[19]. Despite advancements in visuo-tactile capabilities, using those methods in environment exploration is mainly limited in 2D for specific contact shapes, persisting in a gap between current robotic capabilities and real-world application demands [20]–[22].

This paper aims to bridge the disparity between the existing abilities of robots and the requirements posed by real-world scenarios, proposing a novel approach towards developing simple yet effective and intuitive robotic skill programming that does not necessitate specialized control expertise for application: visuo-tactile exploration of unknown rigid 3D curvatures through vision-augmented unified force-impedance control (VA-UFIC). By seamlessly integrating tactile and vision data to span various contact shapes between the tool and the environment, we develop a robust online contact alignment monitoring system, considering factors, e.g., tactile error, local surface curvature, and surface orientation. This information is seamlessly integrated into a vision-augmented unified force-impedance control framework, enabling the adjustment of robot stiffness and force regulation while exploring unknown rigid 3D curvatures. Visuo-tactile exploration is the next step to completing a force-motion planning framework that outputs an object-centric force-motion profile for an arbitrary tactile skill policy.

The contributions of this work include:

I The introduction of online contact alignment monitoring to include various contact shapes between the tool and the environment: combining tactile error, the contact surface's local curvature, and surface orientation derived from tactile and vision data.

II Visuo-tactile exploration of unknown rigid 3D curvatures: integration of contact alignment monitoring into vision-augmented unified force-impedance control to adapt the robot's stiffness and regulate the force profile.

III Implementing virtual energy tanks to ensure system passivity and stability.

IV Evaluation of the proposed method's performance regarding contact alignment monitoring accuracy, real-time feedback latency, computational efficiency, and control performance using a Franka Emika research robot wiping challenging curvatures.

The remainder of the paper is organized as follows. Section II delineates the problem under consideration. Section III presents the methodology, including visuo-tactile exploration of unknown rigid 3D curvatures through contact alignment monitoring using tactile data and vision. Additionally, it covers the passivity-based stability analysis for vision-augmented unified force-impedance control and the implementation of virtual energy tanks for stabilizing the system with variable stiffness and force regulation. The experimental protocol and corresponding results are detailed in Sections IV and V, respectively. Finally, Section VI provides the paper's conclusion.

II. Problem Statement

Robotic manipulation presents many challenges, particularly in adapting predefined contact-rich skills to diverse contexts, as encountered in real-world operations. For instance, polishing strategies designed for flat surfaces may experience difficulties when applied to curved surfaces, where maintaining perpendicular alignment of the manipulation tool is imperative for uniform pressure distribution and consistent cleaning or polishing without causing damage.

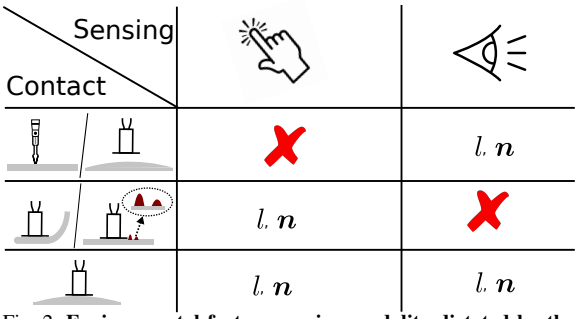


Fig. 2: **Environmental feature sensing modality dictated by the contact shapes.** Point contact or no contact/loss of contact: The surface curvature l and normal vector $\boldsymbol{n}$ can be sensed only by a camera due to lack of force and moments. Concave surface or flat contact with small surface irregularities: The tactile sensor is more effective, where it can sense l and $\boldsymbol{n}$ through contact forces and moments. Conversely, such obstacles are hard to capture by visual sensing, either for lying outside of the sensing area or being too small. Flat contact with convex curvature: Vision or tactile sensor can sense l and $\boldsymbol{n}$ equally efficiently.

Furthermore, the choice of sensing modality for perceiving environmental features depends heavily on the contact shape encountered during manipulation tasks. While cameras excel in discerning surface curvature and normal vectors for pointy tools or even no contact, tactile sensors prove more effective for contact shapes such as concave surfaces or small objects, where they can sense curvature and surface normal through contact forces and moments [23]. Thus, contact alignment monitoring should involve unifying vision and tactile data to span the possible contact scenarios. Additionally, reliance on prior environmental knowledge can lead to unstable robot control and unsafe behaviors in dynamic settings, where sudden deviations from expected contact alignment may result in unintended movements, posing safety risks and potentially damaging equipment or surroundings.

To address these challenges, visuo-tactile exploration of unknown rigid 3D curvatures by VA-UFIC is a promising solution. By detecting deviations in contact alignment and adjusting the end-effector's configuration, robots can maintain desired contacts and identify surface curvatures. Ensuring system stability is crucial to mitigate safety concerns and maintain consistent performance.

The main assumptions made throughout this study can be listed as:

I Highly irregular curvatures are excluded from the study scope due to potential challenges for both sensors to accurately measure surface properties.

II The minimum distance of the depth camera is not violated due to an abrupt change of surface geometry.

III The camera is positioned to observe the current region of interest without predicting future curvatures.

III. Methodology

The methodology begins with designing and implementing unified force-impedance control, a well-established technique governing the robot's response to external forces while ensuring high compliance. This control framework integrates motion and force profiles to facilitate precise environmental interaction. Next, we explore the integration of tactile and vision inputs for contact alignment monitoring. This involves developing algorithms to interpret tactile data and vision cues to comprehensively understand the environment's geometry, e.g., curvatures. Using this sensory information as a foundation, we propose a visuo-tactile exploration of unknown rigid 3D curvatures by vision-augmented unified force-impedance control (VA-UFIC). This framework allows the robot to dynamically adjust its posture and modify

stiffness, motion, and force policies to effectively respond to local faults during interactions with challenging surfaces. A thorough passivity-based stability analysis is conducted to ensure stability, identifying potential instabilities arising from variations in stiffness and force regulations. Additionally, we integrate virtual energy tanks into the control system to provide stability guarantees, particularly in the face of dynamic changes. By implementing the visuo-tactile exploration framework, we aim to develop simple yet effective and intuitive robotic skill definitions that do not necessitate specialized control expertise for application.

A. **Control Design**

For an n-DOF robot manipulator under unified force-impedance control during contact with gravity compensation, the Lagrangian dynamics is

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_{\mathrm{c}} + \boldsymbol{\tau}_{\mathrm{ext}}\,, \tag{1}$$

$$\boldsymbol{\tau}_{\mathrm{c}} = \boldsymbol{\tau}_{\mathrm{i}} + \boldsymbol{\tau}_{\mathrm{f}} + \boldsymbol{\tau}_{\mathrm{g}}\,, \tag{2}$$

where $\boldsymbol{\tau}_{\mathrm{ext}} \in \mathbb{R}^n$ represents the external torque exerted on the robot, while $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ denotes the robot mass matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^{\mathrm{n}}$ signifies the Coriolis and centrifugal vector, and $\boldsymbol{g} \in \mathbb{R}^{\mathrm{n}}$ stands for the gravity vector in joint space. Additionally, $\boldsymbol{\tau}_{\mathrm{c}} \in \mathbb{R}^{\mathrm{n}}$ represents the control torque applied by the robot, which encompasses the torque command for controlling motion and force explicitly and separately, with $\boldsymbol{\tau}_{\mathrm{g}} \in \mathbb{R}^{\mathrm{n}}$ representing gravity compensation. Moreover, $\boldsymbol{\tau}_{\mathrm{i}}$ and $\boldsymbol{\tau}_{\mathrm{f}} \in \mathbb{R}^{\mathrm{n}}$ denote torques individually introduced by impedance and force control, respectively. Subsequently, we develop a control algorithm for the input torque $\boldsymbol{\tau}_{\mathrm{c}}$ to execute the desired tactile manipulation skill. This proposed control law for adaptive tactile skills extends from unified force-impedance control [5], [24]. Unified force-impedance control governs the robot's response to external forces, ensuring compliance while following motion and force profiles separately and explicitly. Starting with the robot's dynamics equation in Cartesian space

$$\boldsymbol{M}_{\mathrm{C}}\ddot{\boldsymbol{x}} + \boldsymbol{C}_{\mathrm{C}}\dot{\boldsymbol{x}} + \boldsymbol{g}_{\mathrm{C}} = \boldsymbol{f}_{\mathrm{c}} + \boldsymbol{f}_{\mathrm{ext}}\,, \tag{3}$$

where

$$\boldsymbol{M}_{\mathrm{C}} = \boldsymbol{J}^{\#\mathrm{T}}\boldsymbol{M}\boldsymbol{J}^{\#}\,, \tag{4}$$

$$\boldsymbol{C}_{\mathrm{C}} = \boldsymbol{J}^{\#\mathrm{T}}\boldsymbol{C}\boldsymbol{J}^{\#}\,, \tag{5}$$

$$\boldsymbol{g}_{\mathrm{C}} = \boldsymbol{J}^{\#\mathrm{T}}\boldsymbol{g}\,. \tag{6}$$

The external wrench to the base frame is denoted as $\boldsymbol{f}_{\mathrm{ext}} \in \mathbb{R}^6$. The robot mass matrix is represented as $\boldsymbol{M}_{\mathrm{C}}(\boldsymbol{q})$, where $\boldsymbol{q}$ is the joint configuration. The Coriolis and centrifugal effects are captured by $\boldsymbol{C}_{\mathrm{C}}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{6\times6}$, and $\boldsymbol{g}_{\mathrm{C}}$ denotes the gravity vector in Cartesian space. Additionally, $\boldsymbol{f}_{\mathrm{c}}$ represents the wrench applied by the robot, which is related to the joint control torque $\boldsymbol{\tau}_{\mathrm{c}} \in \mathbb{R}^{\mathrm{n}}$ through the relationship $\boldsymbol{\tau}_{\mathrm{c}} = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\boldsymbol{f}_{\mathrm{c}}$, where $\boldsymbol{J} \in \mathbb{R}^{6\times\mathrm{n}}$ is the robot Jacobian matrix, and $\boldsymbol{J}^{\#}$ is the pseudo-inverse of the Jacobian. Compliance control, a subset of impedance control, omits inertia shaping and consequently excludes feedback of external forces. The compliance behavior is characterized by a time-varying stiffness matrix $\boldsymbol{K}_{\mathrm{C}}(t) \in \mathbb{R}^{6\times6}$ and damping behavior determined by a positive definite matrix $\boldsymbol{D}_{\mathrm{C}} \in \mathbb{R}^{6\times6}$. Moreover, $\boldsymbol{x} \in \mathbb{R}^6$ denotes the current pose of the end-effector in the base frame, and the pose error is denoted by $\tilde{\boldsymbol{x}}$. A conventional compliance controller for motion tracking can be formulated as

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_{\mathrm{d}}\,, \tag{7}$$

$$\boldsymbol{f}_{\mathrm{i}} = -\boldsymbol{K}_{\mathrm{C}}(t)\tilde{\boldsymbol{x}} - \boldsymbol{D}_{\mathrm{C}}\dot{\boldsymbol{x}}\,, \tag{8}$$

$$\boldsymbol{\tau}_{\mathrm{i}} = \boldsymbol{J}^{T}\boldsymbol{f}_{\mathrm{i}}\,. \tag{9}$$

The force control is established to maintain the target contact force in the task space $\boldsymbol{f}_{\mathrm{d}}^{\mathrm{ee}} \in \mathbb{R}^6$, exerted by the robot concerning the external force $\boldsymbol{f}_{\mathrm{ext}}^{\mathrm{ee}} \in \mathbb{R}^6$, as follows:

$$\boldsymbol{\tau}_{\mathrm{f}} = \boldsymbol{J}(\boldsymbol{q})^{\mathrm{T}}\boldsymbol{f}_{\mathrm{f}}, \tag{10}$$

$$\boldsymbol{f}_{\mathrm{f}} = \begin{bmatrix} [\boldsymbol{R}_{\mathrm{ee}}^{0}]_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & [\boldsymbol{R}_{\mathrm{ee}}^{0}]_{3\times3} \end{bmatrix} (\boldsymbol{f}_{\mathrm{d}}^{\mathrm{ee}} + \boldsymbol{K}_{p}\,\tilde{\boldsymbol{f}}_{\mathrm{ext}}^{\mathrm{ee}} + \boldsymbol{K}_{\mathrm{i}}\int_{0}^{t}\tilde{\boldsymbol{f}}_{\mathrm{ext}}^{\mathrm{ee}}\,d\sigma)\,, \tag{11}$$

$$\tilde{\boldsymbol{f}}_{\mathrm{ext}}^{\mathrm{ee}} = \boldsymbol{f}_{\mathrm{ext}}^{\mathrm{ee}} - \boldsymbol{f}_{\mathrm{d}}^{\mathrm{ee}}\,, \tag{12}$$

In this context, $\boldsymbol{f}_{\mathrm{f}} \in \mathbb{R}^6$ represents a feed-forward and feedback force controller in the base frame, which has been rotated by $\boldsymbol{R}_{\mathrm{ee}}^{0}$. The proportional-integral (PI) controller gains are defined by the diagonal matrices $\boldsymbol{K}_{\mathrm{p}}$ and $\boldsymbol{K}_{\mathrm{i}} \in \mathbb{R}^{6\times6}$. The resultant control torque without the gravity compensation for unified force-impedance control $\boldsymbol{\tau} \in \mathbb{R}^n$ is

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{f}} + \boldsymbol{\tau}_{\mathrm{i}}\,. \tag{13}$$

Next, we introduce contact alignment monitoring based on visual and tactile data to explore unknown rigid 3D curvatures for an arbitrary force-motion policy in the end-effector frame. This rich sensory information is augmented to unified force-impedance control so that the control parameters, such as stiffness and contact force shaping function, are decided. Thus, the robot can maintain contact with the current surface geometry and orientation.

B. **Visuo-Tactile Exploration of Unknown Rigid 3D Curvatures by VA-UFIC**

To guarantee a successful execution of the desired skill and to understand the environment comprehensively, we monitor the contact alignment that utilizes tactile and visual perception. Based on this rich sensory information, we enable the robot to adjust posture, stiffness, motion, and force policy for local fault recovery during interactions with challenging surfaces at the low-level control.

The visual perception algorithm operates through two concurrent threads: (i) data collection and pre-processing and (ii) surface normal estimation. Initially, depth images are transformed into point clouds for use within the Point Cloud Library [25]. RGB and depth images are acquired from the video stream, precisely aligned, and converted into a 3D point cloud. Subsequently, a surface normal estimation method is applied based on the acquired point cloud data to predict contact surface orientation. Inspired by Westfechtel et al. [26], the region growing method clusters the surface normals within similar orientations to segment the point clouds. Principal Component Analysis (PCA) is then employed on the clustered segments to determine their orientation.

The eigenvectors of the covariance matrix $\boldsymbol{\Sigma}_{3\times3} = [\boldsymbol{e}_1;\boldsymbol{e}_2;\boldsymbol{n}_{\mathrm{s}}^{\mathrm{c}}]$, representing the PCA output, characterize a segment's primary directions. Here, $\boldsymbol{n}_{\mathrm{s}}^{\mathrm{c}} \in \mathbb{R}^{3\times3}$ denotes the surface normal of the segment in the camera frame, which represents the direction of a surface segment, while $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ represent the long and short edges, respectively. Furthermore, the local curvature l_{s} of the working surface can be computed using Equation (16), where $\lambda_{\mathrm{i}}\,, i = 1,2,3$ are the eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$ obtained through PCA. Detected surface normal $\boldsymbol{n}_{\mathrm{s}}^{\mathrm{c}}$ and local curvature l_{s} are illustrated in Fig. 3.

$$\boldsymbol{n} = [0\,0\,1]^{\mathrm{T}}\,, \tag{14}$$

$$\theta = |\cos^{-1}(\boldsymbol{n}_{\mathrm{s}}^{\mathrm{c}\mathrm{T}}\boldsymbol{n})|\,, \tag{15}$$

$$l_{\mathrm{s}} = \left|\frac{\lambda_3}{tr(\boldsymbol{\Sigma})}\right|. \tag{16}$$

The end-effector orientation error θ, representing the devi-

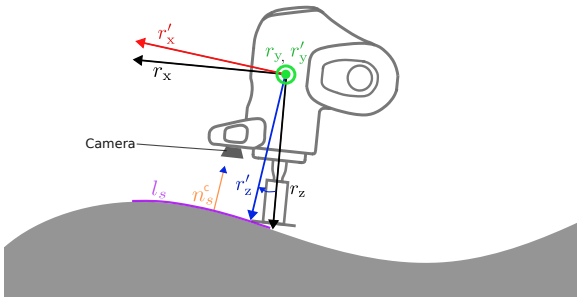


Fig. 3: **Contact Alignment.** Creating the desired end-effector orientation matrix $\boldsymbol{R}^0_{\mathrm{ee,d}}$ based on the detected surface normal $\boldsymbol{n}^{\mathrm{c}}_{\mathrm{s}}$ involves the following steps: Firstly, transfer the detected surface normal $\boldsymbol{n}^{\mathrm{c}}_{\mathrm{s}}$ from camera frame into robot base frame to get $\boldsymbol{n}^0_{\mathrm{s}}$. Next, apply detection $\boldsymbol{n}^0_{\mathrm{s}}$ as new reference z-axis r'_z and project the current end-effector x-axis r_{x} onto the orthogonal surface of the estimated z-axis r'_{z} to get r'_{x}. Lastly, generate the adapted y-axis r'_{y} through the cross-product of r'_{x} and assemble the desired end-effector rotation matrix $\boldsymbol{R}^0_{\mathrm{ee,d}}$ as in (27).

ation in surface normal between $\boldsymbol{n}^{\mathrm{c}}_{\mathrm{s}}$ captured by the camera and the z-axis of the camera (aligned with z-axis of the end-effector), can be determined using the acos function in (15). Undesired contacts lead to deviations from the desired pose, manifesting as either a pose error $\tilde{\boldsymbol{x}}^{\mathrm{ee}} \in \mathbb{R}^6$ or external forces $\boldsymbol{f}^{\mathrm{ee}}_{\mathrm{ext}} \in \mathbb{R}^6$ at the end-effector. In real-time, contact alignment monitoring accumulates all the error terms and their corresponding signal strengths as presented in (17). Simultaneously, the signal strengths for the tactile error, surface normal deviation, and local curvature term, denoted as α, ξ, and γ respectively, contribute to the adaptive process and decide how agile the robot reacts to them. In (18), the contact alignment monitoring C is employed to calculate a normalized coefficient h.

$$\mathrm{C} = |\alpha|\boldsymbol{f}^{\mathrm{eeT}}_{\mathrm{ext}}\tilde{\boldsymbol{x}}^{\mathrm{ee}}| + \xi\theta + \gamma l_{\mathrm{s}}|\,, \tag{17}$$

$$\mathrm{h} = 1 - \frac{\mathrm{C}}{\mathrm{C_m}}\,. \tag{18}$$

The contact alignment margin $\mathrm{C_m}$ is crucial for compensating for minor environmental effects, such as surface friction and measurement inaccuracy, and, notably, employing position rather than velocity or acceleration results in a less noisy signal. The normalized metric h is subsequently linked to the maximum stiffness level at the end-effector frame $\boldsymbol{K}^{\mathrm{ee}}_{\mathrm{max,t}}$ through ρ_{align} and it is rotated back to the base frame by the rotation matrix $\boldsymbol{R}^0_{\mathrm{ee}}$. This inherent behavior is leveraged to robustly respond to undesired contacts and reconfigure the end-effector through adaptive adjustments to the stiffness matrix in the translational directions $\boldsymbol{K}_{\mathrm{C,t}}$.

$$\boldsymbol{K}_{\mathrm{C,t}} = \rho_{\mathrm{align}}\boldsymbol{R}^0_{\mathrm{ee}}\boldsymbol{K}^{\mathrm{ee}}_{\mathrm{max,t}}\,. \tag{19}$$

The alignment parameter ρ_{align} is extended based on studies [5], [23], as outlined in (20).

$$\dot{\rho}_{\mathrm{align}} = \begin{cases} \min\{\rho, 0\}\,, & \rho_{\mathrm{align}} = 1 \\ \rho\,, & 0 < \rho_{\mathrm{align}} < 1,\ \rho_{\mathrm{align}}(0) = 0, \\ \max\{\rho, 0\}\,, & \rho_{\mathrm{align}} = 0 \end{cases} \tag{20}$$

and ρ is given by

$$\rho = \mathrm{h}\rho_{\mathrm{align}} + \rho_{\mathrm{min}}\,. \tag{21}$$

It's important to note that, to ensure an initial increment when $\rho_{\mathrm{align}} = 0$, a small positive constant ρ_{min} is introduced into the dynamics of the shaping function. When the robot is entirely compliant, meaning ρ_{align} equals zero, it becomes capable of adapting to the environment. This implies that the translational component of the actual end-effector pose $\boldsymbol{x}_{\mathrm{ee,t}} \in \mathbb{R}^3$ is fed back to the controller as the desired translational pose $\boldsymbol{x}_{\mathrm{d,t}}$. Subsequently, we calculate the rotation of the end-effector $\boldsymbol{R}^0_{\mathrm{ee,d}} \in \mathbb{R}^{3\times3}$ to adapt to the environment.

