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:	31-03-2024

Deliverable Number:	D3.1
Deliverable Title:	A General Framework for Tactile Skill Definition in Robotic Disassembly
Type:	Report
Dissemination Level:	Public
Authors:	Kübra Karacan, Hamid Sadeghian, Saeed Abdolshah, Fan Wu, and Sami Haddadin
Contributing Partners:	TUM

Estimated Date of Delivery to the EC:	30-09-2021
Actual Date of Delivery to the EC:	04-11-2021

Contents

1 Executive summary

In this deliverable, a general framework is proposed for primitive taxonomy and tactile skill definition for ReconCycle. The taxonomy is established at the manipulation primitive level based on contact geometry, required sensitivity-precision, and the parameter space complexity. It is shown how various tactile skills can be achieved by blending three main primitives with appropriate parameters. The corresponding policies are first obtained remotely through a haptic console and the blending strategy is then applied for seamless integration of primitives into the desired skill. The proposed approach is verified experimentally for different skills such as unscrewing, levering, and cutting.

2 Introduction

Industrial robots currently perform well-defined tasks repetitively under highly constrained conditions. Therefore, the hardware and software of the robots are designed for tasks at hand and any change in product or process should be reintroduced to the settings of the robots. In other words, lack of robustness and flexibility results in that small batch production and unconstrained domains remain inaccessible to robotic automation.

Our work aims to develop a novel general framework for tactile skill definition and primitive taxonomy based on certain manipulation primitives. This framework introduces a high level of flexibility in the definition of different skills such as working with a screwdriver, levering, cutting, polishing, etc, which require precise force and motion control in contact with the environment.

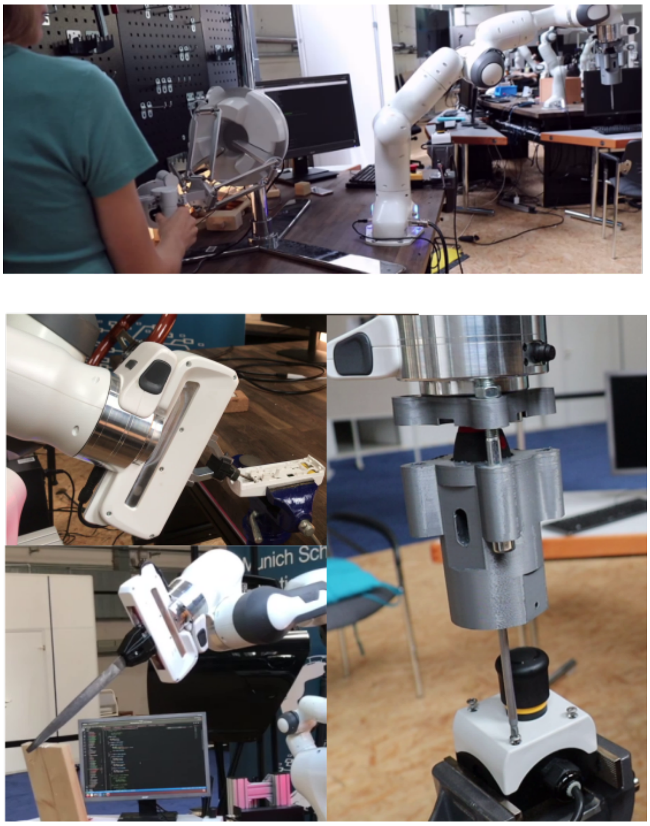

Figure 1: **Tactile skill definition and the use-cases.** Teaching the force/motion profiles for different skills such as unscrewing, levering, and cutting. The robot is commanded with a haptic interface in order to obtain corresponding motion and force profile which further are used to encode and control the policies.

3 Related Literature

Robotic tactile skills such as the ones shown in Fig. 1 require precise control of interaction at the end-effector. In other words, disassembly skills such as unscrewing, levering should contain a motion generation unit that fully incorporates end-effector (EEF) wrenches. There are different perspectives in the literature for combining motion generation and force control in robotic manipulators. For instance, Zielenski et al. [20] studied a method that classifies the manipulator behavior under three different phases: free-motion where the force is of no significance, exerting generalized forces, and lastly the transitions between the latter two behaviors. However, both position and force are controlled within a unified control system. Although force control has been considered in many works, encoding the desired force trajectory has not been in focus and the proposed controllers are validated based on constant force values or in form of thresholds/constraints [1, 4, 6, 11, 14].

Robot manipulators need to develop complex perceptuomotor skills, as they further tackle real-world problems [15]. In other words, they have to be robust enough to autonomously operate in unstructured environments and perform desired tasks despite any uncertainty in perception [16]. Therefore, defining tactile skills with position control is an unfeasible solution and other control methods should be considered instead.