Upon detecting a significant deviation from the intended contact alignment between the robotic tool and the surface, often resulting from abrupt changes in the contact, the robot becomes compliant in translational directions. Afterward, it realigns itself with the detected surface normal in the camera frame, denoted as $\boldsymbol{n}^c_{\mathrm{s}}$, and regenerates the motion and force policy. This process necessitates the knowledge of the desired end-effector orientation $\boldsymbol{R}^0_{\mathrm{ee,d}}$, which can be computed from the detected surface normal $\boldsymbol{n}^c_{\mathrm{s}}$, as shown in Fig. 3. The contact surface normal is initially transformed from the camera frame to the end-effector frame, then to the robot base frame using (22). $\boldsymbol{R}^{\mathrm{ee}}_{\mathrm{c}}$ represents the rotation matrix that transfers from the camera frame to the end-effector frame. Similarly, $\boldsymbol{R}^0_{\mathrm{c}}$ denotes the rotation matrix that transfers from the camera frame to the robot base frame. Subsequently, the surface normal $\boldsymbol{n}^0_{\mathrm{s}}$ in the base frame contributes to the construction of the desired orientation matrix $\boldsymbol{R}^0_{\mathrm{ee,d}}$ through the following steps: i.) read the current orientation of the end-effector and extract the first column $\boldsymbol{r}_{\mathrm{x}}$; ii.) project $\boldsymbol{r}_{\mathrm{x}}$ onto the orthogonal plane of the surface normal, as per (25); iii.) calculate the second column $\boldsymbol{r}_{\mathrm{y}}$ through the cross product of the projected $\boldsymbol{r}^{'}_{\mathrm{x}}$ and surface normal $\boldsymbol{n}^0_{\mathrm{s}}$; iv.) assemble these three distinct components into the desired rotational matrix $\boldsymbol{R}^0_{\mathrm{ee,d}}$.

$$\boldsymbol{R}^0_{\mathrm{c}} = \boldsymbol{R}^0_{\mathrm{ee}}\boldsymbol{R}^{\mathrm{ee}}_{\mathrm{c}}\,, \boldsymbol{n}^0_{\mathrm{s}} = \boldsymbol{R}^0_{\mathrm{c}}\boldsymbol{n}^{\mathrm{c}}_{\mathrm{s}}\,, \tag{22}$$

$$\boldsymbol{R}^0_{\mathrm{ee}} = [[\boldsymbol{r}_{\mathrm{x}}]_{3\times1} \quad [\boldsymbol{r}_{\mathrm{y}}]_{3\times1} \quad [\boldsymbol{r}_{\mathrm{z}}]_{3\times1}]\,, \tag{23}$$

$$\boldsymbol{r}'_{\mathrm{z}} = \boldsymbol{n}^0_{\mathrm{s}}, \tag{24}$$

$$\boldsymbol{r}^{'}_{\mathrm{x}} = \boldsymbol{r}_{\mathrm{x}} - (\boldsymbol{r}^{\mathrm{T}}_{\mathrm{x}}\boldsymbol{n}^0_{\mathrm{s}})\boldsymbol{n}^0_{\mathrm{s}}\,, \tag{25}$$

$$\boldsymbol{r}^{'}_{\mathrm{y}} = \boldsymbol{r}^{'}_{\mathrm{z}} \times \boldsymbol{r}^{'}_{\mathrm{x}}\,, \tag{26}$$

$$\boldsymbol{R}^0_{\mathrm{ee,d}} = \left[[\boldsymbol{r}^{'}_{\mathrm{x}}]_{3\times1} \quad [\boldsymbol{r}^{'}_{\mathrm{y}}]_{3\times1} \quad [\boldsymbol{r}^{'}_{\mathrm{z}}]_{3\times1}\right]\,. \tag{27}$$

To ensure that the input signal provided to the robot is smooth and continuous, a low-pass filter is implemented, facilitating the gradual transition of the signal $\boldsymbol{R}^0_{\mathrm{input}}$ from the initial rotation $\boldsymbol{R}^0_{\mathrm{init}}$ to the desired rotation $\boldsymbol{R}^0_{\mathrm{ee,d}}$. The scaling coefficient ζ falls within the range of 0 to 1, and T represents the time interval governing the convergence of the low-pass filter. Specifically, at $t = 0$, signifying the initiation of contact alignment, the output is the original rotation $\boldsymbol{R}^0_{\mathrm{init}}$. Conversely, when $t = T$, indicating the completion of convergence, the input rotation to the robot becomes $\boldsymbol{R}^0_{\mathrm{ee,d}}$.

$$\zeta = \frac{t}{T}, 0 \le t \le T\,, \tag{28}$$

$$\boldsymbol{R}^0_{\mathrm{input}} = (\boldsymbol{R}^0_{\mathrm{ee,d}}\boldsymbol{R}^{0\mathrm{T}}_{\mathrm{init}})^{\zeta}\boldsymbol{R}^0_{\mathrm{init}}\,. \tag{29}$$

Furthermore, we formulate the force shaping function ρ_{frc}. This function facilitates the alignment of the commanded force to compensate for tool alignment errors and mitigate the undesired loss of contacts. The robot accommodates the tool alignment error $\boldsymbol{f}^{\mathrm{eeT}}_{\mathrm{d}}\tilde{\boldsymbol{x}}^{\mathrm{ee}}$ during contact loss within the error margin $\delta_{\mathrm{c}} > 0$. Additionally, in cases where the robot loses surface contact due to a substantial tool alignment error, it transitions to impedance control, adhering solely to the desired motion.

$$\rho_{\mathrm{frc}} = \begin{cases} 1\,, & \boldsymbol{f}^{\mathrm{eeT}}_{\mathrm{d}}\tilde{\boldsymbol{x}}^{\mathrm{ee}} \le 0 \\ 0.5(1 + \cos((\pi\frac{\tilde{x}^{\mathrm{ee}}_{\mathrm{z}}}{\delta_{\mathrm{c}}}))\,, & 0 < \boldsymbol{f}^{\mathrm{eeT}}_{\mathrm{d}}\tilde{\boldsymbol{x}}^{\mathrm{ee}} \wedge \\ & 0 < \tilde{x}^{\mathrm{ee}}_{\mathrm{z}} \le \delta_{\mathrm{c}} \\ 0\,, & \text{otherwise.} \end{cases} \tag{30}$$

$$\boldsymbol{\tau}_{\mathrm{f}} = \rho_{\mathrm{frc}} \boldsymbol{J}^{\mathrm{T}} \boldsymbol{f}_{\mathrm{f}} \,. \tag{31}$$

Finally, after calculating the orientation matrix, the desired trajectory and force policy can be adapted accordingly in the base frame, as shown in Alg. 1. However, stiffness variation in the impedance controller and, in case of loss of contact, the force controller may jeopardize the controller's stability, resulting in unsafe behaviour [27]. Next, passivity-based stability analysis for vision-augmented unified force-impedance control is implemented with virtual energy tanks for variable stiffness and force regulation to ensure the system's passivity and stability.

Algorithm 1: Visuo-Tactile Exploration by VA-UFIC

Input : $\boldsymbol{x}_{\mathrm{d}}^{\mathrm{ee}}$, $\boldsymbol{f}_{\mathrm{d}}^{\mathrm{ee}}$, $\boldsymbol{f}_{\mathrm{ext}}$, $\boldsymbol{x}$, $\boldsymbol{n}_{\mathrm{s}}^{\mathrm{c}}$, θ, l_{s}
Output: $\boldsymbol{T}_{\mathrm{ee,d}}^{0}$, $\boldsymbol{x}_{\mathrm{d}}$, $\boldsymbol{f}_{\mathrm{d}}$
Receive tactile skill from the task state machine;
while *not skill_finished* **do**
  Read robot pose $\boldsymbol{x}$ and external force $\boldsymbol{f}_{\mathrm{ext}}$;
  Compute robot pose error $\tilde{\boldsymbol{x}}^{\mathrm{ee}} = \boldsymbol{x}^{\mathrm{ee}} - \boldsymbol{x}_{\mathrm{d}}^{\mathrm{ee}}$;
  Compute end-effector orientation error θ and local curvature l_{s} by (15) and (16);
  Monitor contact alignment C;
  $\dot{\rho}_{\mathrm{align}} = \frac{(\mathrm{C_m}-\mathrm{C})}{\mathrm{C_m}} \rho_{\mathrm{align}}$;
  if $0 < \rho_{\mathrm{align}} < 1$ **then**
    $\boldsymbol{K}_{\mathrm{C,t}}^{\mathrm{ee}} = \rho_{\mathrm{align}} \boldsymbol{K}_{\mathrm{max,t}}^{\mathrm{ee}}$;
    Alignment begins, read surface normal $\boldsymbol{n}_{\mathrm{s}}^{\mathrm{c}}$;
    Using $\boldsymbol{n}_{\mathrm{s}}^{\mathrm{c}}$, construct $\boldsymbol{R}_{\mathrm{ee,d}}^{0}$;
    $\boldsymbol{T}_{\mathrm{ee,d}}^{0} \leftarrow \boldsymbol{R}_{\mathrm{ee,d}}^{0}$ and $\boldsymbol{x}_{\mathrm{ee,t}}$;
    Compute the object-centric tactile policy $\boldsymbol{x}_{\mathrm{d}}, \boldsymbol{f}_{\mathrm{d}}$ for the explored environment $\boldsymbol{T}_{\mathrm{ee,d}}^{0}$;

C. Passivity-Based Stability Analysis and Installing Virtual Energy Tanks

Virtual energy tanks are integrated to guarantee stability by identifying potential instabilities arising from stiffness variations and force regulations to ensure stability even amidst dynamic changes [28], [29]. To show the passivity, the storage function S_{r} for the Cartesian robot dynamics that represents the kinetic energy of the robot is

$$S_{\mathrm{r}} = \frac{1}{2} \dot{\boldsymbol{x}}^{T} \boldsymbol{M}_{\mathrm{C}} \dot{\boldsymbol{x}} \,, \tag{32}$$

where the time derivative of the storage function $\dot{S}_{\mathrm{r}}$ is

$$\dot{S}_{\mathrm{r}} = \dot{\boldsymbol{x}}^{T} (\boldsymbol{f} + \boldsymbol{f}_{\mathrm{ext}}) \;\text{ or }\; \dot{S}_{\mathrm{r}} = \dot{\boldsymbol{q}}^{T} (\boldsymbol{\tau} + \boldsymbol{\tau}_{\mathrm{ext}}) \,, \tag{33}$$

which we can say that it is passive for the pair $(\boldsymbol{\tau} + \boldsymbol{\tau}_{\mathrm{ext}}, \dot{\boldsymbol{q}})$. Identifying potential instabilities arising from stiffness variations and force regulations, we split the problem of analyzing the stability of robot dynamics into two cases: without contact (Case I) and during contact (Case II).

*1) **Case I: Stability analysis without any contact**:* When there is no contact, the stiffness $\boldsymbol{K}_{\mathrm{C}}$ remains constant. Thus, only the force controller may cause instability. The stability of the force controller can be assessed using the subsequent storage function S_{f}

$$S_{\mathrm{f}} = \frac{1}{2} \tilde{\boldsymbol{x}}^{T} \boldsymbol{K}_{\mathrm{C}} \tilde{\boldsymbol{x}} \,, \tag{34}$$

$$\dot{S}_{\mathrm{f}} = \dot{\boldsymbol{x}}^{T} \boldsymbol{K}_{\mathrm{C}} \tilde{\boldsymbol{x}} \,, \tag{35}$$

$$= \dot{\boldsymbol{x}}^{T} (-\boldsymbol{f} + \boldsymbol{f}_{\mathrm{f}} - \boldsymbol{D}_{\mathrm{C}} \dot{\boldsymbol{x}}) \,. \tag{36}$$

Due to the pair of $(\boldsymbol{f}_{\mathrm{f}}, \dot{\boldsymbol{x}})$, the force controller should be modified to guarantee stability by augmenting a virtual energy tank such that

$$\boldsymbol{f} = -\boldsymbol{D}_{\mathrm{C}} \dot{\boldsymbol{x}} - \boldsymbol{K}_{\mathrm{C}} \tilde{\boldsymbol{x}} + \lambda \boldsymbol{f}_{\mathrm{f}} + \boldsymbol{f}_{\mathrm{f,var}} \,. \tag{37}$$

To indicate if the force controller $\boldsymbol{f}_{\mathrm{f}}$ is passive, λ is used such that

$$\lambda = \begin{cases} 1, & \dot{\boldsymbol{x}}^{T} \boldsymbol{f}_{\mathrm{f}} < 0 \,, \\ 0, & else \end{cases} \tag{38}$$

Moreover, $\boldsymbol{f}_{\mathrm{f,var}}$ is the modification in the controller regulated by the tank. Being the tank's energy is $S_{\mathrm{t,f}}$, we design its dynamics $\dot{x}_{\mathrm{t,f}}$ as

$$\dot{x}_{\mathrm{t,f}} = \lambda \beta_{\mathrm{f}} \frac{\dot{\boldsymbol{x}}^{T} \boldsymbol{D}_{\mathrm{C}} \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}^{T} \boldsymbol{f}_{\mathrm{f}}}{x_{\mathrm{t,f}}} + u_{\mathrm{t,f}} \,, \tag{39}$$

$$y_{\mathrm{t,f}} = x_{\mathrm{t,f}} \,, \tag{40}$$

$$S_{\mathrm{t,f}} = \frac{1}{2} x_{\mathrm{t,f}}^{2} \,, \tag{41}$$

where $x_{\mathrm{t,f}}$, $u_{\mathrm{t,f}}$, and $y_{\mathrm{t,f}}$ are the tank's state, input, and out variable, respectively. The tank is interconnected to the controllers through the power-preserving Dirac structure

$$\begin{bmatrix} \boldsymbol{f}_{\mathrm{f,var}} \\ u_{\mathrm{t,f}} \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{\omega}_{\mathrm{f}} \\ -\boldsymbol{\omega}_{\mathrm{f}}^{T} & 0 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}} \\ y_{\mathrm{t,f}} \end{bmatrix} \,, \tag{42}$$

$$\boldsymbol{\omega}_{\mathrm{f}} = \frac{\sigma(S_{\mathrm{t,f}})(1-\lambda)\boldsymbol{f}_{\mathrm{f}}}{y_{\mathrm{t,f}}} \,. \tag{43}$$

The design parameter $\boldsymbol{\omega}_{\mathrm{f}}$ is a modulating factor that controls the power transmission between the tank and the controller with the valve $\sigma(S_{\mathrm{t,f}})$ is

$$\sigma(S_{\mathrm{t,f}}) = \begin{cases} \sigma(S_{\mathrm{t,f}}) \in (0,1], & S_{\mathrm{t,f}} > \underline{S}_{\mathrm{t,f}} \,, \\ 0, & else \end{cases} \tag{44}$$

The controller can regulate force if the tank is not depleted. Note that, to avoid singularities, we set a lower limit $\underline{S}_{\mathrm{t,f}}$ for the energy threshold in the tank. Additionally, to ensure the tank is not overloaded, a specific upper-limit $\overline{S}_{\mathrm{t,f}}$ for the tank is introduced:

$$\beta_{\mathrm{f}} = \begin{cases} \kappa_{\mathrm{f}} \in [0,1], & S_{\mathrm{t,f}} < \overline{S}_{\mathrm{t,f}} \,, \\ 0, & else \end{cases} \tag{45}$$

where a smooth transition behavior κ_{f} is embedded. Using $\boldsymbol{f}_{\mathrm{f,var}} = \boldsymbol{\omega}_{\mathrm{f}} y_{\mathrm{t,f}}$, the passivity of the subsystem S_{c} involving the tank and the controller with the combined storage function S_{overall}

$$S_{\mathrm{overall}} = S_{\mathrm{c}} + S_{\mathrm{r}} \,,\; S_{\mathrm{c}} = S_{\mathrm{f}} + S_{\mathrm{t,f}} \,, \tag{46}$$

$$\dot{S}_{\mathrm{overall}} = \dot{S}_{\mathrm{c}} + \dot{S}_{\mathrm{r}} \,, \tag{47}$$

$$= -\dot{\boldsymbol{x}}^{T} \boldsymbol{f} + \lambda(1-\beta_{\mathrm{f}}) \dot{\boldsymbol{x}}^{T} \boldsymbol{f}_{\mathrm{f}} - (1-\lambda\beta_{\mathrm{f}}) \dot{\boldsymbol{x}}^{T} \boldsymbol{D}_{\mathrm{C}} \dot{\boldsymbol{x}} + \dot{\boldsymbol{x}}^{T} (\boldsymbol{f} + \boldsymbol{f}_{\mathrm{ext}}) \,, \tag{48}$$

$$= \dot{\boldsymbol{q}}^{T} \boldsymbol{\tau}_{\mathrm{ext}} - (1-\lambda\beta_{\mathrm{f}}) \dot{\boldsymbol{q}}^{T} \boldsymbol{J}^{T} \boldsymbol{D}_{\mathrm{C}} \dot{\boldsymbol{x}} + \lambda(1-\beta_{\mathrm{f}}) \dot{\boldsymbol{q}}^{T} \boldsymbol{\tau}_{\mathrm{f}} \,. \tag{49, 50}$$

The modified unified force-impedance control ensures stability in case of loss of contact:

$$\boldsymbol{f} = -\boldsymbol{D}_{\mathrm{C}} \dot{\boldsymbol{x}} - \boldsymbol{K}_{\mathrm{C}} \tilde{\boldsymbol{x}} + \rho_{\mathrm{frc}} (\lambda + \sigma(S_{\mathrm{t,f}})(1-\lambda)) \boldsymbol{f}_{\mathrm{f}} \,. \tag{51}$$

*2) **Case II: Stability analysis during contact** :* Contact means that while the robot can move in k-dimensions, the motion is constrained in the rest $6-k$ dimensions. Thus, the force controller during contact does not jeopardize stability, as the components of $\dot{\boldsymbol{x}}$ in the force control direction is zero $\dot{\boldsymbol{x}}^{T} \boldsymbol{f}_{\mathrm{f}} = 0$. However, stiffness variation in the impedance controller during contact may cause instability. Next, we present the stability analysis and the virtual energy tank for the impedance controller. To assess the passivity of

this controller, we examine the storage function, which is regarded as the corresponding spring potential S_i

$$S_\mathrm{i} = \frac{1}{2}\tilde{\boldsymbol{x}}^T \boldsymbol{K}_\mathrm{C}\tilde{\boldsymbol{x}}\,, \tag{52}$$

$$\dot{S}_\mathrm{i} = \dot{\boldsymbol{x}}^T \boldsymbol{K}_\mathrm{C}\tilde{\boldsymbol{x}} + \frac{1}{2}\tilde{\boldsymbol{x}}^T \dot{\boldsymbol{K}}_\mathrm{C}\tilde{\boldsymbol{x}}\,, \tag{53}$$

$$= -\dot{\boldsymbol{x}}^T \boldsymbol{f}_\mathrm{i} - \dot{\boldsymbol{x}}^T \boldsymbol{D}_\mathrm{C}\dot{\boldsymbol{x}} + \frac{1}{2}\tilde{\boldsymbol{x}}^T \dot{\boldsymbol{K}}_\mathrm{C}\tilde{\boldsymbol{x}}\,. \tag{54}$$

Passivity with respect to the pair $(\boldsymbol{f}_\mathrm{i}, \dot{\boldsymbol{x}})$ cannot be guaranteed due to the term of $\frac{1}{2}\tilde{\boldsymbol{x}}^T \dot{\boldsymbol{K}}\tilde{\boldsymbol{x}}$. Therefore, we modify the impedance controller $\boldsymbol{f}_\mathrm{i}$ by adding a control term $\boldsymbol{f}_{\mathrm{i,var}}$ regulated by the energy tank

$$\boldsymbol{f}_\mathrm{i} = -\boldsymbol{K}_\mathrm{const}\tilde{\boldsymbol{x}} - \boldsymbol{D}_\mathrm{C}\dot{\boldsymbol{x}} + \boldsymbol{f}_{\mathrm{i,var}}\,, \tag{55}$$

$$\boldsymbol{K}_\mathrm{C} = \boldsymbol{K}_\mathrm{const} + \boldsymbol{K}_\mathrm{var}(t)\,, \tag{56}$$

where the stiffness matrix has constant $\boldsymbol{K}_\mathrm{const}$, which can also be zero, and time-varying parts $\boldsymbol{K}_\mathrm{var}(t) = \rho_\mathrm{align}\boldsymbol{K}_\mathrm{max}$. Energy tank state $x_{\mathrm{t,i}}$, its dynamics $\dot{x}_{\mathrm{t,i}}$, and the tank energy $S_{\mathrm{t,i}}$ are

$$\dot{x}_{\mathrm{t,i}} = \beta_\mathrm{i}\frac{\dot{\boldsymbol{x}}^T\boldsymbol{D}\dot{\boldsymbol{x}}}{x_{\mathrm{t,i}}} + u_{\mathrm{t,i}}\,, \tag{57}$$

$$y_{\mathrm{t,i}} = x_{\mathrm{t,i}}\,, \tag{58}$$

$$S_{\mathrm{t,i}} = \frac{1}{2}x_{\mathrm{t,i}}^2\,, \tag{59}$$

where $u_{\mathrm{t,i}}$ and $y_{\mathrm{t,i}}$ are input and output variable, respectively. To ensure the tank is not overloaded, a specific upper-limit $\overline{S}_{\mathrm{t,i}}$ for the tank is introduced with a smooth transition behavior κ_i

$$\beta_\mathrm{i} = \begin{cases} \kappa_\mathrm{i} \in [0,1], & S_{\mathrm{t,i}} < \overline{S}_{\mathrm{t,i}}\,, \\ 0, & else \end{cases} \tag{60}$$

The Dirac structure for the ports implies the passivity of the system:

$$\begin{bmatrix} \boldsymbol{f}_{\mathrm{i,var}} \\ u_{\mathrm{t,i}} \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{\omega}_\mathrm{i} \\ -\boldsymbol{\omega}_\mathrm{i}^T & 0 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}} \\ y_{\mathrm{t,i}} \end{bmatrix}\,, \tag{61}$$

$$\boldsymbol{\omega}_\mathrm{i} = -\frac{\sigma(S_{\mathrm{t,i}})\boldsymbol{K}_\mathrm{var}(t)\tilde{\boldsymbol{x}}}{y_{\mathrm{t,i}}}\,. \tag{62}$$

The design parameter $\boldsymbol{\omega}_\mathrm{i}$ is a modulating factor that controls the power transmission between the tank and the impedance controller with the valve $\sigma(S_{\mathrm{t,i}})$

$$\sigma(S_{\mathrm{t,i}}) = \begin{cases} \sigma(S_{\mathrm{t,i}}) \in (0,1], & S_{\mathrm{t,i}} > \underline{S}_{\mathrm{t,i}}\,, \\ 0, & else \end{cases} \tag{63}$$

This indicates that the controller can adjust stiffness if the tank has not been depleted. To prevent singularities, we establish a lower limit, denoted as $\underline{S}_{\mathrm{t,i}}$, for the energy threshold in the tank. Furthermore, ensuring the passivity of subsystem S_c, which comprises the tank and the controller, is achieved by combining the storage function in the following:

$$S_\mathrm{c} = \frac{1}{2}\tilde{\boldsymbol{x}}^T\boldsymbol{K}_\mathrm{const}\tilde{\boldsymbol{x}} + \frac{1}{2}x_{\mathrm{t,i}}^2\,, \tag{64}$$

$$\dot{S}_\mathrm{c} = \dot{\boldsymbol{x}}\boldsymbol{K}_\mathrm{const}\tilde{\boldsymbol{x}} + \dot{x}_{\mathrm{t,i}}x_{\mathrm{t,i}}\,. \tag{65}$$

Using $\boldsymbol{f}_{\mathrm{i,var}} = \boldsymbol{\omega}_\mathrm{i}y_{\mathrm{t,i}}$

$$\dot{S}_\mathrm{c} = -\dot{\boldsymbol{x}}^T\boldsymbol{f}_\mathrm{i} - \dot{\boldsymbol{x}}^T\boldsymbol{D}_\mathrm{C}\dot{\boldsymbol{x}} + \dot{\boldsymbol{x}}^T\boldsymbol{\omega}_\mathrm{i}y_{\mathrm{t,i}} + \tag{66}$$

$$\beta_\mathrm{i}\dot{\boldsymbol{x}}^T\boldsymbol{D}_\mathrm{C}\dot{\boldsymbol{x}} - \boldsymbol{\omega}_\mathrm{i}^T\dot{\boldsymbol{x}}x_{\mathrm{t,i}}\,, \tag{67}$$

$$= -\dot{\boldsymbol{q}}^T\boldsymbol{\tau}_\mathrm{i} - (1-\beta_\mathrm{i})\dot{\boldsymbol{x}}^T\boldsymbol{D}_\mathrm{C}\dot{\boldsymbol{x}}\,. \tag{68}$$

TABLE I: Parameters used in the experiments.