The impedance control is a well-known approach that imposes a dynamic behavior between the external interaction and the desired motion, instead of tracking motion or force trajectory, independently [7]. These dynamics can be realized in the joint space, operational space, or even in the redundant space of a robot manipulator [17]. The impedance inertia, damping, and stiffness parameters are usually selected constant in each direction based on the assigned tasks. It can be easily shown that the impedance dynamics represent a passive mapping between the external force input and the motion of the robot as an output. This ensures that the system does not generate extra energy during interaction with the passive environment and thus it is intrinsically stable [13]. However, adaptive change of the impedance parameters seems beneficial in many applications to impose a more wise and human-like behavior for the external interaction on the EEF [19]. It can be easily shown that varying the impedance parameters will destroy the passivity of the closed-loop system and may lead to instability. To this end, the concept of an energy tank is exploited in [2, 3] to alter the dynamics of the closed-loop system to ensure passivity. To include compliant force tracking on the environment with unknown geometry, the WNN has effectively been used in [5]. The variable impedance approaches can also be used to coordinate the human and robot motion in applications with shared autonomy [2].

In this work, we covered the tactile skills by using taxonomy and the definition of only three manipulation primitives (establish contact, apply force, and break contact). By extending the current methods of motion generation for manipulation skills, unified force/motion trajectory is encoded and sent to the force/impedance controller. The primitives' parameters are obtained by using a remote teaching method. Moreover, a blending approach is proposed to combine primitives seamlessly and smoothly into the desired disassembly skill.

4 Methodology

Fig. 2 illustrates the proposed framework for the taxonomy of manipulation primitives and skill definition. In this framework, the taxonomy for tactile skills is applied at the primitive level based on the contact geometry, required sensitivity, precision, and parameter space complexity. The transition parameters (start and end conditions) between these primitives can be derived automatically from the scene perception through the exteroceptive sensor and external user interaction, or manually through predefined scenarios. The blending strategy is then exploited for seamless integration of the manipulation primitives to achieve the desired outcome.

Together with proprioceptive sensing, the primitive policy ($\dot{\boldsymbol{x}}_d$, $\boldsymbol{f}_d$) is characterized by a set of parameters based on the contact geometry, required sensitivity, precision, and parameter space complexity, which are subject to constraints from the scene interpretation. These parameters are initially

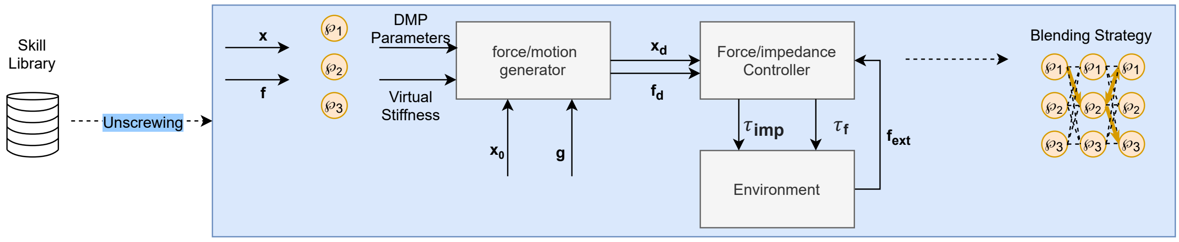

Figure 2: **Tactile skill definition.** Primitive taxonomy: establish the contact ($\mathcal{P}_1$), apply the force/motion profile ($\mathcal{P}_2$), and break the contact ($\mathcal{P}_3$). Teaching the primitives: the resultant force profile $\boldsymbol{f}$ and motion trajectory $\boldsymbol{x}$ are used for learning the parameters of the force/motion trajectory. The unified force/motion generator: sending its output to the force/impedance controller to establish and maintain the contact. Blending strategy: achieve the desired outcome.

determined during remote primitive teaching. Finally, the controller is responsible for the adaptation of the policies based on the purpose of the environment interaction. The multi-level approach for skill definition is employed to significantly reduce the complexity of manipulation for different interaction scenarios.

4.1 Primitive taxonomy and definition

To ensure compatibility with high-level desired objectives and low-level control behavior, any tactile skill is decomposed into three manipulation primitives:

- $\mathcal{P}_1$: establish the contact (making the initial alignment between the tool mounted at the EEF and the workpiece).
- $\mathcal{P}_2$: apply force/motion profile (to achieve the desired interaction in the task space).
- $\mathcal{P}_3$: break the contact (smoothly and safely stop the interaction).

4.2 Remote primitive teaching

The correct measurement of the contact force at the EEF has a significant effect on the generation of the corresponding policy when teaching tactile manipulation primitives. In ReconCycle we use a remote teaching strategy to obtain the actual force exerted by the robot without touching the robot. The setup in Fig. 1 shows an operator guiding the robot to achieve the desired motion and interaction with a haptic device (Sigma.7 from Force Dimension). The haptic device is used as a joystick and sends the desired velocities to the robot. Consequently, the resulting force profile $\boldsymbol{f}$ and motion trajectory $\boldsymbol{x}$ are recorded for policy generation.