Parameter	Unit	Value
$\boldsymbol{K}_\mathrm{max}$	N/m	diag[1000,1000,10,200,200,200]
damping coefficient	-	diag[0.7,0.7,0.7,1,1,1]
C_m	-	0.9
α , ξ , γ	-	1 , 0.08 , 10
$\boldsymbol{K}_p$, $\boldsymbol{K}_i$	-	$0.6\boldsymbol{I}_{6\times6}$, $0.3\boldsymbol{I}_{6\times6}$
δ_c	m	0.04
ρ_min	-	0.001
$x_{\mathrm{t,i}}(0)$	-	7
$\overline{S}_{\mathrm{t,i}}$	J	32
$\underline{S}_{\mathrm{t,i}}$	J	1
$x_{\mathrm{t,f}}(0)$	-	2
$\overline{S}_{\mathrm{t,f}}$	J	2
$\underline{S}_{\mathrm{t,f}}$	J	1

The total storage function S_overall and its time derivative $\dot{S}_\mathrm{overall}$ are

$$S_\mathrm{overall} = S_\mathrm{c} + S_\mathrm{r}\,, \dot{S}_\mathrm{overall} = \dot{S}_\mathrm{c} + \dot{S}_\mathrm{r}\,, \tag{69}$$

$$\dot{S}_\mathrm{overall} = -\dot{\boldsymbol{q}}^T\boldsymbol{\tau} - (1-\beta_\mathrm{i})\dot{\boldsymbol{x}}^T\boldsymbol{D}_\mathrm{C}\dot{\boldsymbol{x}} + \dot{\boldsymbol{q}}^T(\boldsymbol{\tau} + \boldsymbol{\tau}_\mathrm{ext})\,, \tag{70}$$

$$= \dot{\boldsymbol{q}}^T\boldsymbol{\tau}_\mathrm{ext} - (1-\beta_\mathrm{i})\dot{\boldsymbol{q}}^T\boldsymbol{J}^\mathrm{T}\boldsymbol{D}_\mathrm{C}\dot{\boldsymbol{x}}\,. \tag{71}$$

Finally, the modified unified-impedance control ensures stability during contact and no-contact:

$$\boldsymbol{f} = -\boldsymbol{K}_\mathrm{const}\tilde{\boldsymbol{x}} - \boldsymbol{D}_\mathrm{C}\dot{\boldsymbol{x}} - \sigma(S_{\mathrm{t,i}})\rho_\mathrm{align}\boldsymbol{K}_\mathrm{max}\tilde{\boldsymbol{x}} + \tag{72}$$

$$\rho_\mathrm{frc}(\lambda + \sigma(S_{\mathrm{t,f}})(1-\lambda))\boldsymbol{f}_\mathrm{f}\,. \tag{73}$$

Next, the validation scenarios and relevant performance metrics for the exemplary tactile skill to explore the curvatures are discussed.

IV. Experimental Validation

To assess our framework's visuo-tactile exploration and control performance for exploring an unknown rigid curvature, we conduct experiments using a Franka Emika robot to perform a wiping policy. We employ an Intel RealSense D435i camera (Intel Corp., USA) to capture environmental information. The camera is positioned at the robot's flange to minimize body occlusion, and its axis aligns with the z-axis of the end-effector frame during execution. This alignment simplifies the transformation from the task frame to the base frame, enhancing the accuracy of surface normal estimation. The visual pipeline and the robot's master controller run on a mini-ITX PC (HP Z2 Mini G5 Workstation with Intel i7-10700t). The contact surface is 3D-printed with dimensions of $0.26 \times 0.51 \times h$, where $h = 0.02\sin(\frac{\pi}{0.19}y + 0.44) + 0.02$ m.

The experimental procedure evaluates the accuracy of contact alignment monitoring, real-time feedback latency, computational efficiency, and control performance while exploring the unknown 3D rigid curvature for an arbitrarily given wiping policy.

A. Experimental Procedure

During wiping, we conduct experiments for visuo-tactile exploration of the unknown, challenging curved surface. The arbitrary wiping tactile policy used during the exploration is

$$\boldsymbol{f}_\mathrm{d}^\mathrm{ee} = [0, 0, 15, 0, 0, 0]\,,$$
$$\boldsymbol{x}_\mathrm{d}^\mathrm{ee} = [0.04\sin(2t), 0.04(\cos(2t) - 1) - 0.005t, 0, 0, 0, 0]\,,$$

where t is the time parameter. Moreover, Table I shows other parameters designed for the experiments.

First, the robot starts without contacting and aligning to the surface. The expected behavior is that the robot aligns itself, establishes contact, and explores the curvatures while presenting passive and accurate force-motion tracking.

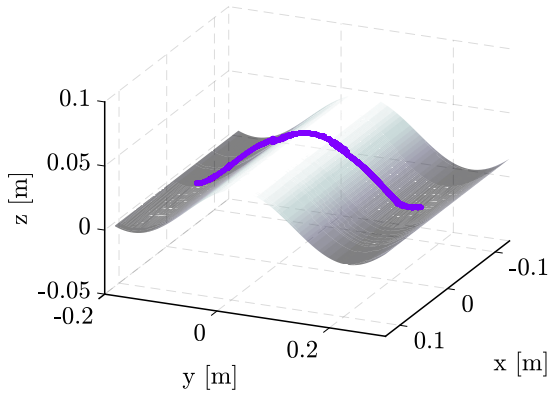


Fig. 4: **Visuo-Tactile Exploration of an Unknown Rigid 3D Curvature.** The robot's actual trajectory along the y- and z-direction in the base frame is compared to the model of the contact surface.

B. Performance Metrics

Performance metrics employed for validation include i.) contact alignment monitoring accuracy, ii.) real-time feedback latency, iii.) computational efficiency, and iv.) control performance. Contact alignment monitoring accuracy measures the precision of exploring the surface being wiped. Real-time feedback latency quantifies the time delay between sensing contact with the surface and adjusting the robot's motion or force. Computational efficiency measures the processing rate achieved by the visual pipeline, considering factors such as frame rate and latency. Control performance encompasses quantifying the accuracy and precision of the robot's movements during the wiping task, measuring deviations from the desired wiping trajectory or force profile to assess control robustness, and evaluating control performance metrics such as root mean square and absolute mean error in tracking. Force control precision considers the uniformity of pressure applied during wiping, ensuring consistent cleaning or polishing without damaging delicate surfaces. It entails measuring the deviation between the desired and actual force exerted on the surface and assessing force control performance using root mean square error and absolute mean error.

V. Results and Discussion

Our experimental results offer quantitative assessments across multiple performance metrics, confirming the effectiveness of our framework in exploring a surface commanded by an arbitrary tactile skill despite unknown physical constraints. We achieve a high accuracy rate in detecting the contact alignment between the robot end-effector and the surface during the exploration using the wiping task. This precision ensures reliable interaction and practical surface exploration, as depicted in Fig. 4 and Fig. 5.

Minimal latency ensures swift and adaptive behavior during the wiping task, enhancing overall efficiency. The vision pipeline, including pre-processing and feature extraction, achieves an average frame rate of 3 frames per second (FPS) with a loop cycle of 300 ms. Even though 300 ms might be considered high performance for vision processed by a standard computer, these values could be improved for better real-time perception and decision-making capabilities, essential for dynamic interaction with the environment. Additionally, the tactile perception runs at 1000 Hz, provided by the robot's internal proprioceptive measurement. Briefly, due to the difference between the loop cycles of the two modalities, vision can be considered a spatial component in the contact monitor alignment. In contrast, the tactile

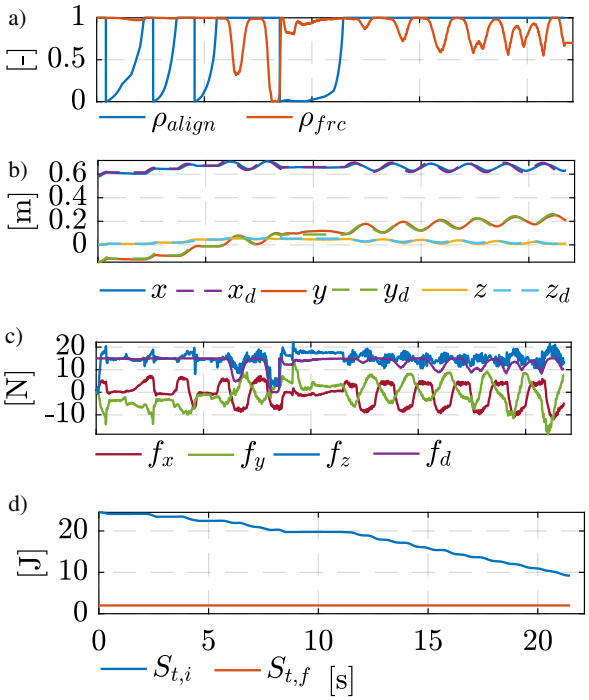


Fig. 5: **Performance Metrics Results for Visuo-Tactile Exploration during Wiping.** a) Controller shaping functions, b) Desired vs. actual motion in the base frame, c) Desired force of 15 N shaped by the function ρ_{frc} vs. measured force in the end effector frame, d) Tank energies.

sensor acts as a temporal modality. Overall, the robot finishes exploring the curvature in 20 s.

Accuracy in monitoring the contact alignment and exploration performance is assessed by comparing the robot's actual trajectory to the model of the contact surface, as depicted in Fig. 4. The robot is commanded only with an arbitrary force-motion policy without modeling the environment. The robot effectively explores the surface while maintaining the desired contact force.

The robot starts without alignment or contact, as illustrated in Fig. 5.a, where ρ_{align} initially decreases and then progressively increases over 0-5 s. Subsequently, it fluctuates between 0 - 1 whenever the contact alignment changes. ρ_{frc} decreases, particularly when the robot moves down the surface due to the margin δ_c. However, the robot adjusts its posture to align with the contact when ρ_{align} is zero. Consequently, the force controller is reactivated with the updated end-effector pose, increasing ρ_{frc} to one such as at approximately around 7 s.

Quantitative analysis of control performance metrics reveals the absolute mean error of 9 mm, 8 mm, and 4 mm in the desired wiping trajectory along the x-, y-, and z-axes, respectively (see Fig. 5.b). The corresponding root mean square errors are approximately 10 mm, 10 mm, and 6 mm. Further quantitative evaluation of the uniformity of pressure applied during visuo-tactile exploration is depicted in Fig. 5.c, illustrating a mean absolute deviation of app. 2 N between the desired force of 15 N shaped by ρ_{frc} and the actual force exerted on the surface about the z-direction in the end-effector frame. Furthermore, the root mean square error in force control performance is app. 3 N. Additionally, the contact surface has high friction, as seen from the forces about the other directions, which fluctuates up to the magnitude of 10 N. Furthermore, Fig. 5.d illustrates that the energy tanks remain within their designated limits.

Specifically, $S_{\mathrm{t,i}}$ remains below 32 J and decreases due to large movements on a surface with friction. Additionally, $S_{\mathrm{t,f}}$ remains constant at 2 J without overloading the tank. This indicates the robot's and its surroundings' safety during contact alignment and visuo-tactile exploration.

The authors would like to mention that further study should focus on deciding C_m instead of fine-tuning the current surface material properties, such as friction and rigidity. Overall, the quantitative evaluations confirm the effectiveness and practicality of our framework for automated visuo-tactile exploration of unknown rigid 3D curvatures handled at the robot's low-level control. The accuracy, precision, latency, and computational efficiency presented demonstrate our method's potential for increasing robotic deployment in manufacturing automation and other industries requiring intricate robotic interaction processes.

VI. Conclusion

In conclusion, this paper aims to bridge the gap between current robotic capabilities and the demands of real-world applications by simple yet effective and intuitive robotic skill programming for arbitrarily given tactile policy without requiring specialized expertise that integrates visuo-tactile exploration of unknown rigid 3D curvatures by vision-augmented unified force-impedance control. Combining tactile and vision data, we formulate a robust online contact alignment monitoring system that considers tactile error, local surface curvature, and surface orientation. This information is seamlessly integrated into a vision-augmented unified force-impedance control framework, allowing for adjusting robot stiffness and regulation of force while exploring the curvatures. Virtual energy tanks ensure system passivity and stability throughout the visuo-tactile exploration of the unknown rigid 3D curvatures. Experimental validation with a Franka Emika research robot executing wiping tasks on challenging surfaces confirms the efficacy of our approach in achieving precise and passive visuo-tactile exploration. Comprehensive performance metrics are used for validation, including contact alignment monitoring accuracy, real-time feedback latency, computational efficiency, and control performance. These metrics provide quantitative insights into our proposed method's precision, uniformity, speed, and effectiveness, ensuring its practical applicability in real-world manufacturing scenarios. As a limitation, highly irregular curvatures are excluded from the study scope due to potential challenges for both sensors in accurately measuring surface properties. Additionally, in the current implementation, the camera can observe only the current region of interest without predicting forthcoming curvatures, a research opportunity to address in the future. Future work will investigate planning the force-motion policy for the explored curvatures toward a complete solution for generating the object-centric tactile policy for arbitrarily given policies, helping increase the robot deployment in real-world manufacturing tasks.

References

[1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[2] A. Lambert and S. Gupta, "Disassembly modeling for assembly, maintenance, reuse and recycling," 12 2004.

[3] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American Control Conference*, 1984, pp. 304–313.

[4] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *The International Journal of Robotics Research*, vol. 14, no. 1, pp. 19–36, 1995.

[5] 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.

[6] J. D. Schutter, T. D. Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[7] A. Kramberger, A. Gams, B. Nemec, 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.

[8] M. Iskandar, C. Ott, A. Albu-Schäffer, B. Siciliano, and A. Dietrich, "Hybrid force-impedance control for fast end-effector motions," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3931–3938, 2023.

[9] 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.

[10] E. C. Balta, K. Jain, Y. Lin, D. Tilbury, K. Barton, and Z. M. Mao, "Production as a service: A centralized framework for small batch manufacturing," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, 2017, pp. 382–389.

[11] T. Migimatsu and J. Bohg, "Object-centric task and motion planning in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 844–851, 2020.

[12] M. Vochten, A. M. Mohammadi, A. Verduyn, T. De Laet, E. Aertbeliën, and J. De Schutter, "Invariant descriptors of motion and force trajectories for interpreting object manipulation tasks in contact," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4892–4912, Dec 2023.

[13] C. Chi, X. Sun, N. Xue, T. Li, and C. Liu, "Recent progress in technologies for tactile sensors," *Sensors*, vol. 18, no. 4, 2018.

[14] S. Ganguly and O. Khatib, "Experimental studies of contact space model for multi-surface collisions in articulated rigid-body systems," in *Proceedings of the 2018 International Symposium on Experimental Robotics*, J. Xiao, T. Kröger, and O. Khatib, Eds. Cham: Springer International Publishing, 2020, pp. 425–436.

[15] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More than a feeling: Learning to grasp and regrasp using vision and touch," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, p. 3300–3307, Oct. 2018.

[16] A. Sachtler, K. Nottensteiner, M. Kaßecker, and A. Albu-Schäffer, "Combined visual and touch-based sensing for the autonomous registration of objects with circular features," in *2019 19th International Conference on Advanced Robotics (ICAR)*, 2019, pp. 426–433.

[17] N. Fazeli, M. Oller, J. Wu, Z. Wu, J. B. Tenenbaum, and A. Rodriguez, "See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion," *Science Robotics*, vol. 4, no. 26, p. eaav3123, 2019.

[18] K. Nottensteiner, A. Sachtler, and A. Albu-Schäffer, "Towards Autonomous Robotic Assembly: Using Combined Visual and Tactile Sensing for Adaptive Task Execution," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 3, p. 49, Feb. 2021.

[19] S. Suresh, H. Qi, T. Wu, T. Fan, L. Pineda, M. Lambeta, J. Malik, M. Kalakrishnan, R. Calandra, M. Kaess, J. Ortiz, and M. Mukadam, "Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation," 2023.

[20] S. Suresh, M. Bauza, K.-T. Yu, J. G. Mangelson, A. Rodriguez, and M. Kaess, "Tactile slam: Real-time inference of shape and pose from planar pushing," 2021.

[21] Y. Kato, P. Balatti, J. M. Gandarias, M. Leonori, T. Tsuji, and A. Ajoudani, "A self-tuning impedance-based interaction planner for robotic haptic exploration," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9461–9468, Oct 2022.

[22] M. Qin, J. Brawer, and B. Scassellati, "Robot tool use: A survey," *Frontiers in Robotics and AI*, vol. 9, 2023.

[23] K. Karacan, D. Grover, H. Sadeghian, F. Wu, and S. Haddadin, "Tactile exploration using unified force-impedance control," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5015–5020, 2023, 22nd IFAC World Congress.

[24] 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.

[25] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *2011 IEEE ICRA*. IEEE, 2011, pp. 1–4.

[26] T. Westfechtel, K. Ohno, B. Mertsching, R. Hamada, D. Nickchen, S. Kojima, and S. Tadokoro, "Robust stairway-detection and localization method for mobile robots using a graph-based model and competing initializations," *The International Journal of Robotics Research*, vol. 37, no. 12, pp. 1463–1483, 2018.

[27] K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, 2016.

[28] S. Stramigioli, "Energy-aware robotics," in *Mathematical Control Theory I*, M. K. Camlibel, A. A. Julius, R. Pasumarthy, and J. M. Scherpen, Eds. Cham: Springer International Publishing, 2015, pp. 37–50.

[29] Y. Michel, C. Ott, and D. Lee, "Passivity-based variable impedance control for redundant manipulators," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9865–9872, 2020, 21st IFAC World Congress.

A.3 "Tactile Robot Programming: Transferring Task Constraints into Constraint-Based Unified Force-Impedance Control"

This is the pre-print version (author accepted manuscript) of the following publication: 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)*. 2024.

Tactile Robot Programming: Transferring Task Constraints into Constraint-Based Unified Force-Impedance Control

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

***Abstract*— Flexible manufacturing lines are required to meet the demand for customized and small batch-size products. Even though state-of-the-art tactile robots may provide the versatility for increased adaptability and flexibility, their potential is yet to be fully exploited. To support robotics deployment in manufacturing, we propose a task-based tactile robot programming paradigm that uses an object-centric tactile skill definition that directly links identified object constraints of the task to the definition of constraint-based unified force-impedance control. In this study, we first explain the basic concept of abstracting the task constraints experienced by the object and transferring them to the robot's operational space frame. Second, using the object-centric tactile skill definition, we synthesize unified force-impedance control and formalized holonomic constraints to enable flexible task execution. Later, we propose the quantified analysis metrics for the process by analyzing them as a typical example of flexible manipulation disassembly skills, e.g., levering and unscrew-driving regarding their object requirements. Supported by realistic experimental evaluation using a Franka Emika robot, our tactile robot programming approach for the direct translation between task-level constraints and robot control parameter design is shown to be a viable solution for increased robotic deployment in flexible manufacturing lines.**

I. Introduction

Today's primary demand for flexible manufacturing lines is customization and, thus, small batch-size production [1]. This necessitates robots that are adaptable to changing task constraints. State-of-the-art tactile robots provide the versatility for increased adaptability and flexibility [2]. Nevertheless, their deployment for tactile and flexible interaction requires control experts. Consequently, their potentials are yet to be fully exploited [3].

One solution for increased robot deployment is to enable simple yet effective and intuitive robotic skill definitions that do not require control expertise for application. To elaborate, human experts in manufacturing sectors have comprehensive knowledge about the desired task and its requirements [4]. For instance, the task constraints, such as force and motion expected to be experienced by objects during manipulation, are well-defined. Take, for example, processing applications

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 the Centre for Tactile Internet with Human-in-the-Loop (CeTI). 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 and the research and development program under grant no. DIK0249), the Lighthouse Initiative Geriatronics by LongLeif GaPa gGmbH (Project Y), 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, and SafeRoBAY (grant number: DIK0203/01).

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

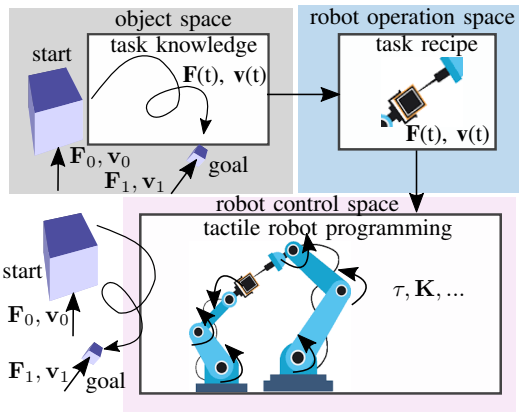


Fig. 1: **Direct transfer of task information and constraints into tactile robot control for increased flexibility.** A task-oriented tactile robot programming framework translates the desired object-centric force/motion task into the robot domain.

such as milling, where the contact forces and feed speed are calculated in a standardized manner. Consequently, robots for flexible production need to be programmed to manipulate the objects, respecting those well-defined forces and motion. In the robotics community, numerous strategies for force-motion interaction have been developed, such as admittance control [5], impedance control [6], force control [7], and unified control [8], [9]. Nevertheless, robots operating under highly varying conditions, such as small batch-size sectors, lead to changing task constraints, requiring re-configuring and tuning the robot controllers accordingly.