4.3 Force/motion generator for the primitives

Each primitive $\mathcal{P}_i$ contributing to a desired skill has its own policy ($\dot{\boldsymbol{x}}_d$, $\boldsymbol{f}_d$) and, therefore, force-motion generation unit. The command, $\boldsymbol{\tau}_{\text{in}}$, is generated based on the desired motion and force profile ($\boldsymbol{x}_d$,$\boldsymbol{f}_d$), by using the general robot dynamics equation. Together with proprioceptive and exteroceptive sensing, the required policy can be adapted or learned based on encoding the unified force and motion trajectories.

The robot dynamics equation in Cartesian space can be expressed as follows,

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

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

where $\boldsymbol{M}_C(\boldsymbol{q}), \boldsymbol{C}_C(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{6\times 6}$ and $\boldsymbol{g}_C(\boldsymbol{q})$ are the robot mass matrix, Coriolis/centrifugal matrix and gravity vector in Cartesian space, respectively. Furthermore, $\boldsymbol{\tau}_{\text{in}} \in \mathbb{R}^n$ is the control input torque and $\boldsymbol{f}_{\text{ext}} \in \mathbb{R}^6$ the external wrench acting on the robot.

4.3.1 Encoding the motion trajectory

Even though it is possible to use directly cyclic and discrete motion generators, Dynamic Movement Primitives (DMPs) have been designed to represent any smooth robot motion [9]. A DMP controlling the motion of one degree of freedom denoted by x, is given by a second order differential equation system,

$$\lambda\dot{z} = \alpha_z(\beta_z(g - x) - z) + \gamma(s), \tag{3}$$

$$\lambda\dot{x} = z, \tag{4}$$

where g is the attractor point namely goal, and $s \in R$ is the phase variable and used to make the time dependency implicit. Its dynamics is given by

$$\lambda\dot{s} = -\alpha_s s. \tag{5}$$

Furthermore, the temporal scaling factor $\lambda > 0$, and constant parameters α_z, β_z, $\alpha_s > 0$, are selected such that the system has a unique attractor point at $y = g$. The nonlinear forcing term $\gamma(s)$ is as follows:

$$\gamma(s) = \frac{\sum_{i=1}^{N} \omega_i \psi_i(s)}{\sum_{i=1}^{N} \psi_i(s)} s(g - x_0), \tag{6}$$

$$\psi_i(s) = \exp\left(-\frac{1}{2\delta_i^2}(s - c_i)^2\right), \tag{7}$$

The forcing term γ allows us to design any point-to-point trajectory from the initial position x_0 to the desired goal position g by a linear combination of N radial-basis functions $\psi_i(s)$. c_i is the center of the functions and δ_i determines their widths. Additionally, $\alpha_z, \beta_z, \alpha_s, c_i$, and δ_i are generally designed to be constants, whereas time constant λ and the weights ω_i are determined from the recorded data. The generated trajectory (x_d, $\dot{x}_d$, and $\ddot{x}_d$) is finally obtained by Euler integration of (3), with the initial values set to $s = 1$, $x = x_0$, and $z = \lambda\dot{x} = 0$.

4.3.2 Encoding the force profile

Feed-forward force profile is especially useful to mitigate the chattering/jamming behavior of the robot while trying to maintain the contact force when the pose error is high due to friction, etc. To encode force from the recorded profiles during teaching, we used the study of Hogan [8]. In this approach, the force profile is encoded to be state-dependent and time-invariant in (9) based on zero-force trajectory, $\boldsymbol{x}_{\mathrm{zf}}$, (free motion). Therefore, generating force is defined to be responses to deviations between actual motion $\boldsymbol{x}_r$ and nominal motion $\boldsymbol{x}_{\mathrm{zf}}$

$$\boldsymbol{f} = \boldsymbol{\Omega}(\boldsymbol{x}_{\mathrm{zf}} - \boldsymbol{x}_r). \tag{8}$$

To create a virtual contact model and compute the virtual stiffness for the desired skill, $\boldsymbol{\Omega}$ is learnt from the recorded force profile for instance by Reinforcement Learning (RL):

$$\boldsymbol{f}_{\mathrm{d}} = \begin{cases} \exists\, \boldsymbol{\Omega}\, \forall\, \boldsymbol{x}_d \mid \boldsymbol{f}_{\mathrm{d}} = \boldsymbol{\Omega}(\boldsymbol{x}_{\mathrm{zf}} - \boldsymbol{x}_{\mathrm{d}}) & \text{if } \boldsymbol{f}_{ext}^T \tilde{\boldsymbol{x}} < 0 \\ & \quad (contact), \\ 0 & \text{if } \boldsymbol{f}_{ext}^T \tilde{\boldsymbol{x}} \geq 0 \\ & \quad (no\ contact), \end{cases} \tag{9}$$

4.4 Control Design

The proposed law is designed as a force-impedance controller, which has three main tasks: i) to track the desired motion $\boldsymbol{x}_d$, ii) to establish and regulate the desired force $\boldsymbol{f}_d$, and iii) to compensate the robot gravity $\boldsymbol{\tau}_g$ [18]. The input torque $\boldsymbol{\tau}_{\mathrm{in}}$ is specified as follows:

$$\boldsymbol{\tau}_{\mathrm{in}} = \boldsymbol{\tau}_{\mathrm{imp}} + \boldsymbol{\tau}_{\mathrm{frc}} + \boldsymbol{\tau}_g, \tag{10}$$

where $\boldsymbol{\tau}_{\mathrm{imp}}$ and $\boldsymbol{\tau}_{\mathrm{frc}} \in \mathbb{R}^n$ are the input torques for the motion and force control.

4.4.1 Force/impedance control

In order to introduce an impedance behavior to the end-effector, the following controller is used:

$$\boldsymbol{\tau}_{\mathrm{imp}} = \boldsymbol{J}^T(\boldsymbol{q})(\boldsymbol{K}_C \tilde{\boldsymbol{x}} + \boldsymbol{D}_C \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{M}_C(\boldsymbol{q})\ddot{\boldsymbol{x}}_d + \boldsymbol{C}_C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{x}}_d), \tag{11}$$

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}, \tag{12}$$

To establish and regulate the desired contact force $\boldsymbol{f}_d$ at the end-effector according to the external force, $\boldsymbol{f}_{\mathrm{ext}} \in \mathbb{R}$, the force control is defined as follows:

$$\boldsymbol{\tau}_{\mathrm{frc}} = \boldsymbol{J}^T(\boldsymbol{q}) \underbrace{[\boldsymbol{R}, \boldsymbol{0}_{3\times 3}]^T\, [0, 0, \rho_{\mathrm{frc}}\, f_{\mathrm{frc}}]^T}_{\boldsymbol{f}_{\mathrm{frc}}}, \tag{13}$$

$$\boldsymbol{f}_{\mathrm{frc}} = \boldsymbol{f}_d + k_p\, \tilde{\boldsymbol{f}}_{\mathrm{ext}} + k_i \int \tilde{\boldsymbol{f}}_{\mathrm{ext}}\, dt + k_d\, \dot{\tilde{\boldsymbol{f}}}_{\mathrm{ext}}, \tag{14}$$

$$\tilde{\boldsymbol{f}}_{\mathrm{ext}} = \boldsymbol{f}_d - \boldsymbol{f}_{\mathrm{ext}}, \tag{15}$$

where $\boldsymbol{f}_{\mathrm{frc}} \in \mathbb{R}^6$ is the feedback + feedforward force term and k_p, k_i, k_d are the PID gains. The input shaping function ρ_{frc} in (13) deactivates the force controller when the robot deviates from the set-point of the end-effector. The idea behind the shaping function is to avoid undesired motions, especially when contact loss happens. Therefore, ρ_{frc} is designed as follows:

$$\rho_{\mathrm{frc}} = \begin{cases} 1 & \text{if } \boldsymbol{f}_d^T \tilde{\boldsymbol{x}} > 0, \\ 0.5(1 - \cos(\pi(\frac{\tilde{x}_z}{\delta_{\mathrm{frc}}} - 1))) & \text{if } \boldsymbol{f}_d^T \tilde{\boldsymbol{x}} < 0 \text{ and } 0 \leq \tilde{x}_z \leq \delta_{frc}, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Using a regular hybrid force-position controller instead of the force-impedance controller may cause incompatibility between the directions of the desired motion and force. For instance, when the desired motion is in the opposite direction to the force, either force or motion controller would be deactivated. Hence we address this issue by introducing the respective threshold $\delta_{\text{frc}} > 0$ to the shaping function ρ_{frc}, which enables one to control the force in the desired region.

4.5 From Primitives to Skills: Blending Strategy

To achieve robust and flexible behaviors and thereby avoid longer running times, the primitives are integrated by a blending strategy [10], [12]. We developed a blending approach that seamlessly and smoothly combines the primitives into the desired skill. The proposed skill definition enables us to program complex skills rather easily and in part autonomously. Algorithm 1 illustrates the proposed approach for an example scenario of unscrewing skill.