In order to realize more natural and intuitive robot programming, it would be desired to understand the task constraints directly and feed these to the controller, see Fig. 1. Representations, e.g., the operational space framework [10], constrained-based or object-centric task specifications [11], [12], are significant steps towards this easier-to-use programming paradigm. However, directly embedding the constraints an object experiences during task execution for robot control also requires using the task constraints and directly combining them with modern controllers for flexible tactile task execution. As such, controllers can be tuned by non-experts and learned by demonstration based on analyzing the desired task constraints.

Numerous studies consider force control as a method for developing adaptive robotic skills, testing the proposed controllers with constant force values, thresholds, or constraints [13]–[17]. Nonetheless, such strategies struggle with robustness when faced with environmental uncertainties and

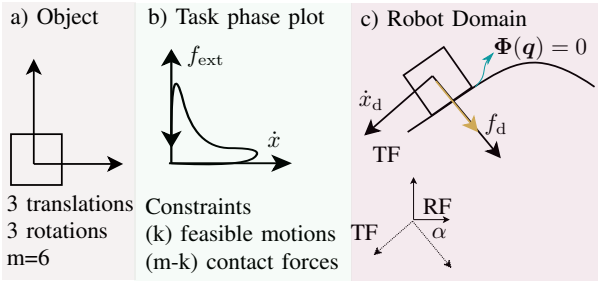


Fig. 2: **Tactile skill definition imposed by the physical task constraints.** The object's desired state dictates the task phase plot, whereas the present environment's circumstances shape it. TF: Task Frame and RF: Robot Frame.

may fail when perceptual imprecision occurs [18], [19]. Impedance control is an established method for enforcing dynamic behavior to achieve the desired motion while interacting with the environment [6]. Adaptive tuning of impedance parameters is advantageous in various applications [20], [21]. Variable impedance approaches are also used in shared autonomy applications to coordinate the motions of humans and robots and to update the predefined skill motion policy [22]. Although multiple efforts were taken among the robotic community to realize adaptive manipulation with perception uncertainties [11], [23]–[28], this is yet to be solved in principle and has not found its way into the real-world industry.

This paper proposes a task-oriented tactile robot programming paradigm that uses an object-centric tactile skill definition. This concept links identified object constraints of the task directly to the definition of constrained-based unified force-impedance control, enabling the translation between task-level constraints and robot control parameter design. For this, we

1. Introduce the basic concept of abstracting the task constraints experienced by the object and transferring them to the robot operational space frame,
2. Extend the controller schemes of our previous works [22], [29] by the formalized holonomic constraints to enable flexible task execution,
3. Analyze as a challenging, however, representative example of flexible manipulation disassembly skills starting from the respective object requirements,
4. Propose suitable process analysis metrics, and
5. Experimentally validate the approach with a Franka Emika robot.

The remainder of this paper is structured as follows. First, Sec. II formulates the processes using their required task constraints under the tactile skills and introduces the tactile robot programming method, transferring the task constraints into the robot control. The validation scenarios, the proposed task performance metrics, and the corresponding results are demonstrated and discussed in Sec. III and Sec. IV. Finally, Sec. V concludes the paper.

II. Methodology

We refer to interaction skills requiring force and motion profiles and compliant behavior as *tactile manipulation skills* [29]. Successful execution of the tactile skills is a challenging problem that involves force and form closures between the robot end-effector and the manipulation object. Every object in the real world is subject to force and motion constraints in three translational and three rotational directions, as depicted in Fig. 2a). This object-centric abstraction defines any tactile skill independent of the execution instance, like the robot. We introduced the phase plot [29] to represent the force-velocity task constraints for required skills. This representation now serves as the basis to understand the dynamic between the skill constraints based on the simple object-centric force-velocity analysis as depicted in Fig. 2b). Roughly saying, the task phase plot is the recipe for the task execution by any instance. The great challenge is formalizing this recipe so that it fits into the robot control and can handle changes in execution, which we describe in detail in the following.

A. Tactile Skill Representation

As previously mentioned, any tactile process, such as levering or unscrew-driving, is 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 k-dimension. In other words, during task execution, ideally, the tool moves along the free axes at velocity $\dot{\boldsymbol{x}}^{\text{t}}_{\text{k}\times 1}$, while the contact forces $\boldsymbol{f}^{\text{t}}_{(6-\text{k})\times 1}$ occur along the other axes. Selection matrices $\boldsymbol{T}_{6\times\text{k}}$ and $\boldsymbol{Y}_{6\times(6-\text{k})}$ are comprised of 1 and 0 to decouple the motion and force sub-spaces [7], [30]. Ideally, the task phase plot (Fig. 2b) demonstrates the entire power cycle that the object goes through, in which the force-velocity relation evolves such that the contact is established smoothly $\dot{\boldsymbol{x}} = 0$ with the surface, at the same time an external force $\boldsymbol{f}_{\text{ext}} > 0$ is exerted to it.

For the exemplary scenario in Fig. 2c) the selection matrices $\boldsymbol{Y}$ and $\boldsymbol{T}$ are deduced from the physical task constraints and computed as follows. Assuming $\text{k} = 5$, $\boldsymbol{Y} = [\delta_i(d)]_{6\times 1}$, where a Kronecker function $\delta_i(d)$ is defined as

$$\delta_i(d) = \begin{cases} 0, & \text{if } i \neq d \\ 1, & \text{if } i = d \end{cases} \tag{1}$$

By adding zero columns to $\boldsymbol{Y}$ up to the dimension of six by six to span matrix $\boldsymbol{Y}'$, we get

$$\boldsymbol{Y}' = [\delta_{ij}(d)]_{6\times 6}, \tag{2}$$

where

$$\delta_{ij}(d) = \begin{cases} 1, & \text{if } i = j = d \\ 0, & \text{else} \end{cases} \tag{3}$$

Let $\boldsymbol{T}' = \boldsymbol{I} - \boldsymbol{Y}'$ and discarding the zero columns of $\boldsymbol{T}'$ leads to $\boldsymbol{T}$. In case $d = 3$ which holds for the examples to be discussed in Sec. III, we have:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{Y} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4}$$

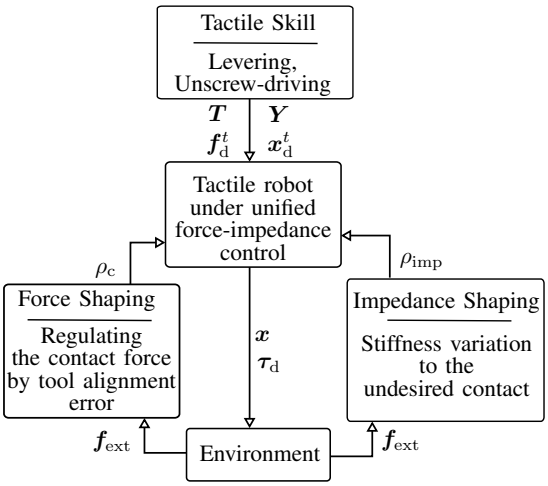


Fig. 3: **Task-oriented Tactile Robot Programming.** The controller shaping functions ensure that the controller interacts with the environment robustly and safely.

Rearranging the kinematics equation for the object using natural (geometric) and artificial constraints in the robot frame yields:

$$\dot{\boldsymbol{x}}_{6\times1} = \boldsymbol{R}^{0}_{\mathrm{ee}}(\alpha)\boldsymbol{T}_{6\times k}\dot{\boldsymbol{x}}^{\mathrm{t}}_{k\times1}\,, \dot{\boldsymbol{x}}^{\mathrm{t}}_{k\times1} = \boldsymbol{T}^{\#}\boldsymbol{R}^{0}_{\mathrm{ee}}(\alpha)^{\mathrm{T}}\boldsymbol{J}\dot{\boldsymbol{q}}\,, \quad (5)$$

$$\boldsymbol{f}_{6\times1} = \boldsymbol{R}^{0}_{\mathrm{ee}}(\alpha)\boldsymbol{Y}_{6\times(6-k)}\boldsymbol{f}^{\mathrm{t}}_{(6-k)\times1}\,, \quad (6)$$

$$\boldsymbol{J}_{\mathrm{con}} = \boldsymbol{Y}^{\#}\boldsymbol{R}^{0}_{\mathrm{ee}}(\alpha)^{\mathrm{T}}\boldsymbol{J}\,, \boldsymbol{J}_{\mathrm{free}} = \boldsymbol{T}^{\#}\boldsymbol{R}^{0}_{\mathrm{ee}}(\alpha)^{\mathrm{T}}\boldsymbol{J}\,, \quad (7)$$

where $\boldsymbol{R}^{0}_{\mathrm{ee}}(\alpha)$ is the rotation matrix of the end-effector, $\boldsymbol{J}$ is the robot Jacobian matrix. The Moore–Penrose pseudo-inverse of the matrices $\boldsymbol{Y}$ and $\boldsymbol{T}$ are:

$$\boldsymbol{T}^{\#} = (\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}})^{-1}\boldsymbol{T}\,, \qquad \boldsymbol{Y}^{\#} = (\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}})^{-1}\boldsymbol{Y}\,. \quad (8)$$

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 develop the unified force-impedance control paradigm to command the object motion and force imposed by the task constraints. We also set the control shaping functions to maintain the continuity in the task phase plot by stiffness variation and force adaptation, as framed in Fig. 3.

B. Controller Design

The proposed control law for adaptive tactile skills is synthesized unified force-impedance control [8], [22] and constrained control [30], [31]. The controller has four main features:

I) following the desired motion $\boldsymbol{x}_d$ with impedance control
II) regulating the model-based contact force $\boldsymbol{\lambda}$ based on the desired force $\boldsymbol{f}_d$ without having to tune additional parameters, e.g., PID gains,
III) gravity compensation,
IV) null-space control.

The corresponding control torque $\boldsymbol{\tau}_{\mathrm{d}} \in \mathbb{R}^n$ is defined as

$$\boldsymbol{\tau}_{\mathrm{d}} = \boldsymbol{\tau}_{\mathrm{imp}} + \boldsymbol{\tau}_{\mathrm{frc}} + \boldsymbol{\tau}_{\mathrm{g}} + \boldsymbol{\tau}_{\mathrm{null}}\,, \quad (9)$$

where $\boldsymbol{\tau}_{\mathrm{imp}}$, $\boldsymbol{\tau}_{\mathrm{frc}}$, $\boldsymbol{\tau}_{g}$, and $\boldsymbol{\tau}_{\mathrm{null}} \in \mathbb{R}^n$ are the input torque for (i) impedance control; (ii) force control; (iii) gravity compensation; and (iv) null-space control.

*1) **Constrained Robot Dynamics**:* The partially constrained robot dynamics can be deduced by an augmented Lagrangian, where the Lagrangian multiplier $\boldsymbol{\lambda}$ are the generalized contact forces when attempting to break the constraints. Using the Euler-Lagrange equations in the extended space of generalized coordinates $\boldsymbol{q} \in \mathbb{R}^n$, multiplier $\boldsymbol{\lambda} \in \mathbb{R}^{(6-k)}$, and collocated external force along the free directions $\boldsymbol{f}_{\mathrm{free}} \in \mathbb{R}^k$ subjected to the holonomic constraints $\boldsymbol{\Phi}(\boldsymbol{q}) = \boldsymbol{0} \in \mathbb{R}^{(6-k))}$ where feasible motions are allowed in k dimensions yields

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_{\mathrm{d}} + \boldsymbol{\tau}_{\mathrm{ext}}\,, \quad (10)$$

$$\boldsymbol{\tau}_{\mathrm{ext}} = \boldsymbol{J}^{\mathrm{T}}_{\mathrm{con}}(\boldsymbol{q})\boldsymbol{\lambda} + \boldsymbol{J}^{\mathrm{T}}_{\mathrm{free}}\boldsymbol{f}_{\mathrm{free}}\,, \quad (11)$$

where $\boldsymbol{\tau}_{\mathrm{ext}} \in \mathbb{R}^n$ represents the external torque exerted on the robot, while $\boldsymbol{M}(\boldsymbol{q})$ denotes the robot mass matrix, $\boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{\mathrm{n}}$ signifies the Coriolis and centrifugal vector, and $\boldsymbol{g}$ stands for the gravity vector in joint space. Additionally, $\boldsymbol{\tau}_{\mathrm{d}} \in \mathbb{R}^{\mathrm{n}}$ represents the control torque applied by the robot. Finally, we define the Jacobian of the constraints $\boldsymbol{J}_{\mathrm{con}} = \frac{\partial \boldsymbol{\Phi}(\boldsymbol{q})}{\partial(\boldsymbol{q})} \in \mathbb{R}^{(6-k)\times n}$ computed in (7):

$$\dot{\boldsymbol{\Phi}}(\boldsymbol{q}) = \boldsymbol{0}_{(6-\mathrm{k})\times1} = \boldsymbol{J}_{\mathrm{con}}\dot{\boldsymbol{q}}\,. \quad (12)$$

*2) **Impedance Control**:* The desired impedance behavior along the free axes at the tooltip is

$$\boldsymbol{f}_{\mathrm{imp}} = \boldsymbol{K}_C\tilde{\boldsymbol{x}} + \boldsymbol{D}_C\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{M}_C(\boldsymbol{q})\ddot{\boldsymbol{x}}_{\mathrm{d}} + \boldsymbol{C}_C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{x}}_{\mathrm{d}}\,, \quad (13)$$

$$\boldsymbol{\tau}_{\mathrm{imp}} = \boldsymbol{J}^{\mathrm{T}}_{\mathrm{free}}\boldsymbol{f}_{\mathrm{imp}}\,, \quad (14)$$

where $\boldsymbol{x} \in \mathbb{R}^k$ and $\boldsymbol{x}_{\mathrm{d}} \in \mathbb{R}^k$ are the actual pose and the desired pose along the free axes, respectively, as well as, the pose error is $\tilde{\boldsymbol{x}} = \boldsymbol{x}_{\mathrm{d}} - \boldsymbol{x}$. Furthermore, $\boldsymbol{K}_C$ and $\boldsymbol{D}_C \in \mathbb{R}^{k\times k}$ are diagonal stiffness and damping matrices, respectively. $\boldsymbol{M}_C(\boldsymbol{q})$ is the robot mass matrix in task space along the free axes, $\boldsymbol{C}_C(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{k\times k}$ is the Coriolis and centrifugal matrix.

The undesired contacts cause deviations from the desired pose that create either a pose error $\tilde{\boldsymbol{x}}^{\mathrm{ee}} \in \mathbb{R}^6$, or external forces $\boldsymbol{f}^{\mathrm{ee}}_{\mathrm{ext}} \in \mathbb{R}^6$ at the end-effector. This phenomenon is exploited to react robustly to the undesired contacts and to re-configure the end-effector [22] by adapting the stiffness matrix $\boldsymbol{K}_{\mathrm{C}}$. Having the S_{t} threshold is critical for compensating for minor environmental effects such as friction on the surface and measurement inaccuracies. It is also worth noting that using position instead of velocity or acceleration results in a less noisy signal. The normalized metric β is then coupled to $\boldsymbol{K}_{\mathrm{C}}$ via ρ_{imp}.

$$\beta = 1 - \frac{\|\boldsymbol{f}^{\mathrm{ee}}_{\mathrm{ext}} \cdot \tilde{\boldsymbol{x}}^{\mathrm{ee}}\|}{S_{\mathrm{t}}}\,, \quad (15)$$

$$\boldsymbol{K}_C = \rho_{\mathrm{imp}}\boldsymbol{K}_{\max}\,, \quad (16)$$

where the stiffness adaptation parameter ρ_{imp} is calculated by

$$\dot{\rho}_{\mathrm{imp}} = \begin{cases} \min\{\rho_{\mathrm{r}}, 0\}\,, & \rho_{\mathrm{imp}} = 1 \\ \rho_{\mathrm{r}}\,, & 0 < \rho_{\mathrm{imp}} < 1,\ \rho_{\mathrm{imp}}(0) = 0, \\ \max\{\rho_{\mathrm{r}}, 0\}\,, & \rho_{\mathrm{imp}} = 0 \end{cases} \quad (17)$$

and ρ_{r} is

$$\rho_{\mathrm{r}} = \beta\rho_{\mathrm{imp}} + \rho_{\min}. \quad (18)$$

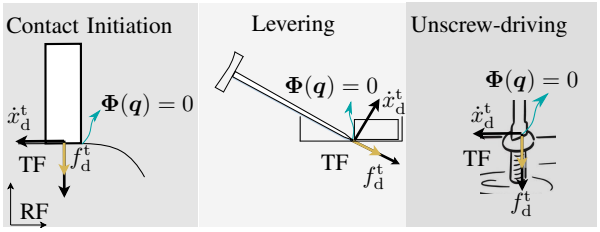


Fig. 4: **Tactile disassembly skill examples.** Task constraints $\boldsymbol{T}$ and $\boldsymbol{Y}$ encode the motion and force sub-spaces and are the same in the shown cases.

Once the robot's behavior is compliant $\rho_{\mathrm{imp}} = 0$, it reacts to the environment. Adapted to the current environmental conditions, the robot recovers its maximum stiffness and resumes the desired motion from its present configuration. It should be noted that a slight positive constant $\rho_{\min}$ is included in the shaping function dynamics to provide an initial increment for the situation $\rho_{\mathrm{imp}} = 0$.

*3) **Force Control**:* Instead of having to tune gains and parameters to specific situations, we chose to design the force controller $\boldsymbol{f}_{\mathrm{frc}}$ to be the difference between the desired $\boldsymbol{f}_{\mathrm{d}} \in \mathbb{R}^{(6-k)}$ and the model-based contact force $\boldsymbol{\lambda}$, considering $\boldsymbol{\lambda}$ should be equal to $\boldsymbol{f}_{\mathrm{d}}$ [30]

$$\boldsymbol{f}_{\mathrm{frc}} = \boldsymbol{f}_{\mathrm{d}} - \boldsymbol{\lambda}\,. \tag{19}$$

To calculate the model-based contact force $\boldsymbol{\lambda}$, the kinematics equation at acceleration level in (20) is sol it.

$$\ddot{\boldsymbol{\Phi}}(\boldsymbol{q}) = \mathbf{0}_{(6-\mathrm{k})\times 1} = \dot{\boldsymbol{J}}_{\mathrm{con}}\dot{\boldsymbol{q}} + \boldsymbol{J}_{\mathrm{con}}\ddot{\boldsymbol{q}} \tag{20}$$

After inserting the joint accelerations from (10) and the input torque (9), rearranging the terms yields

$$\begin{aligned}\boldsymbol{\lambda} = &-\boldsymbol{J}_{\mathrm{con}}^{\#}\boldsymbol{J}_{\mathrm{con}}\boldsymbol{M}^{-1}(\boldsymbol{J}_{\mathrm{free}}^{\mathrm{T}}\boldsymbol{f}_{\mathrm{imp}} + \boldsymbol{\tau}_{\mathrm{null}}) + \\ &\boldsymbol{J}_{\mathrm{con}}^{\#}\boldsymbol{J}_{\mathrm{con}}\boldsymbol{M}^{-1}\boldsymbol{c} - \boldsymbol{J}_{\mathrm{con}}^{\#}\dot{\boldsymbol{J}}_{\mathrm{con}}\dot{\boldsymbol{q}} + \\ &\boldsymbol{J}_{\mathrm{con}}^{\#}\boldsymbol{J}_{\mathrm{con}}\boldsymbol{M}^{-1}(\boldsymbol{J}_{\mathrm{free}}^{\mathrm{T}}\boldsymbol{f}_{\mathrm{free}})\,.\end{aligned} \tag{21}$$

The inertia-weighted pseudo-inverse of the constraint Jacobian $\boldsymbol{J}_{\mathrm{con}}$ is $\boldsymbol{J}_{\mathrm{con}}^{\#} = (\boldsymbol{J}_{\mathrm{con}}\boldsymbol{M}^{-1}\boldsymbol{J}_{\mathrm{con}}^{\mathrm{T}})^{-1}$. Finally, the input torque to control the desired contact force is

$$\boldsymbol{\tau}_{\mathrm{frc}} = \rho_{\mathrm{frc}}\boldsymbol{J}_{\mathrm{con}}^{\mathrm{T}}\boldsymbol{f}_{\mathrm{frc}}\,. \tag{22}$$

Additionally, we design the force shaping function ρ_{frc}. The force shaping function combines ρ_{c} and ρ_{imp} to adapt the commanded force caused by the tool alignment error and undesired contacts.

$$\rho_{\mathrm{frc}} = \rho_{\mathrm{imp}}\rho_{\mathrm{c}}\,. \tag{23}$$

The robot tolerates the tool alignment error $\|\boldsymbol{f}_{\mathrm{d}}^{\mathrm{ee}} \cdot \tilde{\boldsymbol{x}}^{\mathrm{ee}}\|$ during the contact by the lower limit of $S_{\min}$ within $\delta_{\mathrm{c}} > 0$. Moreover, if the robot loses surface contact due to the large tool alignment error, the robot is only impedance-controlled and follows the desired motion.

$$\rho_{\mathrm{c}} = \begin{cases} 1\,, & \|\boldsymbol{f}_{\mathrm{d}}^{\mathrm{ee}} \cdot \tilde{\boldsymbol{x}}^{\mathrm{ee}}\| \le S_{\min} \\ 0.5(1+\cos((\pi\frac{\|\boldsymbol{f}_{\mathrm{d}}^{\mathrm{ee}}\cdot\tilde{\boldsymbol{x}}^{\mathrm{ee}}\|-S_{\min}}{\delta_{\mathrm{c}}}))\,, & S_{\min} < \|\boldsymbol{f}_{\mathrm{d}}^{\mathrm{ee}} \cdot \tilde{\boldsymbol{x}}^{\mathrm{ee}}\| \\ & \le S_{\min} + \delta_{\mathrm{c}} \\ 0\,, & \text{otherwise.} \end{cases} \tag{24}$$

It is noteworthy to mention that even though the robot behaves compliantly to the undesired contacts with the help of the control shaping functions as well as we fully decouple the motion and force sub-spaces, using $\boldsymbol{T}$ and $\boldsymbol{Y}$ based on the task constraints, the unification of force and impedance control, as well as, variable stiffness in the impedance controller may compromise the stability [32]. However, one may guarantee stability by installing virtual energy tanks [33].

Next, the validation scenarios and relevant evaluation metrics for the exemplary tactile skills are discussed.

III. Validation Scenarios

We focus on the tasks from the disassembly processes as our representative examples. Levering and unscrew-driving are two crucial skills widely used in disassembly tasks involved in electronics waste recycling, a field heavily dependent on manual labor and challenging to automate by using robots [34].

A. 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. 4. One approach to levering is to use periodic motions $\boldsymbol{x}_d^t$ while maintaining contact $\boldsymbol{f}_d^t$ perpendicular to the tooltip, essentially when the desired force is complicated to define to lever an object [34]. Levering is likely successful when the locking mechanism is broken or fully opened, thereby stuck. In other words, it is difficult to define a goal for a successful execution.

We design an experimental setup to enable reproducible comparisons by choosing a car outlet socket as our exemplary object and manufacturing an aluminum counterpart to fix it firmly. The lid of a car socket outlet is levered by using the peg. The length and diameter of the peg are $20\,\mathrm{mm}$ and $3\,\mathrm{mm}$, respectively. 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 behavior is that if no contact is sensed, the robot should stop force control and restart when contact is sensed. The motion is a function of time t, whereas amplitude and frequency are $B = 0.04\,\mathrm{m}$ and $\omega = 0.15\,\mathrm{Hz}$, respectively.

$$\boldsymbol{x}_d^t = [0\,|B\sin(2\pi\omega t)|\,0\,0\,0]^T\,,\ \boldsymbol{f}_d^t = [12]\,, \tag{25}$$

$$S_t = 0.5\,, S_{\min} = 0.0001\,, \delta_c = 0.7\,. \tag{26}$$

The task constraints $\boldsymbol{T}$ and $\boldsymbol{Y}$ are the same as derived in (4).

B. Unscrew-driving

Electronic unscrew-driving is possible in two ways: button-triggered or push-to-start. As button-triggered screwdriver requires additional setups [35], we analyze the push-to-start electronic screwdriver-based process. The process requires the screwdriver to be pushed while the screw moves in the opposite direction. Push-to-start is generally triggered by a certain amount of force, as provided in the tool's datasheet. The tool should also be perpendicular to the screw to maintain contact. During our experiments, we use an M8x25mm screw and drill the thread through an aluminum

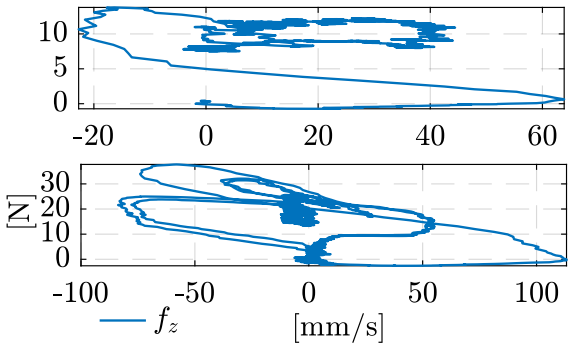


Fig. 5: **Task phase plots.** Task evolution is presented for Top: levering and Bottom: unscrew-driving. Continuity in the plots shows the robustness to varying external forces, thereby, the successful execution of the process.

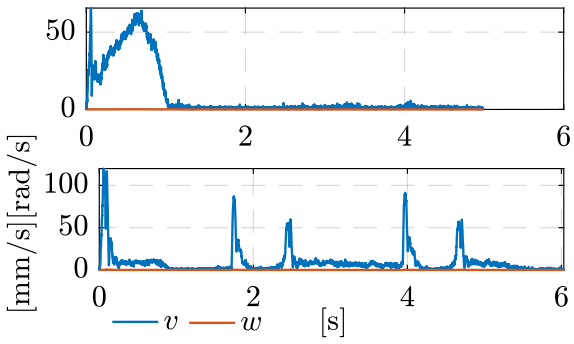


Fig. 6: **Norm of the linear and rotational velocities of the end effector.** Top: Levering and Bottom: unscrew-driving. Constant orientation due to rotational velocity around zero while moving means that the robot maintains contact.

counterpart to fix it in our experimental setup. While the same task constraints $\boldsymbol{T}$ and $\boldsymbol{Y}$ are used, the rest of the parameters are as follows:

$$\boldsymbol{x}_d^t = [0\,0\,0\,0\,0]^T\,,\, \boldsymbol{f}_d^t = [20]\,, \tag{27}$$

$$S_t = 3.5\,,\, S_{\min} = 0.0001\,,\, \delta_c = 1.8\,. \tag{28}$$

The translational $\boldsymbol{K}_{\max,\mathrm{t}}$ and rotational $\boldsymbol{K}_{\max,\mathrm{r}}$ stiffness used in the experiments are $1500\,\mathrm{N/m}$ and $200\,\mathrm{Nm/rad}$, respectively, whereas the damping ratio is $\mathrm{diag}[0.7, 0.7, 0.7, 1, 1, 1]$.

C. Remarks

Based on the levering and unscrew-driving process definitions, for successfully executing the process, the robot should be i) positioning the tool as accurately as possible at a desired region of interest, ii) applying a force profile as accurately as possible, iii) ensuring an accurate motion and process success even if undesired external forces occur, iv) deviating from a defined motion profile as little as possible. Thus, our task-oriented tactile robot programming framework is evaluated for these four items under the categories of a) position accuracy, b) displacement tolerance, c) force tolerance, and d) force and motion error. A Franka Emika robot is used for the experiments, and the robot's internal sensing capabilities are used to measure the position, velocity, and external force/torque at the end-effector [36].

IV. Results and Discussion

For the levering scenario, the expected behavior for the robot is to move to the contact and maintain the contact force of $12\,\mathrm{N}$ with the lid in the z-direction while moving along the x-direction in the end-effector frame. In contrast, the lid is levered about the y-direction. As the robot is compliant in the z-direction, it moves with the lid along the z-direction due to force control while moving along the x-direction due to impedance control. In this case, the motion in the z-direction is treated as a tool alignment error and regulated by the force control shaping function ρ_c. Thus, the robot can only move within the threshold δ_c, meaning that the force is reduced to zero after a certain distance by ρ_{frc}. During unscrew-driving, the robot pushes the screw with the force of $20\,\mathrm{N}$ in the z-direction while moving in the opposite direction. As the screw gets loose during the process, it starts moving in other directions, altering the external force at the end-effector. Therefore, the stiffness adaptation ρ_{imp} is activated, and due to this compliant behavior, the robot's end-effector is pushed by the external force and reconfigured itself. This process continues until the screw is fully unscrewed to $25\,\mathrm{mm}$.

The task phase plot is developed with the external force and velocity in the z-direction in the robot's task space. The force and velocity evolution also translates to the power development during the task, as shown in Fig. 5. Therefore, the continuity in the plot means the task proceeds successfully. In particular, the levering process starts with no-contact $0\,\mathrm{N}$. The initial back and forth motion around $60\,\mathrm{mm/s}$ and $-20\,\mathrm{mm/s}$ with around $15\,\mathrm{N}$ is the initial contact. After the initial contact with the lid, the force decreases to $12\,\mathrm{N}$ while it moves together with the lid with $40\,\mathrm{mm/s}$. Later, the contact force is maintained around $7\,\mathrm{N}$ at $0\,\mathrm{mm/s}$, where the lid is fully opened and cannot move anymore. During the unscrew-driving process, after the initial contact of $30\,\mathrm{N}$, the robot starts unscrew-driving by the force of $20\,\mathrm{N}$. As it can be seen in the plot in Fig. 5, while the robot applies the force of $20\,\mathrm{N}$, it also moves up to the velocity of $-50\,\mathrm{mm/s}$, as the screw keeps unscrewed. However, later, the robot stops moving and applying force. After adapting to the current configuration, the robot keeps repeating the pattern in the task phase plot. The continuity in the plot during levering and unscrew-driving can be interpreted as a successful task. Additionally, it shows robustness to varying external forces.

The position accuracy of aligning the tool requires constant end-effector orientation during the processes, which is crucial to establishing and maintaining contact. The norm of the angular velocity w.r.t the end effector in Fig. 6 is $\approx 0\,\mathrm{rad/s}$ both in levering and unscrew-driving, which shows us the robot maintains the contact robustly during the process.

Force-motion profile errors can be analyzed in the force and position plots, as shown in Fig. 7 and Fig. 8. Notably, the commanded force to the robot is the desired force times ρ_{frc}. For instance, as shown in Fig. 7, ρ_{frc} decrease to app. 0.6, such that the commanded force $\boldsymbol{f}_z$ reduced from $12\,\mathrm{N}$ to app.

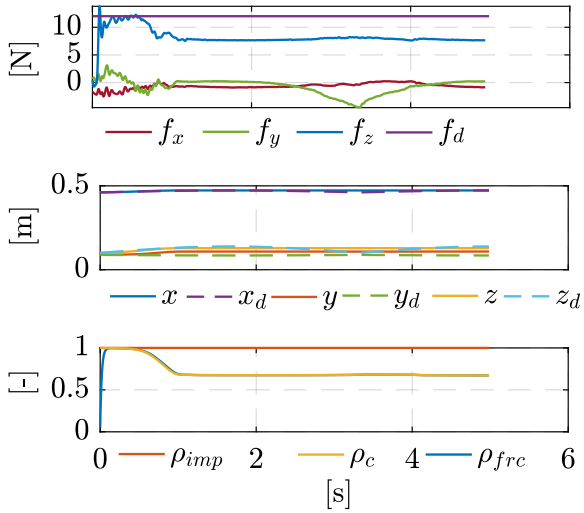


Fig. 7: **Performance metrics results for levering.** Top: desired vs. measured force in the end effector frame, Middle: desired vs. actual motion in the base frame, and Bottom: controller shaping functions.

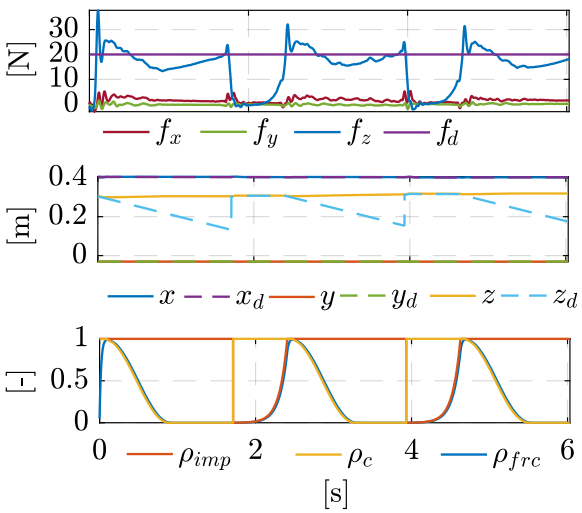


Fig. 8: **Performance metrics results for unscrew-driving.** Top: desired vs. measured force in the end effector frame, Middle: desired vs. actual motion in the base frame, and Bottom: controller shaping functions.

7.2 N. While tracking the motion and forces as accurately as possible is important, for successful execution, levering and unscrew-driving are the processes that need to tolerate the force and displacement imperfections due to, e.g., a loose screw moving in the thread. Here, we can comment that our tactile robot programming framework allows, yet limits the force and displacement tolerances by the control shaping functions such that they regulate the stiffness and commanded forces for the certain thresholds $S_t, S_{\min}$, and δ_c that the human experts set beforehand to allow acceptable deviations while ensuring successful executions.

In general, the focus in tactile skills relies on contact/tool alignment ρ_c and compliant behavior ρ_{imp}, namely, force and displacement tolerance, such that after specific tool alignment error, the robot should stop applying force or if the impedance shaping is activated due to the motion error and external forces occurred. Specifically, the levering process is analyzed and based on the results in Fig. 7, the contact/tool alignment during sinusoidal motion or force-displacement tolerance is crucial to achieving a robust levering process as the lid moves primarily, and the robot should maintain contact between the tool and the lid during the motion.

In addition, unscrew-driving demands compliant behavior or displacement tolerance, as can be interpreted from the results in Fig. 8. It is also predictable as the robot should allow the screw to move upwards while pushing it to trigger the screwdriver, and this requires the screwdriver to be perpendicular to the screw to maintain contact. The impedance shaping is activated if the contact is about to be broken, leading to external force and motion error. Here, the robot stops force control while compliant due to decreasing ρ_{imp}. The stiffness is fully recovered in the current configuration, and the robot re-starts applying force after correct tool alignment. The authors would like to mention that further study should focus on deciding S_t, $\rho_{\min}$, and δ_c instead of fine-tuning the current surface material properties, such as friction and rigidity.

V. Conclusion

Cutting-edge tactile robots offer improved adaptability and flexibility, but still, their programming using force- or impedance control is relatively static and requires expert knowledge. Reaching the full potential of flexible manipulation task execution in real-world scenarios requires highly simplified programming for non-experts. Thus, we propose a task-oriented tactile robot programming framework to successfully deploy tactile robotics in manufacturing that exploits object-centric tactile skill definition.

The core concept consists of a) the basic knowledge of the forces and motion constraints a real-world object is subject to; b) using a force-velocity representation called task phase showing the change of these constraints during the task; and c) transferring this intuitive representation into the robot control, without requiring the robot operators expert knowledge about controller parameterization. We apply this scheme to establish constraint-based unified force-impedance control for common manipulation skills, which we validate in real-life experiments using the tasks of unscrew-driving and levering with a Franka Emika robot manipulator by evaluating in terms of the proposed analysis metrics i) position accuracy, ii) displacement tolerance, iii) force tolerance, and iv) force and motion error. Finally, our approach to simplify tactile robot programming and enable the direct translation between task-level constraints and robot control is a potential solution for increased robotic deployment in flexible manufacturing lines.

Future work will focus on blending the tactile disassembly skills by extending our task-oriented tactile robot programming approach.

References

[1] E. C. Balta, K. Jain, Y. Lin, D. Tilbury, K. Barton, and Z. M. Mao, "Production as a service: A centralized framework for small batch manufacturing," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, 2017, pp. 382–389.

[2] 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.

[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] A. Lambert and S. Gupta, "Disassembly modeling for assembly, maintenance, reuse and recycling," 12 2004.

[5] W. S. Newman, "Stability and Performance Limits of Interaction Controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 114, no. 4, pp. 563–570, Dec. 1992, _eprint: https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/114/4/563/5551865/563_1.pdf. [Online]. Available: https://doi.org/10.1115/1.2897725

[6] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American Control Conference*, 1984, pp. 304–313.

[7] 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.

[8] 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.

[9] M. Iskandar, C. Ott, A. Albu-Schäffer, B. Siciliano, and A. Dietrich, "Hybrid force-impedance control for fast end-effector motions," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 3931–3938, July 2023.

[10] O. Khatib, "Inertial properties in robotic manipulation: An object-level framework," *The International Journal of Robotics Research*, vol. 14, no. 1, pp. 19–36, 1995. [Online]. Available: https://doi.org/10.1177/027836499501400103

[11] J. D. Schutter, T. D. Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007. [Online]. Available: https://doi.org/10.1177/0278364907078091 07

[12] T. Migimatsu and J. Bohg, "Object-centric task and motion planning in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 844–851, 2020.

[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] A. Kramberger, A. Gams, B. Nemec, 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.

[20] 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.

[21] 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

[22] 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)*, 2022, pp. 9604–9611.

[23] 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.

[24] 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.

[25] 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

[26] 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

[27] European Union, 1994-2023, "Self-reconfiguration of a robotic workcell for the recycling of electronic waste," https://cordis.europa.eu/project/id/871352/reporting/de, Last accessed on 2023-06-15.

[28] 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.

[29] K. Karacan, R. J. Kirschner, H. Sadeghian, F. Wu, and S. Haddadin, "The inherent representation of tactile manipulation using unified force-impedance control," December 2023, p. under publication, 2023 IEEE 62nd Conference on Decision and Control (CDC).

[30] A. De Luca and C. Manes, "Modeling of robots in contact with a dynamic environment," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 542–548, Aug 1994.

[31] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, ser. Advanced Textbooks in Control and Signal Processing. Springer London, 2010. [Online]. Available: https://books.google.de/books?id=jPCAFmE-logC

[32] K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, 2016.

[33] E. Shahriari, 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

[34] 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.

[35] S. Hjorth, E. Lamon, D. Chrysostomou, and A. Ajoudani, "Design of an energy-aware cartesian impedance controller for collaborative disassembly," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 7483–7489.

[36] 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.

A.4 "Determining Exception Context in Assembly Operations from Multimodal Data"

This is a copy of the open access paper: 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)*. 2024.

sensors

MDPI

Article

Determining Exception Context in Assembly Operations from Multimodal Data

Mihael Simonič [1,2,*], Matevž Majcen Hrovat [1], Sašo Džeroski [3,4], Aleš Ude [1,2] and Bojan Nemec [1,4]

[1] Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Jamova Cesta 39, 1000 Ljubljana, Slovenia
[2] Faculty of Electrical Engineering, University of Ljubljana, Tržaška Cesta 25, 1000 Ljubljana, Slovenia
[3] Department of Knowledge Technologies, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[4] Jožef Stefan International Postgraduate School, Jamova Cesta 39, 1000 Ljubljana, Slovenia
* Correspondence: mihael.simonic@ijs.si

Abstract: Robot assembly tasks can fail due to unpredictable errors and can only continue with the manual intervention of a human operator. Recently, we proposed an exception strategy learning framework based on statistical learning and context determination, which can successfully resolve such situations. This paper deals with context determination from multimodal data, which is the key component of our framework. We propose a novel approach to generate unified low-dimensional context descriptions based on image and force-torque data. For this purpose, we combine a state-of-the-art neural network model for image segmentation and contact point estimation using force-torque measurements. An ensemble of decision trees is used to combine features from the two modalities. To validate the proposed approach, we have collected datasets of deliberately induced insertion failures both for the classic peg-in-hole insertion task and for an industrially relevant task of car starter assembly. We demonstrate that the proposed approach generates reliable low-dimensional descriptors, suitable as queries necessary in statistical learning.

Keywords: sensor fusion; predictive clustering trees; autonomous exception handling; autonomous assembly; peg-in-hole

Citation: Simonič, M.; Majcen Hrovat, M.; Džeroski, S.; Ude, A.; Nemec, B. Determining Exception Context in Assembly Operations from Multimodal Data. *Sensors* **2022**, *22*, 7962. https://doi.org/10.3390/s22207962

Academic Editors: Abdelaziz Benallegue and A. El Hadri

Received: 27 August 2022
Accepted: 14 October 2022
Published: 19 October 2022

1. Introduction

Assembly tasks, such as inserting parts into fixtures, are among the most common industrial applications [1]. Robot assembly typically requires a good understanding of the procedure and knowledge about part properties and geometry [2]. Therefore, most of the deployed robotic systems used today are carefully programmed [3]. As such, they are limited to performing a specific assembly task in structured environments without external disturbances. Nevertheless, they can fail due to various errors that cannot be foreseen in advance. Possible causes include deviations in the geometry of the workpiece, imprecise grasping, etc. In such cases, it is necessary for the operator to manually eliminate the cause of the error, reset the system, and restart the task [4]. Current robotic systems do not learn from such situations. If a similar situation repeats, human intervention is needed again. To ensure robust execution of robot assembly tasks, it is increasingly important to handle such exception scenarios autonomously, possibly by incorporating previous experience.

The first step toward building such an autonomous system is to determine the reason for the failure. For example, a robot may fail to assemble two parts, but it is unclear whether it has failed because the parts do not match or because of an ineffective manipulation strategy [5]. Understanding or at least classifying the reason for the failure is, therefore, crucial for the successful design of a corresponding exception policy.

Recently, we have proposed a framework for the learning of exception strategies [6], which is based on determining the context of the failure. The extracted context is associated with different robot policies needed to resolve the cause of the error. In the event of an error,

the system stops, and the robot switches to gravity compensation mode. Using incremental kinesthetic guidance [7], the operator performs a sequence of movements to allow the continuation of regular operation. First, the robot builds a database of corrective actions and associates them with the detected error contexts. Then, using statistical methods, it computes an appropriate action by generalizing the corrective actions associated with different error contexts. In this way, the robot becomes increasingly able to resolve errors on its own and eventually does not require human intervention to resolve assembly failures anymore (see Figure 1).

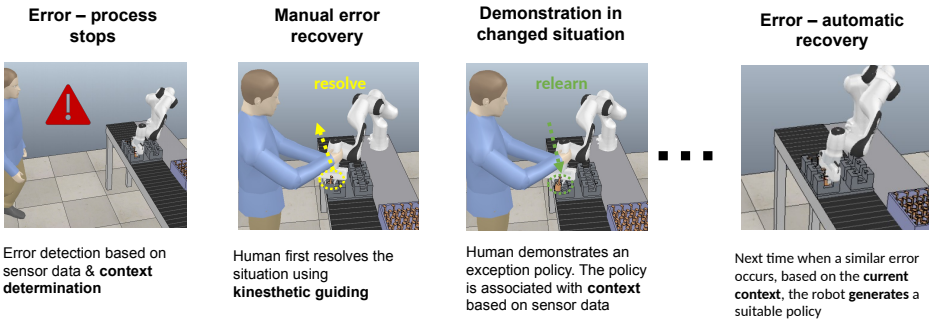


Figure 1. Simplified exception handling workflow [6]. This paper is about context determination, which is an essential requirement of the workflow.

Modern robotic systems are equipped with a wide variety of sensors that can be used to detect a possible failure. On the other hand, context determination can be seen as inferring the circumstances that have resulted in the given outcome. The first step in the exception strategy learning framework is, therefore, context determination, which can be seen as inferring a minimal representation of the circumstances that have resulted in the given outcome. As raw sensor data are usually high-dimensional, they cannot be directly used with statistical learning methods that allow us to relate the observed state (context) to the previous states and generate an appropriate robot action to resume regular operation. Moreover, for reliable context determination, it is often necessary to combine complementary information from different sensor modalities. This process is known as data fusion and can lead to improved accuracy of the model compared to a model based on any of the individual data sources alone [8]. Ensemble learning methods have proven to be appropriate for addressing multimodal classification and regression problems in many domains [9].