Algorithm 1 Skill definition for unscrewing

procedure UNSCREWING, ξ^c
  parameter server, $\theta \leftarrow \boldsymbol{x}_0, \boldsymbol{g}, \boldsymbol{f}_d, \rho_{frc}$
  establish contact, $\mathcal{P}\ \{\dot{\boldsymbol{x}}_d, \boldsymbol{f}_d\}_1 \leftarrow \boldsymbol{g}$
  apply force, $\mathcal{P}\ \{\dot{\boldsymbol{x}}_d, \boldsymbol{f}_d\}_2 \leftarrow \boldsymbol{x}_0, \boldsymbol{g}, \boldsymbol{f}_d, \rho_{frc}$
  break contact, $\mathcal{P}\ \{\dot{\boldsymbol{x}}_d, \boldsymbol{f}_d\}_3 \leftarrow \boldsymbol{g}$
  loop, $\xi_\theta(d_t)$:
  establish contact:
  if *arrived at the desired pose* $\boldsymbol{g} = \boldsymbol{x}_r$ **then return** the primitive as successful $(\mu = 1)$, update the initial pose $(\boldsymbol{x}_0)$ with the current pose $(\boldsymbol{x}_r)$, and *apply force*
  apply force:
  if *contact loss* **then return** repeat the primitive $(\mu = 0)$, update the goal $\boldsymbol{g}$ with the last pose $\boldsymbol{x}_r$ before contact loss $(\boldsymbol{g} = \boldsymbol{x}_r \mid \boldsymbol{f}_{ext} \neq 0)$, and *establish contact*
  if *achieved the desired motion and force profile* **then return** the primitive as successful $(\mu = 1)$, update the initial pose $(\boldsymbol{x}_0)$ with the current pose $(\boldsymbol{x}_r)$, and deactivate the force controller $(\rho_{frc} \mid \boldsymbol{f}_d^T \tilde{\boldsymbol{x}} < 0,\ 0 \leq \tilde{x}_z \leq \delta_{frc})$ *break contact*
  break contact:
  if *arrived at the desired pose* (g) **then return** the primitive as successful $(\mu = 1)$

The blending strategy is specified in Alg. 1, which allows one to achieve the overall control policy (ξ^c) by autonomously calculating a blending of each primitive policy $(\mathcal{P}_m)$ based on environment state, d_t. The control policy ξ^c is defined by the blending strategy and the primitive policies as follows:

$$\xi^c = \sum_{m=1}^{3} \mu_m \mathcal{P}\{\dot{\boldsymbol{x}}_d, \boldsymbol{f}_d\}_m, \tag{17}$$

$$d_t = \{\boldsymbol{x}_r, \dot{\boldsymbol{x}}_r, \ddot{\boldsymbol{x}}_r, \boldsymbol{f}_{ext}\}, \tag{18}$$

$$\theta = \{\boldsymbol{x}_0, \boldsymbol{g}, \boldsymbol{f}_d, \rho_{frc}\}, \tag{19}$$

where θ is a set of parameters for the state dependent blending strategy and state d_t is the sensor information at time step t

$$
\begin{cases} \mu = 1 \wedge \boldsymbol{x}_0 = \boldsymbol{x}_r & \text{if } \boldsymbol{f}_{ext}^T \tilde{\boldsymbol{x}} \leq 0 \wedge \; \boldsymbol{g} = \boldsymbol{x}_r, \\ \mu = 0 \wedge \boldsymbol{g} = \boldsymbol{x}_r \mid \boldsymbol{f}_{ext} \neq 0 & \text{if } \boldsymbol{f}_{ext}^T \tilde{\boldsymbol{x}} > 0, \end{cases} . \tag{20}
$$

Any tactile disassembly skill such as unscrewing is defined by combining the above primitives (establish contact, apply force, and break contact). Each manipulation primitive $\mathcal{P}_i\{\dot{\boldsymbol{x}}_d, \boldsymbol{f}_d\}_m$ is comprised of a DMP, which is a time-variant policy, and a time-invariant and state-dependent force profile. However, the blending and combination strategy can be time- and state-dependent. Furthermore, the DMP equation in (3) has an initial position, $\boldsymbol{x}_0$, and goal, $\boldsymbol{g}$, which vary between the primitive skills and depend on the blending policy. In order to solve the dependency problems in (17), the initial position $\boldsymbol{x}_0$ and goal $\boldsymbol{g}$ are updated and combined according to the blending strategy. The idea behind this operation is that when one primitive executes, the canonical system, s, of this primitive policy is updated to generate the motion $\boldsymbol{x}_d$ and force trajectory $\boldsymbol{f}_d$ of this policy. Next it is decided if the same primitive should be repeated $(\mu = 0)$ or the execution is marked as successful $(\mu = 1)$.

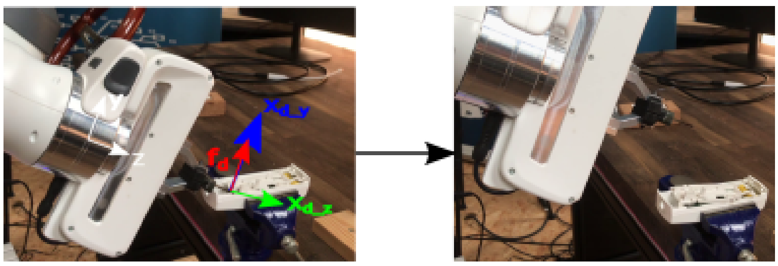

Figure 3: **Implementation of the framework for a recycling scenario.** Removing the batteries out of a heat cost allocator fixed with various inclination angles.