We propose to train models that generate low-dimensional context descriptions based on multimodal sensor data from vision systems and force-torque sensors. In this sense, context determination can be defined as determining of the type of circumstances from multimodal data. We use an intermediate-fusion approach, where we first extract modality-specific features, as shown in Figure 2. We rely on a state-of-the-art neural network model for image segmentation to extract features from images, whereas we use contact point estimation to extract data from the measured forces and torques. To generate a low-dimensional context description of the circumstances that resulted in the given outcome from the extracted features, we use ensembles of predictive clustering trees (PCTs) [10], which are well suited for handling hierarchical multi-label classification (HMLC) tasks. With the proposed hierarchical approach, the training of the context estimation model can be divided into multiple phases, allowing for an incremental approach. The time-consuming training of the image segmentation model only needs to be performed once, whereas fast training of the high-level ensemble model can be performed each time a new class needs to be added.

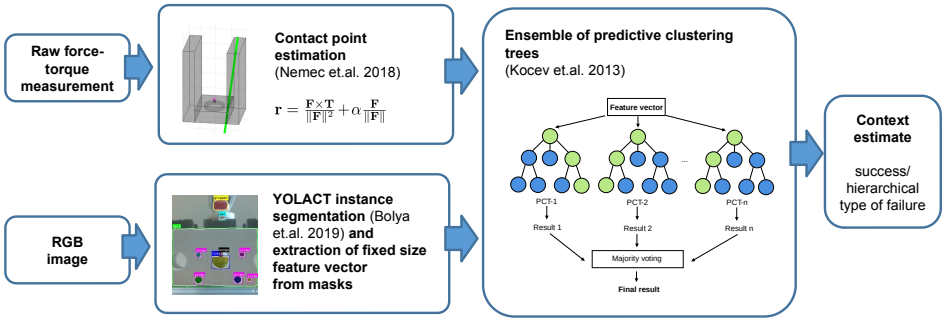


Figure 2. Context determination pipeline. Features are processed for each modality separately (Sections 3.4–3.6) and later merged by using ensembles of predictive clustering trees (Section 3.7). References: [10–12].

The evaluation of the proposed approach comprised two scenarios. Peg-in-hole assembly is chosen as the first use case because it reflects the typical complexity of industrial assembly tasks [13]. We show that it is possible to apply the method to other tasks by performing an evaluation of a car starter assembly task. The approach can also be applied to further situations. Apart from identifying error classes, the only process-specific step is selecting the image segments of interest and training the instance segmentation model accordingly.

The main highlights of our context determination approach are:

- with the use of multimodal data we get an improved predictive performance of ensembles;
- it is easy to add new classes (this is necessary as we discover new failure cases incrementally as they arise);
- the approach generalizes well to new cases (we can make useful predictions based on a model trained on a limited amount of data).

The paper is organized into six sections. Section 2 reviews current strategies to handle exceptions in robot assembly and the usage of tactile and vision sensor data in robotics. The details of our approach are presented in Section 3. In Section 4, the predictive performances of different variations of the developed model for context determination are evaluated. Section 5 discusses the results as well as future plans. We conclude with a brief summary of the paper in Section 6.

2. Related Work

Fault-tolerant robotic systems that are able to detect and autonomously deal with system failures have been the subject of research for many years [14]. While some researchers are concerned with fault tolerance in medical, space, nuclear, and other hazardous applications, our research focuses on industrial processes, where we can ensure operator presence, at least in the learning phase. In such environments, strategies based on various heuristic movement patterns (random search, spiral search, dithering, vibrating, etc.) are often used to deal with unexpected situations [15,16].

Laursen et al. [17] proposed a system that can automatically recover from certain types of errors by performing the task in reverse order until the system returns to a state from which the execution can resume. Error recovery can also be performed collaboratively so that the robot recognizes when it is unable to proceed and asks for human intervention to complete the task [18]. Recently, it was proposed to use ergodic exploration to increase the insertion task success rate based on information gathered from human demonstrations [19]. Another method exploits variability in human demonstrations to consider task uncertainties and does not rely on external sensors [2].

On the other hand, another line of research highlights the importance of sensorimotor interaction for future learning methods in robotics [20]. During the assembly task execution, monitoring the exerted forces and torques is necessary to prevent damaging the parts or

robot [6]. These data can be used to avoid large impact forces exploiting compliance and on-line adaptation [21], to speed up the process in the subsequent repetitions [22], to determine contact points and learn contact policies [11], and to predict [23] or classify [24] robot execution failures.

Various previous works have studied the idea to calculate contact points based on force-torque measurements [11,25–27]. In our previous work [6], we used force-torque measurements to calculate trajectory refinements that enable successful insertion despite the grasping error. Force-torque data carry enough information to generate an appropriate refinement, given that we already know the nature of the problem (orientation vs. position grasping error). However, in general, sufficient information cannot be obtained from force-torque data only. For example, the policies for correcting unsuccessful assembly attempts often depend on which part of the peg is in contact with the environment [28]. Thus additional sensors are required. Using a force-torque sensor only is also problematic due to sensor noise. Many applications in process automation, therefore, rely on vision systems to extract the necessary information. While vision can be quite sensitive to calibration errors and typically requires a well-designed workcell to ensure optimal lighting conditions and avoid occlusions, it is fast and can be used for the global detection of multiple features [29].

The advantage of combining visual and contact information has been investigated in multiple works in robotics over a longer period of time. This research includes dimension inspection [30], object recognition [31], and localization [29]. In the context of robot assembly, visual and tactile sensing has been used to continuously track assembly parts using multimodal fusion based on particle filters [32] and Bayesian state estimation [13]. We share the principal idea of combining data from visual and force-based sensing. We want to further develop these concepts towards structured representations of the task context in order to develop an integrated solution for the automatic handling of failures in assembly processes.

Multimodal fusion combines information from a set of different types of sensors. Detection and classification problems can be addressed more efficiently by exploiting complementary information from different sensors [8]. Different methods for data fusion from multimodal sources exist. Generally, we can distinguish three levels of data fusion: early fusion, where the raw data are combined ahead of feature extraction and the result is obtained directly; intermediate fusion, where modality-specific features are extracted and joined before obtaining the result; and late fusion, where the modality-wise results are combined [33,34]. It may seem that combining multimodal data at the raw data level should yield the best results, as there would be no loss of information. However, due to the unknown inter-dependencies in raw data, fusion at a higher level of abstraction may be a more helpful approach in practice [35].

Ensemble learning is a general approach in machine learning that seeks better predictive performance by combining the predictions from multiple models [9]. Ensemble learning methods have proven to be an appropriate tool to address multimodal fusion, achieving comparable results or even outperforming other state-of-the-art methods in many other domains [36,37]. The idea of ensemble learning is to employ multiple models and combine their predictions. This is often more accurate than having a complex individual model to decide about a given problem. Data from heterogeneous sources, such as different modalities [38], can easily be combined. In this paper, we consider an ensemble of predictive clustering trees (PCTs) [10] to perform hierarchical multi-label classification (HMLC). PCTs are a generalization of ordinary decision trees and have been successfully used for a number of modeling tasks in different domains, i.e., to predict several types of structured outputs, including nominal/real value tuples, class hierarchies, and short time series [39,40]. A detailed description of PCTs for HMLC is given by Vens et al. [41].

3. Materials and Methods

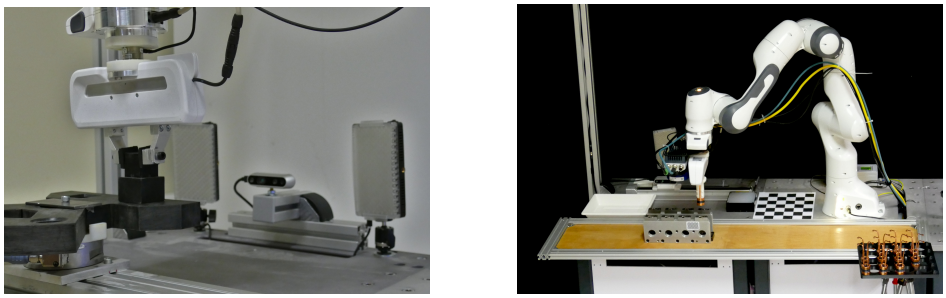
In this section, we first describe the robotic workcell used to collect the data described in Section 3.1. The assembly tasks to perform the evaluation are presented in

Sections 3.2 and 3.3. Next, we present our contact determination method in detail. The approach consists of three main parts. First, force-torque sensor measurements are processed using a method for contact point estimation described in Section 3.4. Data from the vision modality are passed through a neural-network model performing instance segmentation (Section 3.5), and features are extracted from the instance masks using standard computer vision methods (Section 3.6). Finally, the features from both modalities are combined using an ensemble of predictive clustering trees, as described in Section 3.7.

3.1. Experimental Environment

In our research, we focused on robotic assembly and considered two tasks—square peg insertion using the Cranfield benchmark [42] and the industrially relevant car starter assembly [43]. To perform both tasks and collect data for the context determination model, we rely on a modular workcell design that enables easy mounting of task-specific equipment, e.g., robots, sensors, and auxiliary devices [44] and a ROS-based software architecture that allows for easy integration of new components [45].

The workcell consists of two modules and a control workstation. The first module supports a seven-degree-of-freedom collaborative robot, Franka Emika Panda. The other module is equipped with sensors and cameras to support the specific assembly process, as shown in Figure 3. An Intel RealSense D435i RGB-D camera is used to supervise the insertion visually. To control the light conditions, we utilize an adaptive lighting setup based on Aputure Amaran F1 LED panels. Besides images, we can also capture forces and torques. To measure the forces exerted in the peg-in-hole task, we utilize an ATI Delta force-torque sensor mounted under the Cranfield benchmark plate. To measure the forces exerted in the copper ring insertion task, we utilize a wrist-mounted ATI Nano25 sensor. Additional peripheral devices, visible in Figure 3b, are used to assist different aspects of the human–robot collaboration, which is not the subject of this paper.



(**a**) (**b**)

Figure 3. Experimental setup for testing exception strategy learning using multimodal data. (**a**) Setup for the *Cranfield benchmark*. (**b**) Setup for the copper ring insertion task.

To perform the assigned tasks, we applied a passivity-based impedance controller for manipulators with flexible joints [46]. We assume that the controller parameters were carefully tuned to ensure stable and compliant operation in unstructured environments, where we can expect deviations in task parameters.

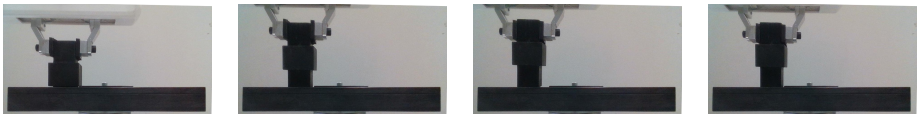
3.2. Peg-in-Hole Insertion Task

The peg-in-hole (PiH) task is an abstraction of the most typical task in assembly processes, accounting for approximately 40% of the total assembly tasks [47]. Over time, many different approaches and control strategies to address this problem have emerged. Nowadays, the efficiency of the applications is enhanced by integrating machine vision and other sensor technology accompanied by artificial intelligence approaches. As such, it is a commonly accepted benchmark in assembly research.

To generate a dataset for comparing different methods for failure context determination, we repeatedly executed the task of square peg insertion using the Cranfield benchmark.

It requires the insertion of a square peg into the corresponding hole of the base plate. The main challenge is the transition of the peg from free space into a highly constrained target hole. Relatively tight tolerances combined with imprecise positioning can prevent the successful completion of the insertion process.

Different factors influence the outcome of the PiH task. For instance, both imprecise grasping and wrong target position can lead to insertion failure, as shown in Figure 4.



(**a**) (**b**) (**c**) (**d**)

Figure 4. Different outcomes of the PiH task. (**a**) Successful insertion with correct parameters. (**b**) Insertion failure due to grasping error. (**c**) Insertion failure due to a positional error in the x-direction. (**d**) Insertion failure due to a positional error in the y-direction.

In order to collect a database of different insertion outcomes, we deliberately set different positional offsets in either the x or the y-direction from -10 to 10 mm in 1 mm steps. In this way, we generated 40 cases that resulted in insertion failure and 1 that led to successful insertion. Due to the offset, the robot fails to insert the peg and stops the execution when it exceeds a force threshold, set to 10 N in the z-direction. The insertion is successful when there is no positional offset in either direction.

In total 180 data entries were recorded. The robot attempted to insert the peg into the plate three times for each failure case. Additionally, 60 successful attempts were recorded. Robot pose, force and torque measurements, and RGB images of the outcome were captured when the insertion was complete or stopped (force threshold exceeded).

Note that the data can be organized hierarchically into three categories (no error, positional error in x direction, and positional error in y direction). The latter two categories can be further split in half depending on the direction of the error ($x/+$, $x/-$, $y/+$, $y/-$). Finally, we can split based on the magnitude of the error (e.g., $x/+/2$, meaning that we have a 2 mm error in the $x+$ direction).

3.3. The Task of Inserting Copper Sliding Rings into Metal Pallets

The car starter assembly process includes inserting copper sliding rings into metal pallets, as shown in Figure 5. This can be categorized as a multiple peg-in-hole problem, as it is necessary to insert the bottom of the copper ring and both upper part lugs correctly. The insertion process is challenging due to the deformability of the sliding rings. This task has been taken from a real production process where it is performed manually. Previous automation attempts have failed due to the low success rate that was achieved. To ensure robust insertion, we proposed to use the exception strategy framework [43].

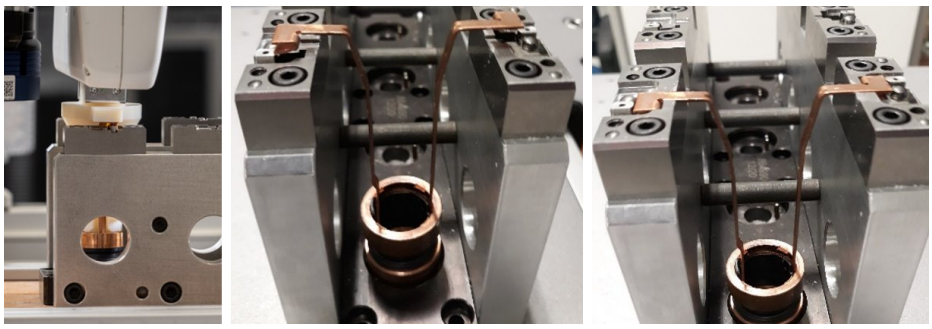


Figure 5. Left: Insertion of the ring into a modeling fixture. Center: Incorrect insertion. Right: Correct insertion.

We have collected a database of twelve copper ring insertions, which includes both successful insertions and deliberately induced insertion failures. The failures were caused by the displacement of the target position for insertion in the x and y directions:

- no displacement, leading to successful insertion;
- positional displacement in the x direction, with Δp_x between 1 and 3 mm in 1 mm steps;
- positional displacement in the y direction with Δp_y between 1 and 3 mm in 1 mm steps, both leading to unsuccessful insertion.

Additionally, we recorded insertion attempts with deformed parts, which also led to an unsuccessful insertion. For each of the cases, we recorded at least four insertion attempts. The process was repeated for all four slot positions in the molding cast. Each entry consists of a snapshot of the outcome of the insertion task (cropped RGB image) and the time series of force $\mathbf{F} = (F_x, F_y, F_z)$ and torque $\mathbf{T} = (T_x, T_y, T_z)$ measurements. Similarly to the PiH dataset, the gathered data can be organized hierarchically.

3.4. Force-Torque Data Extraction: Contact Vector Estimation

In our previous work [6], we have considered only errors due to the offset in the grasping angle and have shown that force-torque data can be used to determine a suitable context descriptor using principal component analysis (PCA), which correlated most strongly with the grasping error. Such a dimensionality reduction is beneficial because the generation of an appropriate refinement trajectory based on statistical learning is sensitive to the dimension of the feature space. Another possible approach to reduce the dimensionality of force-torque measurements is to determine contact points [48].

The point of contact between two parts can be estimated based on the relationship between force $\mathbf{F}$, torque $\mathbf{T}$, and lever $\mathbf{r}$ by using the following formulation [11]

$$\mathbf{r}(\alpha) = \frac{\mathbf{F} \times \mathbf{T}}{\|\mathbf{F}\|^2} + \alpha \frac{\mathbf{F}}{\|\mathbf{F}\|}, \tag{1}$$

where α is a suitably chosen constant so that the vector $\mathbf{r}$ touches the environment as illustrated in Figure 6a. The measured forces and torques must be expressed in the robot end-effector coordinate system.

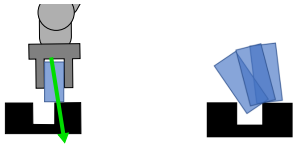
However, the contact point estimation cannot always distinguish between the different types of errors, as illustrated in Figure 6b. Thus, forces and torques, as well as the positional data, cannot uniquely determine the context. In order to resolve this ambiguity, we introduce another modality, as discussed in the remainder of this paper.

Nevertheless, Equation (1) provides a suitable representation that can distinguish between different conditions of the same outcome type. A graphical example of contact vector estimation for both experiments is shown in Figure 7.

Our preliminary results have shown that the inclusion of raw force-torque data as features decreases the performance of the final model. Thus the feature vector for the FT modality was chosen to include only the contact point vector estimate. For each example $k \in \mathcal{E}$, the feature vector is calculated as:

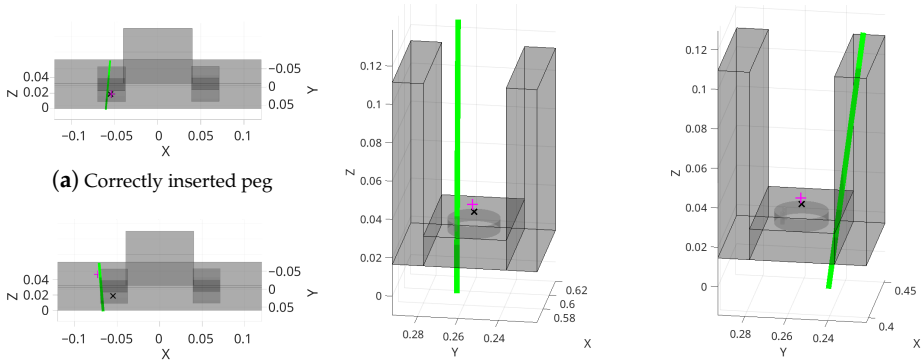
$$\mathbf{f}_{\mathrm{FT}}^{k} = (r_x, r_y, r_z), \tag{2}$$

where r_x, r_y, r_z are components of the vector $\mathbf{r}(\alpha)$.



(**a**) (**b**)

Figure 6. A scheme depicting contact point estimation (**a**) and another example where the grasped part comes into contact with the environment at the same point (**b**). The robot and the gripper are represented by the dark gray shape, whereas the grasped part and the environment are shown in blue and black, respectively. The contact vector estimate is shown with a green arrow.



(**a**) Correctly inserted peg (**b**) Not fully inserted peg (**c**) Correctly inserted ring (**d**) Wrongly inserted ring

Figure 7. Contact vector estimation in four examples of the considered assembly tasks. A schematic model of the Cranfield base plate and the casting mold is shown in (**a**–**d**), respectively. The green line shows the contact vector, whereas the pink plus symbol shows the robot's tool center point (TCP) at the time of contact and the black cross shows the target reference position.

3.5. Vision Data Extraction: Instance Segmentation with YOLACT

We applied deep neural networks (DNN) to perform feature extraction from image data. They provide good flexibility because pre-trained NN models and frameworks can be re-trained by using a custom dataset for a specific use case, in contrast to the classic computer vision (CV) algorithms, which tend to be more domain specific [49]. Compared to the traditional computer vision methods (e.g., edge detection), they often require less manual fine-tuning.

Various convolutional neural networks (CNNs) have proven to be suitable for analyzing image data. An essential issue with a custom network that directly extracts features is that retraining is needed when a new error class is added or the camera position is changed. For these reasons, we rely on models that are designed to be less prone to changes in object position in the picture. This has been extensively studied in object detection and instance segmentation models. Instance segmentation is an enhanced type of object detection that generates a segmentation map for each detected instance of an object in addition to the bounding boxes.

In order to meet the above-listed requirements, we used the state-of-the-art instance segmentation model YOLACT [12]. YOLACT builds upon the basic principles of RetinaNet [50] with the Feature Pyramid Network [51] and ResNet-101 [52] as a convolutional backbone architecture for feature extraction. It utilizes a fully convolutional network to directly predict a set of prototype masks on the entire image. Lastly, a fully connected layer assembles the final masks as linear combinations of the prototype masks, followed by bounding box cropping. Compared to most of the previous instance segmentation approaches, such as Mask R-CNN [53], which are inherently sequential (the first image is scanned for regions with object candidates, then each of them is processed separately), YOLACT is a one-stage algorithm that skips this intermediate localization step. This allows for nearly real-time performance. By using shallower computational backbones, such as ResNet-50, even faster performance can be achieved at a minimal accuracy cost when compared to ResNet-101 [12].

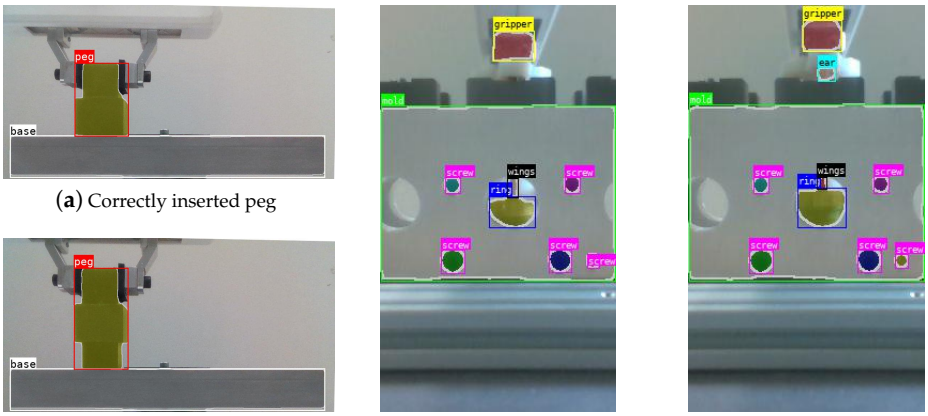
The (re)training of YOLACT requires labeled images and ground truth image masks. We have used an open-source graphical image annotation tool, Labelme, to annotate images in our datasets (https://github.com/wkentaro/labelme, accessed on 13 October 2022). For the PiH dataset, we manually annotated 30 images for each position using four different light settings. We manually annotated 10 images for each position using two different light settings for the copper ring insertion dataset. We split the annotated datasets into training and validation partitions, with 80% and 20% of the data, respectively. Finally, the annotations had to be transformed into a format compatible with the YOLACT

training script (COCO). For this purpose, we prepared an open-source tool—labelme2coco (https://github.com/smihael/labelme2cocosplit, accessed on 13 October 2022).