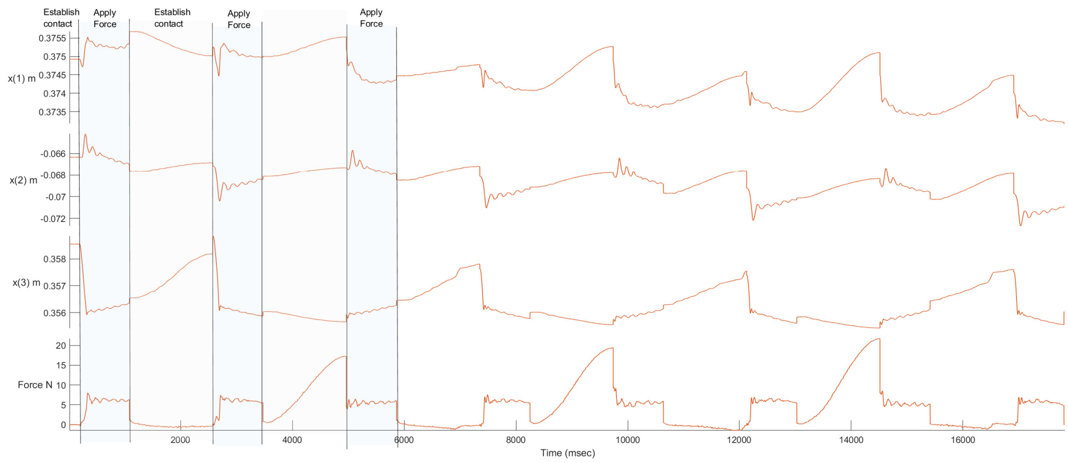

Figure 4: **Results of the tactile skill definition by using unified force-motion profile, the proposed primitives: establish contact, apply force, and break contact, and the blending strategy for learning a real-life disassembly skill: unscrewing.** The first three rows show the resulting motion in task space $(x(1), x(2), x(3))$, whereas the bottom one presents the coupled force profile of the end-effector $(Force)$. Force profile represents unscrewing repetitively and robustly with the contact force of ≈5N.

5 Results and Discussion

To prove the performance of our framework with real robots, we implemented some daily life disassembly skills such as unscrewing and levering (in different inclinations). They were performed with a Franka Emika Panda robot as shown in Fig. 3. First, the results for unscrewing in Fig. 4 show that contact between the screwdriver and the screw was successfully established and robustly maintained until the desired conditions were reached. In parallel, the desired motion was tracked smoothly especially during the transition from one primitive to another. Even though the contact was disturbed and, thus, broken, the robot safely stopped applying the force and re-established the desired contact to achieve the goal of the skill.

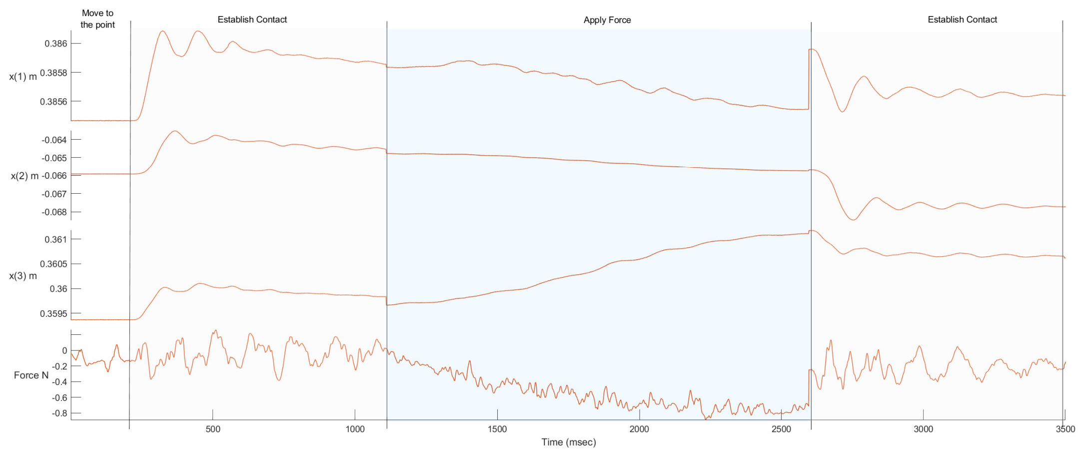

Figure 5: **Results for unscrewing only by using the motion without feed-forward force term.** First three rows show the resulting motion in task space $(x(1), x(2), x(3))$, whereas the bottom one presents the external force at the end-effector $(Force)$. Force around zero newton means that the contact failed to be established.

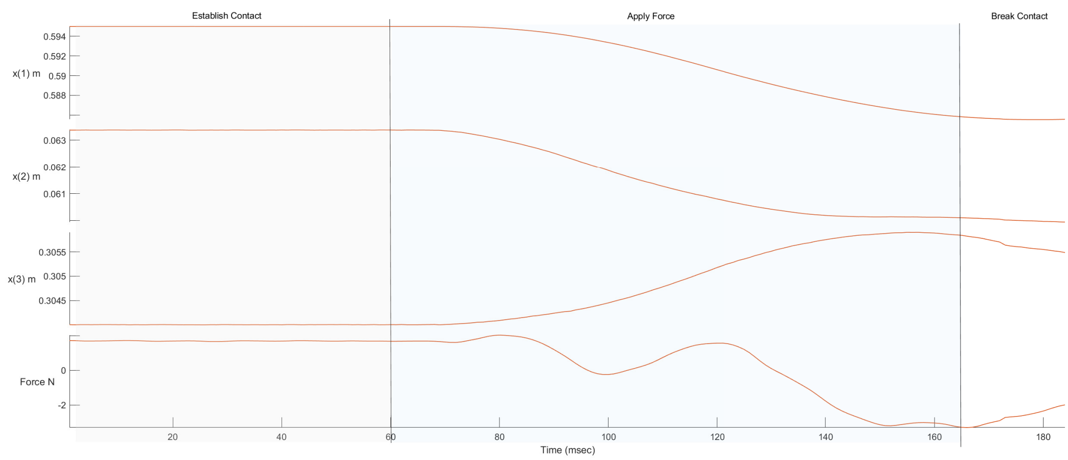

Figure 6: **Results for levering by using establish contact, apply force, and break contact and the blending strategy on the lid of a heat cost allocator.** First three rows show the resulting motion in task space $(x(1), x(2), x(3))$, whereas the bottom one presents the coupled force profile of the end-effector $(Force)$.

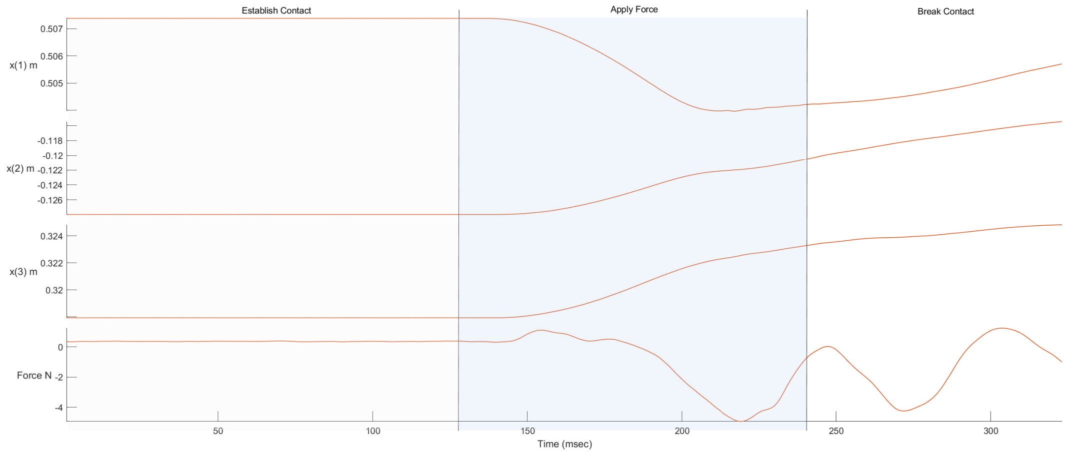

Figure 7: **Results for levering fixed on an inclined surface by using the proposed primitives establish contact, apply force, and break contact and the blending strategy on the lid of a heat cost allocator.** The first three rows show the resulting motion in task space ($x(1), x(2), x(3)$), whereas the bottom one presents the coupled force profile of the end-effector ($Force$).

To analyze the comparability of our novel framework to the broadly used manipulation methods, we have also conducted the same experiments by using impedance control and the motion generation with an implicit force profile. The results in Fig. 5 present that without feed-forward force term, maintaining contact and therefore applying the desired force requires additional tuning and is easy to be disturbed.

Additionally, the adaptability of our framework without changing any parameters was analyzed under two scenarios: levering the lid of a heat cost allocator fixed on flat and inclined surfaces. The results in Fig. 6 and Fig. 7 show that the force-motion trajectory was successfully adapted and the robot achieved its goal by executing the desired skill using the proposed primitives with smooth transitions.