In the proposed pipeline, we configured YOLACT to use a computationally lighter ResNet-50 as the backbone. This enabled us to use original-resolution images while retaining high training and inference speed. We trained two models for each of the above-presented datasets. During training, the algorithm used a batch size of 8, weight decay of 0.0005, and image size of 1280 × 720 pixels (PiH dataset) or 221 × 381 pixels (copper ring insertion dataset). The initial learning rate was set to 0.001. The model was trained for 40,000 iterations, and the decay rate of 0.1 was applied once each 10,000 iterations. The training took 8 h on a GeForce GTX 1060 GPU for the PiH dataset and 6 h for the copper ring insertion dataset.

Once the models were trained, we deployed them to a workstation in the robotic workcell. The integration was done using a modified yolact_ros package (https://github.com/smihael/yolact_ros, accessed on 13 October 2022), which also allows using the learned model for inference without a GPU, thus lowering the computational requirements.

The results are shown in Figure 8. The PiH model is trained to distinguish between the peg and the base plate, whereas the copper ring insertion model is able to distinguish the following segments: gripper, mold, screws, ring base, wings, and ears (lugs).



(**a**) Correctly inserted peg

(**b**) Not fully inserted peg

(**c**) Correctly inserted ring

(**d**) Not fully inserted

Figure 8. Bounding boxes and masks obtained by YOLACT using snapshots of the outcome of both tasks as input. For the PiH task (**a**,**b**), the base plate and peg are detected, regardless of the position/occlusion of the latter. In the copper ring insertion task (**c**,**d**), the gripper, mold, screws, ring base, wings, and ears are detected. Notice that ears are only detected when the part is not fully inserted.

Note that the PiH model can be equally used for any of the two insertion slots in the PiH task. Likewise, the copper rings model can be used for any of the four slots in the copper ring insertion task. Since the model is position invariant, meaning that the model is able to correctly mark the area of different image parts regardless of where in the image they appear, we can apply it for the analysis of new error cases.

3.6. Extracting a Fixed-Size Feature Vector from Instance Segmentation Results

Using the trained YOLACT segmentation models, we obtain bounding boxes and image masks for all images in the PiH and copper ring insertion datasets. The information obtained from instance segmentation needs to be further processed to be used in further steps of the pipeline, as shown in Figure 9. We extract fixed-size feature vectors, as ensembles of predictive clustering trees do not operate over image masks.

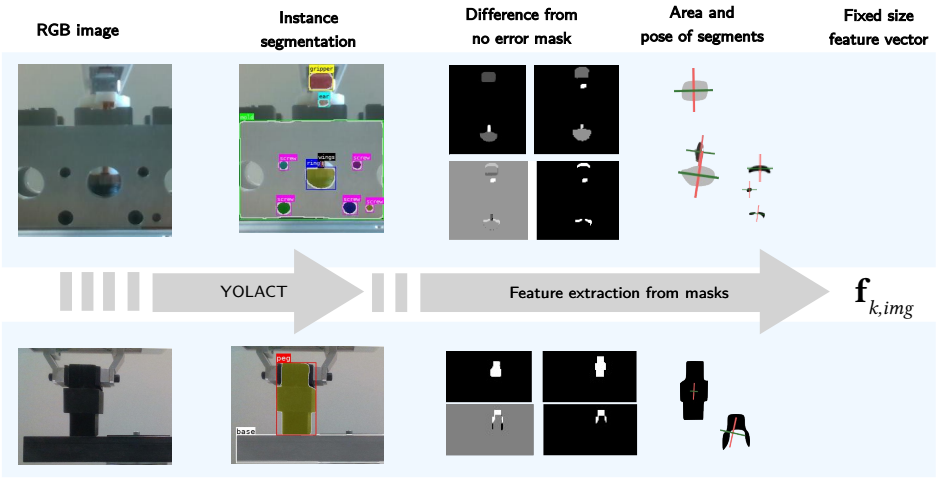


Figure 9. Image features extraction pipeline. The RGB snapshot of a situation is processed by a YOLACT model to obtain masks of different parts of interest. The obtained masks are then processed to obtain a low-dimensional fixed-size feature vector.

An image is represented as a $w \times h \times 3$ matrix of pixels $\mathbf{I}(x, y, c) \in \{0, 1, \ldots, 255\}$, representing the RGB color channels. The image can contain multiple instances of different objects. For each segmented object instance s, we obtain its type, the bounding box, and the mask. The bounding box $\mathbf{B}_s$ is given as a pair of pixel coordinates of two diagonal corners $\{(x_1, y_1), (x_2, y_2)\}$. The bounding box can be represented by the centroid $\mathbf{c}_s$, width w_s, and height h_s of the rectangle

$$\mathbf{c}_s = [c_{s,x}, c_{s,y}]^\top = \left[\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right]^\top, \tag{3}$$

$$w_s = x_2 - x_1, \tag{4}$$

$$h_s = y_2 - y_1. \tag{5}$$

The pixels belonging to the specific object instance s are represented with masks. Each mask is a $w \times h$ binary matrix $\mathbf{M}_s \in \mathbb{B}^{w \times h}$, which tells whether a pixel is part of the mask or not. Using PCA, we determine the first principal component for each instance's mask $\mathbf{e}_{s,1} = (x, y)$. This result can be used to calculate the orientation of the part in the image plane (visualized in Figure 10)

$$\varphi_s = \arctan\left(\frac{-y}{x}\right) - \frac{\pi}{2}. \tag{6}$$

(a) **(b)** **(c)** **(d)** **(e)**

Figure 10. The extracted peg mask for different executions of the copper ring insertion task. Red and green lines show the principal directions and determine the mask's orientation. (**a**) $\Delta p_x = -10$ mm; (**b**) $\Delta p_x = -5$ mm; (**c**) $\Delta p_x = 0$ mm; (**d**) $\Delta p_x = 5$ mm; (**e**) $\Delta p_x = 10$ mm.

Additionally, we calculate the pixel area of each instance mask as a total number of all true elements in the instance matrix

$$a_s = \sum_{x=0}^{w} \sum_{y=0}^{h} m_s(x, y). \tag{7}$$

In both experiments, we define a set of object of interests $\mathcal{S}_{\text{int}}$ (see Figure 8). For the peg-in-hole insertion task, it consists of the peg only, while for the copper ring insertion task, $\mathcal{S}_{\text{inst}}$ contains the gripper, ring, wings, and ear. Note that additional features, e.g., screws on the molding cast or the base of the Cranfield benchmark, can be used as calibration features. In our case, this was not needed as the datasets were recorded with a fixed camera position.

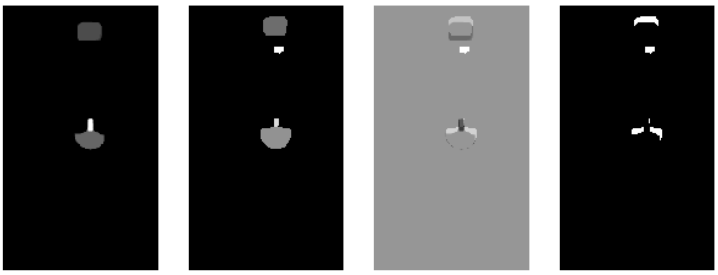
From the set of examples with no visible errors $\mathcal{E}_0$, we calculate the average segment mask $\overline{\mathbf{M}}_s$ for each object instance s of interest from $\mathcal{S}_{\text{int}}$. The average mask is calculated as the element-wise mean of the mask matrices:

$$\overline{\mathbf{M}}_s = \frac{1}{|\mathcal{E}_0|} \sum_{k \in \mathcal{E}_,} \mathbf{M}_s^k \in \mathbb{R}_+^{w \times h}. \tag{8}$$

For other examples, we compute the difference between theirs masks and the average mask of examples with no error $\widetilde{\mathbf{M}}_{s,\text{diff}}^k = \mathbf{M}_s^k - \overline{\mathbf{M}}_s$, and take only its positive elements to define binary matrix $\mathbf{M}_{s,\text{diff}}^k$,

$$m_{s,\text{diff}}^k(x, y) = \begin{cases} 1, & \widetilde{m}_{s,\text{diff}}^k(x, y) > 0 \\ 0, & \text{otherwise} \end{cases}. \tag{9}$$

An example is shown in Figure 11. For each of the obtained difference segments, we then calculate its center $\mathbf{c}_{s,\text{diff}} = [c_{s,\text{diff},x}, c_{s,\text{diff},y}]^\top$ and pixel area $A_{s,\text{diff}}$ using Equations (3) and (7), respectively.



(**a**) Average no error (**b**) dy = −2 mm (**c**) Difference (**d**) Binary difference

Figure 11. From left to right: (**a**) average segmentation mask for the successful copper ring insertion attempts, (**b**) segmentation mask for a failed insertion attempt (positional error in the y-direction), (**c**) difference of the segmentation masks, and (**d**) positive part of the difference.

In this way, we obtain an image feature vector for each example $k \in \mathcal{E}$ and object of interest $s \in \mathcal{S}_{\text{int}}$:

$$\mathbf{f}_s^k = \left[\mathbf{c}_s^{k\top},\ w_s^k,\ h_s^k,\ \phi_s^k,\ a_s^k,\ \mathbf{c}_{s,\text{diff}}^{k\top},\ a_{s,\text{diff}}^k\right]^\top. \tag{10}$$

3.7. Combining Image Features and Force-Torque Measurements Using Ensembles of Predictive Clustering Trees

We formulate the determination of the outcome of the insertion task as a hierarchical multi-label classification (HMLC) problem. Given the extracted image features and the estimate of the contact point, the type of outcome should be predicted. For the different types of outcomes, a hierarchy of class labels defines the direction and magnitude of the underlying error, as described below.

We applied ensembles of predictive clustering trees (PCTs) [10] for this task. PCTs are a generalization of ordinary decision trees [41]. Generally, in a decision tree, an input is entered at the top and as it traverses down the tree, the data gets bucketed into smaller and smaller sets until the final prediction can be determined. The PCT framework, however, views the decision tree as a hierarchy of clusters: the top node corresponds to the cluster containing all of the data, which is recursively partitioned into smaller clusters so that per-cluster variance is minimized [39]. In this way, cluster homogeneity is maximized, and consequently, the predictive performance of the tree is improved.

PCT ensembles consist of multiple trees. In an ensemble, the predictions of classifiers are combined to get the final prediction. For an ensemble to have better predictive performance than its individual members, the base predictive models must be accurate and diverse [9]. The diversity between trees in the PCT framework is obtained by using multiple replicas of the training set and by changing the feature set during learning, as in the random forest method [54].

In our setting, each example k from the set of examples $\mathcal{E}$ consists of all extracted features $\mathbf{f}_k$ and the corresponding label vector $\mathbf{l}_k$. The feature vectors are obtained by concatenating per-modality features:

$$\mathbf{f}_k = \left[{\mathbf{f}_{\mathrm{FT}}^k}^\top, {\mathbf{f}_1^k}^\top, {\mathbf{f}_2^k}^\top, \ldots, {\mathbf{f}_{|\mathcal{S}|}^k}^\top \right]^\top, \tag{11}$$

with $\mathbf{f}_{\mathrm{FT}}^k$ and $\mathbf{f}_s^k$, $s = 1, \ldots, |\mathcal{S}|$, defined as in Sections 3.4 and 3.6, respectively. To define the corresponding label vector, we first observe that in HLMC, each example can have multiple labels. Classes are organized in a hierarchical structure, i.e., an example belonging to a class also belongs to all of its superclasses. The resulting ordered set of classes is used to define a binary label vector $\mathbf{l}_k \in \mathbb{B}^L$. The components of $\mathbf{l}_k$ are equal to 1 if the example is labeled with the corresponding class and 0 otherwise. L denotes the number of all classes in the hierarchy.

For the PiH task, the set of labels at the first hierarchical level consists of "no error", and "x" and "y" for the error in one of the two directions. At the second level, we have "x+" and "x−" as subclasses of "x", and "y+" and "y−" as subclasses of "y". Likewise, we have "x + 1", "x − 1", "y + 1", "y − 1", "x + 2", ..., "y − 10" at the third hierarchical level. For the copper ring insertion task, the set of labels at the first hierarchical level consists of: "no error", "bad part", and "x" and "y" for the error in one of the two directions. Similarly, as in the PiH task, the sub-classes at the second and third levels are representing various magnitudes of error (ranging from −10 to 10 mm in 1 mm steps) in both considered directions (x and y).

In summary, to train the ensemble of predictive trees, we collect the dataset $\mathcal{E}$

$$\mathcal{E} = \{\mathbf{f}_k, \mathbf{l}_k\}_{k=1}^K. \tag{12}$$

After training we can use the resulting ensemble of predictive trees to predict the labels $\mathbf{l}$ given the extracted feature vector $\mathbf{f}$.

We trained multiple ensembles for two different tasks. See Section 4 for more details. We used PCT ensembles, i.e., random forests of PCTs, as implemented in the CLUS system (CLUS is available for download at http://source.ijs.si/ktclus/clus-public, accessed on 13 October 2022) for this purpose. Each ensemble consisted of 50 trees. As a heuristic to evaluate the splits in decision trees, we used the variance reduction [39]. The variance for the set of examples $\mathcal{E}$ is defined as the average squared distance between each example's label vector $\mathbf{l}_k$ and the set's mean label vector $\widehat{\mathbf{l}}$, i.e.,

$$\mathrm{Var}(\mathcal{E}) = \frac{1}{|\mathcal{E}|} \sum_{k \in \mathcal{E}} d(\mathbf{l}_k, \widehat{\mathbf{l}})^2 \tag{13}$$

The distance measure used in the above formula is the weighted Euclidean distance:

$$d(\mathbf{l}_1, \mathbf{l}_2) = \sqrt{\sum_{i=1}^{L} w(c_i)(l_{1,i} - l_{2,i})^2}, \quad (14)$$

where the class's weight $w(c_i)$ depends on its depth within the hierarchy. The similarities at higher levels in the hierarchy are considered more important than the similarities at lower levels. Therefore, the class weights $w(c_i)$ decrease with the depth of the class in the hierarchy. $w(c_i)$ is typically set as w_0^d, where d is the depth of the label in the hierarchy: w_0 was set to 0.75 in our experiments. The number of randomly selected features at each node was set to $\lfloor \sqrt{L} \rfloor + 1$, where L is the total number of features.

To combine the predictions of all classifiers in the ensemble and obtain the final prediction, their average is taken.

4. Results

In this section, we evaluate the performance of our proposed approach along two dimensions: its generalizability to handle unseen data and the effect of including/excluding features from separate modalities. Finally, the setup was experimentally verified in the robotic workcell.

4.1. Generalizability of Classification

In order to verify how well the approach can generalize to unseen data, we train a model on a subset where we do not include any examples of a particular outcome case. Since the database for the copper ring insertion task is not sufficiently fine-grained, this aspect was evaluated only for the peg-in-hole task. We excluded all cases where the positional error in any direction equals 5 mm and observed if the model could correctly predict the direction of error for the excluded examples. The results are shown in Figure 12. The model correctly predicted the direction of error for all the excluded examples, both at the first and the second level of hierarchy. As the predictions at the third hierarchical level describe the magnitude of error, they can also be evaluated using root mean square error (RMSE). At the third hierarchical level, it assigned all the excluded examples to the closest lower error class that was presented in training for the x direction ($\text{RMSE}_x = 1$ mm), whereas for the y direction it did so for 4 of the 6 examples (resulting in $\text{RMSE}_y = 1.91$ mm).

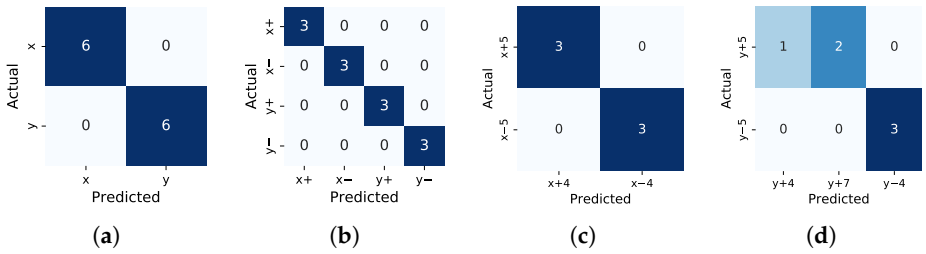


Figure 12. Confusion matrices for classification at different levels of the hierarchy. (**a**) First level of hierarchy (x or y displacement). (**b**) Second level (negative or positive displacement). (**c**) Third level (magnitude of x displacement); (**d**) Third level (magnitude of y displacement).

4.2. Single Modality versus Multimodal Models for Classification

We assessed the effectiveness of including/excluding features from the individual modalities by training multiple PCTs on different subsets of features for both tasks:

- only features based on the image data (see Section 3.6);
- only features based on the force-torque sensor data (see Section 3.4);
- features from both modalities.

Models were trained using 80% of the data and tested on the remaining 20% of the data.

The results for the PiH task are given in Figure 13. We found out that the model performed best when all features were included, indicating that the features from both modalities are complementary and improve the model's performance. The overall classification accuracy was calculated for the multi-class classification problem by taking the sum of the true positives and true negatives for each label, divided by the total number of predictions made. The accuracy was then averaged by support (the number of true instances for each label). For the model that uses all features, the overall classification accuracy at the first hierarchical level was 0.98. At the second level, the accuracy was 1.0. For the third level, the overall classification accuracy was 0.68. For error classes in the x and y directions, the classification accuracy was 0.55 and 0.5, respectively. For the model that only uses features from the vision modality, the overall classification accuracy at the first two hierarchical levels stayed the same, indicating that vision features can distinguish well between different types of outcomes. The overall classification accuracy at the third level was 0.68, and 0.5 and 0.55 for the x and y directions, respectively. When evaluating the model that only uses features from the FT modality, the overall classification accuracy at the first level dropped to 0.92, at the second to 0.95, and at the third to 0.61, whereas it was 0.5 and 0.35 for the x and y directions, respectively.

The results for the copper ring insertion task are given in Figure 14. Similar to before, we found that the model performed best when all features were included. For the model that uses all features, the overall classification accuracy at the first two hierarchical levels was 0.88. For the third level, the overall classification accuracy was 0.81. For error classes in x direction, the classification accuracy is 0.67, and 0.9 for y direction. For the model that only uses features from the vision modality, the overall classification accuracy at the first two hierarchical levels dropped slightly to 0.85. The overall classification accuracy at the third level was 0.62, and 0.5 and 0.6 for the x and y directions, respectively. The drop was even more pronounced when evaluating the model that only uses features from the FT modality. The overall classification accuracy at the first level was 0.81 and at the second and third it was 0.77, whereas it was 0.58 and 0.9 for the x and y directions, respectively. When comparing the results of the vision- and FT-features-only models, it is evident that while the earlier achieved a higher overall accuracy, the latter achieved higher accuracy when distinguishing among different magnitudes of error in the y direction.

4.3. Verification of Error Context Determination for the Generation of Exception Strategies

The proposed framework was experimentally verified on both the PiH and the copper ring insertion task. The initial PiH policy was carefully programmed and executed in the workcell with the same setup as described in Section 3.2. In order to cause an exception, the target position was displaced by 6 mm. The proposed approach correctly estimated the error context to "x/-/6". Since the exception strategy for this case has not yet been programmed, the robot stopped and prompted the operator. Using kinesthetic guidance, the operator guided the robot back along the policy to an appropriate point, where it is possible to resume the operation. The operator then demonstrated the correction, which resolved the problem. When we displaced the target position by 6 mm again, which resulted in a similar outcome, the robot again classified it as "x/-/6". As the exception strategy is now known, the robot could resolve the situation using the demonstrated exception strategy. In a similar manner, the operator demonstrated policies for the case where the target was displaced by 4 mm in a positive x direction. When we displaced the target position by 5 mm, the robot correctly classified the context to be "x/-", whereas the magnitude was not determined precisely (4 mm) as we used the model that did not include error contexts of this magnitude in the training set. Nevertheless, by combining the policies demonstrated for the other two cases in the "x/-" category and using locally weighted regression, as proposed in [6,43], the robot was able to perform the insertion successfully.