6 Conclusion

In this deliverable, we described and experimentally validated a complete framework for primitive taxonomy and tactile skill definition to solve a class of disassembly tasks. We developed the framework by providing taxonomy at the manipulation primitive level based on contact geometry, required sensitivity, precision, and parameter space complexity: establish contact, apply force, and break contact. To measure the correct parameters, we include remote primitive teaching, which enables us to encode and generate the corresponding policies. The blending strategy is then exploited for the seamless integration of manipulation primitives in order to implement the desired skills such as unscrewing. We achieved an increased robustness and flexibility in comparison to the state-of-the-art methods and proved their operation with a real robot. In future work, we will extend our framework by incorporating human interaction at the primitive level. A blending strategy will be designed such that a human operator can collaborate with the robot during disassembly tasks.

References

[1] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse. "Collaborative manufacturing with physical human–robot interaction". In: *Robotics and Computer-Integrated Manufacturing* 40 (2016), pp. 1–13. ISSN: 0736-5845.

[2] F. Ferraguti, C. T. Landi, L. Sabattini, M. Bonfè, C. Fantuzzi, and C. Secchi. "A variable admittance control strategy for stable physical human-robot interaction". In: *The International Journal of Robotics Research* 38.6 (2019), pp. 747–765. DOI: 10.1177/0278364919840415.

[3] F. Ferraguti, C. Secchi, and C. Fantuzzi. "A tank-based approach to impedance control with variable stiffness". In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 4948–4953. DOI: 10.1109/ICRA.2013.6631284.

[4] F. Ficuciello, L. Villani, and B. Siciliano. "Variable Impedance Control of Redundant Manipulators for Intuitive Human–Robot Physical Interaction". In: *IEEE Transactions on Robotics* 31.4 (2015), pp. 850–863. ISSN: 1941-0468. DOI: 10.1109/TRO.2015.2430053.

[5] M. H. Hamedani, H. Sadeghian, M. Zekri, F. Sheikholeslam, and M. Keshmiri. "Intelligent Impedance Control using Wavelet Neural Network for dynamic contact force tracking in unknown varying environments". In: *Control Engineering Practice* 113 (2021), p. 104840. ISSN: 0967-0661. URL: https://www.sciencedirect.com/science/article/pii/S0967066121001179.

[6] W. He, Y. Chen, and Z. Yin. "Adaptive Neural Network Control of an Uncertain Robot With Full-State Constraints". In: *IEEE Transactions on Cybernetics* 46.3 (2016), pp. 620–629. ISSN: 2168-2275. DOI: 10.1109/TCYB.2015.2411285.

[7] N. Hogan. "Impedance Control: An Approach to Manipulation". In: *1984 American Control Conference*. 1984, pp. 304–313. DOI: 10.23919/ACC.1984.4788393.

[8] N. Hogan. *Physical interaction via dynamic primitives*. Vol. 117. 2017, pp. 269–299. ISBN: 9783319515472. DOI: 10.1007/978-3-319-51547-2_12.

[9] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. "Dynamical movement primitives: Learning attractor models formotor behaviors". In: *Neural Computation* 25.2 (2013), pp. 328–373. ISSN: 08997667. DOI: 10.1162/NECO_a_00393.

[10] L. Johannsmeier, M. Gerchow, and S. Haddadin. "A framework for robot manipulation: Skill formalism, meta learning and adaptive control". In: *Proceedings - IEEE International Conference on Robotics and Automation* 2019-May (2019), pp. 5844–5850. ISSN: 10504729. arXiv: 1805.08576.

[11] 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)*. 2015, pp. 102–105. DOI: 10.1109/SCP.2015.7342075.

[12] T. Narita and O. Kroemer. "Policy Blending and Recombination for Multimodal Contact-Rich Tasks". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2721–2728. DOI: 10.1109/LRA.2021.3061982.

[13] C. Ott. *Cartesian impedance control of redundant and flexible-joint robots*. Springer, 2008.

[14] C. Ott, A. Dietrich, and A. Albu-Schäffer. "Prioritized multi-task compliance control of redundant manipulators". In: *Automatica* 53 (2015), pp. 416–423. ISSN: 0005-1098.

[15] 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*. 2011, pp. 3828–3834. DOI: 10.1109/ICRA.2011.5980200.

[16] 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)*. 2012, pp. 309–315. DOI: 10.1109/HUMANOIDS.2012.6651537.

[17] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano. “Task-space control of robot manipulators with null-space compliance”. In: *IEEE Transactions on Robotics* 30.2 (2013), pp. 493–506.

[18] C. Schindlbeck and S. Haddadin. “Unified Passivity-Based Cartesian Force / Impedance Control for Rigid and Flexible Joint Robots via Task-Energy Tanks”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), pp. 440–447. DOI: 10.1109/ICRA.2015.7139036.

[19] 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”. In: *IEEE transactions on robotics* 27.5 (2011), pp. 918–930.

[20] C. Zieliñski and T. Winiarski. “Motion Generation in the MRROC++ Robot Programming Framework”. In: *The International Journal of Robotics Research* 29.4 (2010), pp. 386–413.