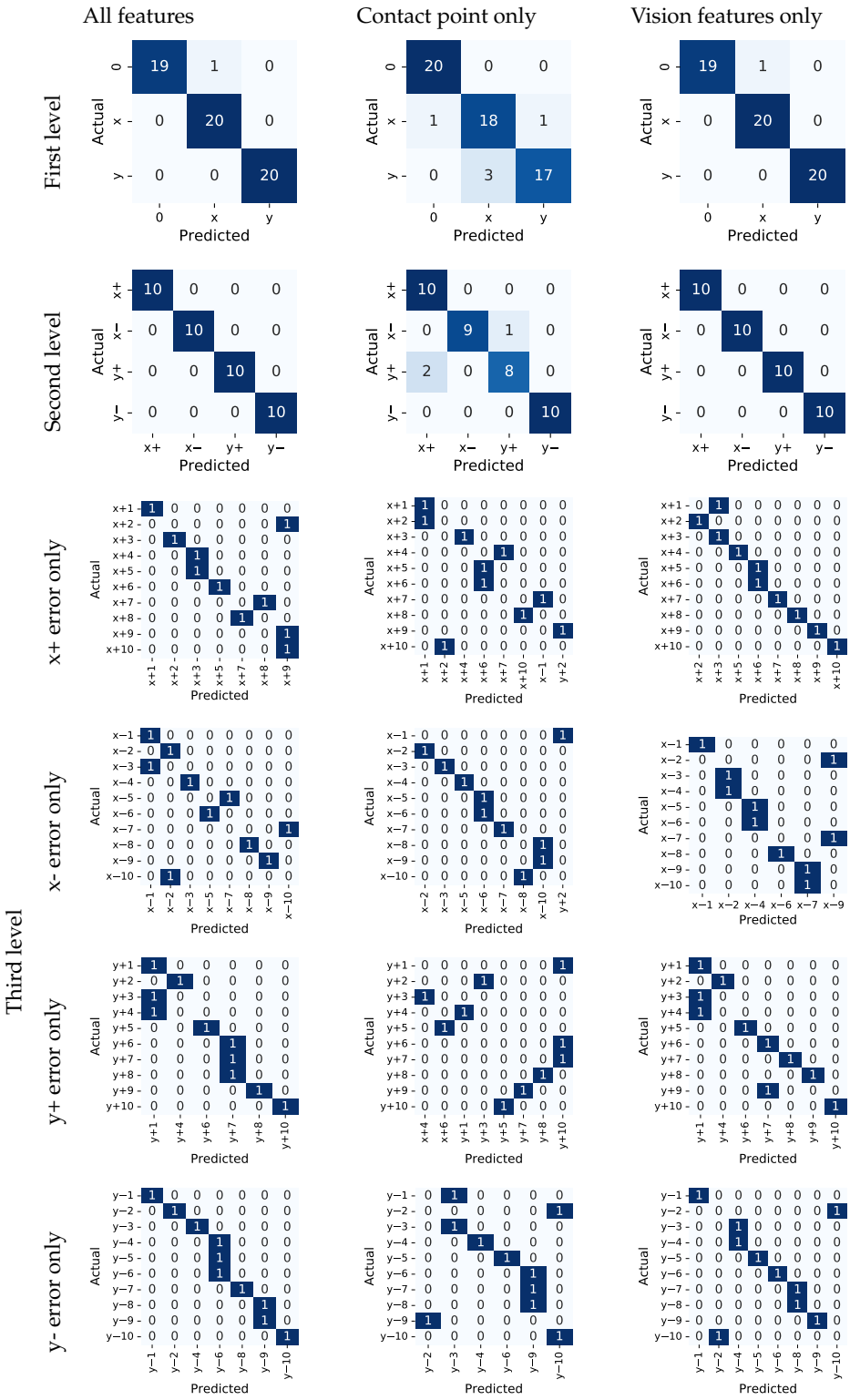


Figure 13. Confusion matrices for the PiH use case.

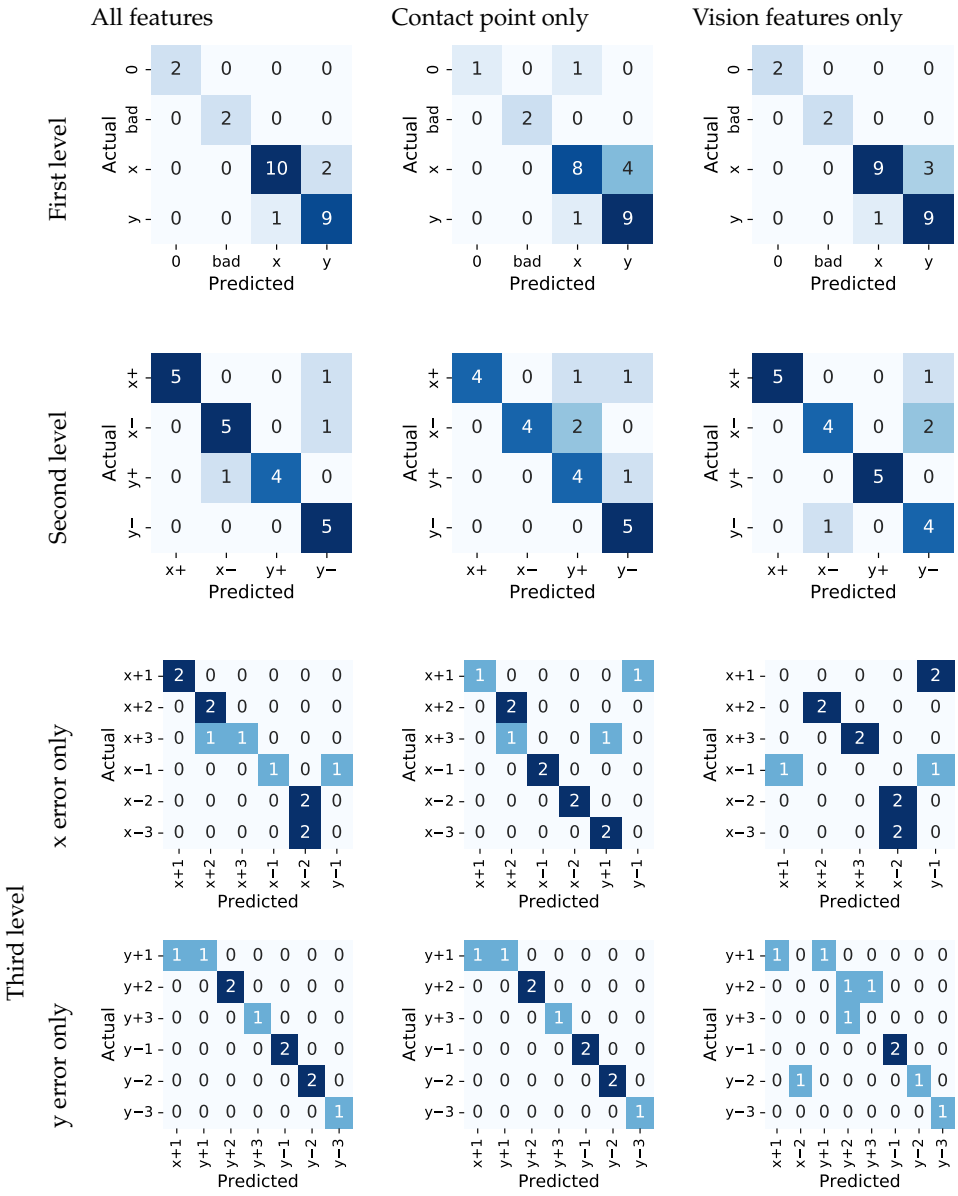


Figure 14. Confusion matrices for the copper ring insertion task.

When inserting a sliding ring into the casting mold, there are two major types of errors. The first type is when the base of the ring is not properly seated into the model (see Figure 5 middle). This type of error mainly arises due to imprecise grasping or due to errors in the target position. The second type of error occurs when the sliding ring is deformed. Both types of error can be reliably determined by using the proposed approach. We first displaced the target position by 2 mm in the negative y direction so that the insertion failed. As the exception strategy for this case has not yet been programmed, the operator demonstrated how to resume the operation and resolve the issue using iterative kinesthetic guidance [7]. When the target was displaced by the same offset again, the robot was able to resolve the problem. We also started the insertion procedure with a deformed part. It was correctly determined, and the robot placed it into the bin for deformed parts. An example video of both experiments can be found in the Supplementary Materials.

Note that the application of the exception strategy learning framework does not affect the cycle time in successful attempts. Once the model is deployed, the time to obtain context estimation is negligible. In unsuccessful attempts, where an alternate policy needs to be demonstrated or executed, the cycle time is, however, prolonged. However, since these situations are less frequent, this has very little effect on the average cycle time of an automated line.

5. Discussion

The results of our study indicate that the application of multimodal features leads to an improved classification accuracy of the ensemble models employed for classification. This implies that the features are complementary and taken together provide greater discrimination power than the features stemming from a single modality. Prediction errors that arose when applying vision-only-based models, showed the limitations of two-dimensional image data, thus depth information should be considered in the future.

It is important to note that, to a large extent, the models were able to correctly assign error types to examples with a magnitude of error not included in the training data. This is a critical finding as it indicates that the computed models are robust and can be used in real-world applications, also in less-structured non-industrial environments, where error types can not be predicted in advance. To evaluate this aspect, we excluded all examples with a certain magnitude of error from the training set. The results show that the models still perform well, indicating that they are not overfitting the training data.

Based on the observation that features obtained from different modalities contributed to improving the classification results at different levels, a more explicit hierarchical pipeline could be considered in the future, exploiting the robot as an agent that can interact with the environment. Data from different modalities would contribute towards the final prediction at different stages of the process, consisting of, e.g.,

(1) the type of error (due to positional displacement, part geometry, imprecise grasping), determined based on image data;
(2) the magnitude of error, based on force-torque or depth data.

The context determination does not have to occur instantaneously but can include exploring the environment as part of the pipeline. We could first use the vision data to determine the direction in which the robot should move in order to reduce the error (left/right). The robot can then move in this direction until it detects a new state (one of the force-torque components changes or a compliant robot stops moving as it hits an obstacle—see [55]). In the newly found state, the robot again estimates the direction in which it needs to continue or stop.

In the future, we intend to expand the proposed approach by considering other possible error types (e.g., arising from erroneous orientation when grasping) and their combinations (displacements in multiple degrees of freedom at the same time), as well as properly handling continuous data (regression at the lower hierarchical level instead of classification). We believe that the presented framework is not only applicable to learning error context but could also be extended to cognitive systems that will be able to respond autonomously to changes in the environment. To achieve these goals, the improved versions of our approach should consider additional modalities and alternative features extraction methods.

6. Conclusions

In this work, we have proposed a novel method for context determination based on multimodal features that can be used for learning exception strategies in various assembly tasks.

Our approach was validated on two tasks, the classic peg-in-hole, and the copper ring insertion. To evaluate its effectiveness, we deliberately induced different types of errors, which led to failed task executions. Using the proposed approach, the error type

was correctly obtained in all cases, allowing for correction of the task execution parameters and finally leading to successful task performance.

The study results indicate that the features used in the ensemble models are complementary and that the multimodal setup achieves the highest classification accuracy. Moreover, the model can correctly assign error types to examples with an unknown error magnitude.

In the current implementation, the context was calculated based on the measurement of forces and torques and RGB sensor data. The introduction of further sensor modalities (such as depth data) could lead to a further increase in classification accuracy.

Supplementary Materials: The following are available at https://www.mdpi.com/article/10.3390/s22207962/s1.

Author Contributions: Conceptualization, M.S. and B.N.; methodology, M.S., B.N. and S.D.; software, M.S., M.M.H. and S.D.; validation, M.S.; formal analysis, M.S. and A.U.; investigation, M.S.; resources, B.N. and A.U.; data curation, M.S. and M.M.H.; writing—original draft preparation, M.S.; writing—review and editing, M.S., S.D., A.U. and B.N.; visualization, M.S. and M.M.H.; supervision, B.N., A.U. and S.D.; funding acquisition, B.N. and A.U. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the European Union's Horizon 2020 through the ReconCycle project (contract no. 871352), the CoLLaboratE project (contract no. 820767), and by the Slovenian Research Agency (core research funding no. P2-0076).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data analyzed in this study are openly available in https://doi.org/10.5281/zenodo.7221443 (accessed on 13 October 2022) and https://doi.org/10.5281/zenodo.7221387 (accessed on 13 October 2022).

Acknowledgments: The authors would like to thank Rok Pahič for valuable discussions on extracting image features and Martin Breskvar for help with the CLUS setup in the early stages of this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. International Federation of Robotics. *World Robotics 2021 Industrial Robots*; VDMA Services GmbH: Frankfurt, Germany, 2021.
2. Roveda, L.; Magni, M.; Cantoni, M.; Piga, D.; Bucca, G. Human–robot collaboration in sensorless assembly task learning enhanced by uncertainties adaptation via Bayesian Optimization. *Robot. Auton. Syst.* **2021**, *136*, 103711. [CrossRef]
3. Gašpar, T.; Deniša, M.; Radanovič, P.; Ridge, B.; Savarimuthu, R.; Kramberger, A.; Priggemeyer, M.; Roßmann, J.; Wörgötter, F.; Ivanovska, T.; et al. Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells. *Robot. Comput.-Integr. Manuf.* **2020**, *66*, *101979*. [CrossRef]
4. Zhu, Z.; Hu, H. Robot Learning from Demonstration in Robotic Assembly: A Survey. *Robotics* **2018**, *7*, 17. [CrossRef]
5. Zachares, P.; Lee, M.A.; Lian, W.; Bohg, J. Interpreting Contact Interactions to Overcome Failure in Robot Assembly Tasks. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 3410–3417.
6. Nemec, B.; Simonič, M.; Ude, A. Learning of Exception Strategies in Assembly Tasks. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 6521–6527.
7. Simonič, M.; Petrič, T.; Ude, A.; Nemec, B. Analysis of Methods for Incremental Policy Refinement by Kinesthetic Guidance. *J. Intell. Robot. Syst.* **2021**, *102*, 5. [CrossRef]
8. Roheda, S.; Krim, H.; Riggan, B.S. Robust multi-modal sensor fusion: An adversarial approach. *IEEE Sens. J.* **2020**, *21*, 1885–1896. [CrossRef]
9. Polikar, R. Ensemble learning. In *Ensemble Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2012.
10. Kocev, D.; Vens, C.; Struyf, J.; Džeroski, S. Tree ensembles for predicting structured outputs. *Pattern Recognit.* **2013**, *46*, 817–833. [CrossRef]
11. Nemec, B.; Yasuda, K.; Mullennix, N.; Likar, N.; Ude, A. Learning by demonstration and adaptation of finishing operations using virtual mechanism approach. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 7219–7225.

12. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-Time Instance Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9156–9165.
13. Nottensteiner, K.; Sachtler, A.; Albu-Schäffer, A. Towards Autonomous Robotic Assembly: Using Combined Visual and Tactile Sensing for Adaptive Task Execution. *J. Intell. Robot. Syst.* **2021**, *101*, 49. [CrossRef]
14. Visinsky, M.; Cavallaro, J.; Walker, I. Robotic fault detection and fault tolerance: A survey. *Reliab. Eng. Syst. Saf.* **1994**, *46*, 139–158. [CrossRef]
15. Abu-Dakka, F.; Nemec, B.; Kramberger, A.; Buch, A.; Krüger, N.; Ude, A. Solving peg-in-hole tasks by human demonstration and exception strategies. *Ind. Robot. Int. J.* **2014**, *41*, 575–584. [CrossRef]
16. Marvel, J.A.; Bostelman, R.; Falco, J. Multi-Robot Assembly Strategies and Metrics. *ACM Comput. Surv.* **2018**, *51*, 1–32. [CrossRef] [PubMed]
17. Laursen, J.S.; Schultz, U.P.; Ellekilde, L.P. Automatic error recovery in robot assembly operations using reverse execution. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–3 October 2015; pp. 1785–1792.
18. Nicolescu, M.N.; Mataric, M.J. Learning and interacting in human-robot domains. *IEEE Trans. Syst. Man Cybern. Part A* **2001**, *31*, 419–430. [CrossRef]
19. Shetty, S.; Silverio, J.; Calinon, S. Ergodic Exploration Using Tensor Train: Applications in Insertion Tasks. *IEEE Trans. Robot.* **2021**, *38*, 906–921. [CrossRef]
20. Sigaud, O.; Droniou, A. Towards deep developmental learning. *IEEE Trans. Cogn. Dev. Syst.* **2016**, *8*, 99–114. [CrossRef]
21. Nemec, B.; Abu-Dakka, F.J.; Ridge, B.; Ude, A.; Jørgensen, J.A.; Savarimuthu, T.R.; Jouffroy, J.; Petersen, H.G.; Krüger, N. Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile. In Proceedings of the 2013 16th International Conference on Advanced Robotics (ICAR), Montevideo, Uruguay, 25–29 November 2013.
22. Vuga, R.; Nemec, B.; Ude, A. Speed adaptation for self-improvement of skills learned from user demonstrations. *Robotica* **2016**, *34*, 2806–2822. [CrossRef]
23. Diryag, A.; Mitić, M.; Miljković, Z. Neural networks for prediction of robot failures. *Proc. Inst. Mech. Eng. Part J. Mech. Eng. Sci.* **2013**, *228*, 1444–1458. [CrossRef]
24. Alonso-Tovar, J.; Saha, B.N.; Romero-Hdz, J.; Ortega, D. Bayesian Network Classifier with Efficient Statistical Time-Series Features for the Classification of Robot Execution Failures. *Int. J. Comput. Sci. Eng.* **2016**, *3*, 80–89. [CrossRef]
25. Bicchi, A.; Salisbury, J.K.; Brock, D.L. Contact sensing from force measurements. *Int. J. Robot. Res.* **1993**, *12*, 249–262. [CrossRef]
26. Kitagaki, K.; Ogasawara, T.; Suehiro, T. Contact state detection by force sensing for assembly tasks. In Proceedings of the 1994 IEEE International Conference on MFI'94, Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, NV, USA, 2–5 October 1994; pp. 366–370.
27. Liu, S.; Li, Y.F.; Xing, D. Sensing and control for simultaneous precision peg-in-hole assembly of multiple objects. *IEEE Trans. Autom. Sci. Eng.* **2019**, *17*, 310–324. [CrossRef]
28. JSI; IDIAP; CERTH. *D4.7—Assembly Policy Learning and Improvement*; Deliverable, CoLLaboratE Project; CoLLaboratE Project Consortium: Thessaloniki, Greece, 2020.
29. Sachtler, A.; Nottensteiner, K.; Kassecker, M.; Albu-Schäffer, A. Combined Visual and Touch-based Sensing for the Autonomous Registration of Objects with Circular Features. In Proceedings of the 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, 2–6 December 2019; pp. 426–433.
30. Nashman, M.; Nashman, M.; Hong, T.H.; Herman, M. *An Integrated Vision Touch-Probe System for Dimensional Inspection Tasks*; US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, 1995.
31. Allen, P.; Bajcsy, R. Robotic Object Recognition Using Vision and Touch. In Proceedings of the 9th International Joint Conference on Artificial Intelligence, Milan, Italy, 23–28 August 1987.
32. Thomas, U.; Molkenstruck, S.; Iser, R.; Wahl, F.M. Multi Sensor Fusion in Robot Assembly Using Particle Filters. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Roma, Italy, 10–14 April 2007; pp. 3837–3843. [CrossRef]
33. Bayoudh, K.; Knani, R.; Hamdaoui, F.; Mtibaa, A. A survey on deep multimodal learning for computer vision: Advances, trends, applications, and datasets. *Vis. Comput.* **2021**, *38*, 2939–2970. [CrossRef] [PubMed]
34. Boulahia, S.Y.; Amamra, A.; Madi, M.R.; Daikh, S. Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition. *Mach. Vis. Appl.* **2021**, *32*, 121. [CrossRef]
35. Lahat, D.; Adali, T.; Jutten, C. Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. *Proc. IEEE* **2015**, *103*, 1449–1477. [CrossRef]
36. Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; Ma, Q. A survey on ensemble learning. *Front. Comput. Sci.* **2020**, *14*, 241–258. [CrossRef]
37. Ren, Y.; Zhang, L.; Suganthan, P. Ensemble Classification and Regression—Recent Developments, Applications and Future Directions. *IEEE Comput. Intell. Mag.* **2016**, *11*, 41–53. [CrossRef]
38. Džeroski, S.; Panov, P.; Ženko, B. Ensemble Methods in Machine Learning. In *Encyclopedia of Complexity and Systems Science*; Meyers, R.A., Ed.; Springer: New York, NY, USA, 2009; pp. 5317–5325. [CrossRef]
39. Levatić, J.; Kocev, D.; Debeljak, M.; Džeroski, S. Community structure models are improved by exploiting taxonomic rank with predictive clustering trees. *Ecol. Model.* **2015**, *306*, 294–304. [CrossRef]
40. Levatić, J.; Ceci, M.; Kocev, D.; Džeroski, S. Semi-supervised Predictive Clustering Trees for (Hierarchical) Multi-label Classification. *arXiv* **2022**, arXiv:2207.09237. [CrossRef] .

41. Vens, C.; Struyf, J.; Schietgat, L.; Džeroski, S.; Blockeel, H. Decision trees for hierarchical multi-label classification. *Mach. Learn.* **2008**, *73*, 185–214. [CrossRef]
42. Collins, K.; Palmer, A.; Rathmill, K. The development of a European benchmark for the comparison of assembly robot programming systems. In *Robot Technology and Applications*; Springer: Berlin/Heidelberg, Germany, 1985; pp. 187–199.
43. Nemec, B.; Mavsar, M.; Simonič, M.; Hrovat, M.M.; Škrabar, J.; Ude, A. Integration of a reconfigurable robotic workcell for assembly operations in automotive industry. In Proceedings of the IEEE/SICE International Symposium on System Integration (SII), Virtual, 9–12 January 2022; pp. 778–783. [CrossRef]
44. Radanovič, P.; Jereb, J.; Kovač, I.; Ude, A. Design of a Modular Robotic Workcell Platform Enabled by Plug & Produce Connectors. In Proceedings of the 20th International Conference on Advanced Robotics (ICAR), Ljubljana, Slovenia, 6–10 December 2021.
45. Simonič, M.; Pahič, R.; Gašpar, T.; Abdolshah, S.; Haddadin, S.; Catalano, M.G.; Wörgötter, F.; Ude, A. Modular ROS-based software architecture for reconfigurable, Industry 4.0 compatible robotic workcells. In Proceedings of the 20th International Conference on Advanced Robotics (ICAR), Ljubljana, Slovenia, 6–10 December 2021.
46. Albu-Schaffer, A.; Ott, C.; Hirzinger, G. A passivity based Cartesian impedance controller for flexible joint robots—Part II: Full state feedback, impedance design and experiments. In Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA), New Orleans, LA, USA, 26 April–1 May 2004; Volume 3, pp. 2666–2672.
47. Jiang, J.; Huang, Z.; Bi, Z.; Ma, X.; Yu, G. State-of-the-Art control strategies for robotic PiH assembly. *Robot. Comput.-Integr. Manuf.* **2020**, *65*, 101894. [CrossRef]
48. Nemec, B.; Yasuda, K.; Ude, A. A virtual mechanism approach for exploiting functional redundancy in finishing operations. *IEEE Trans. Autom. Sci. Eng.* **2020**, *18*, 2048–2060. [CrossRef]
49. O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G.V.; Krpalkova, L.; Riordan, D.; Walsh, J. *Deep Learning vs. Traditional Computer Vision*; Advances in Computer Vision; Arai, K., Kapoor, S., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 128–144.
50. Lin, T.Y.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007.
51. Lin, T.Y.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
52. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
53. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
54. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
55. Simonič, M.; Žlajpah, L.; Ude, A.; Nemec, B. Autonomous Learning of Assembly Tasks from the Corresponding Disassembly Tasks. In Proceedings of the IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), Toronto, ON, Canada, 15–17 October 2019; pp. 230–236.