

This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: https://dx.doi.org/10.1007/978-3-031-32606-6_49 . Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use <https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms>

Link from the accepted manuscript version to the URL of the published article on the journal's website: https://link.springer.com/chapter/10.1007/978-3-031-32606-6_49

Adaptive Robotic Levering for Recycling Tasks

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

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

² Jožef Stefan International Postgraduate School, Ljubljana, Slovenia

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

Keywords: Robotic disassembly · Levering · Force control.

1 Introduction

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

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

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

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

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

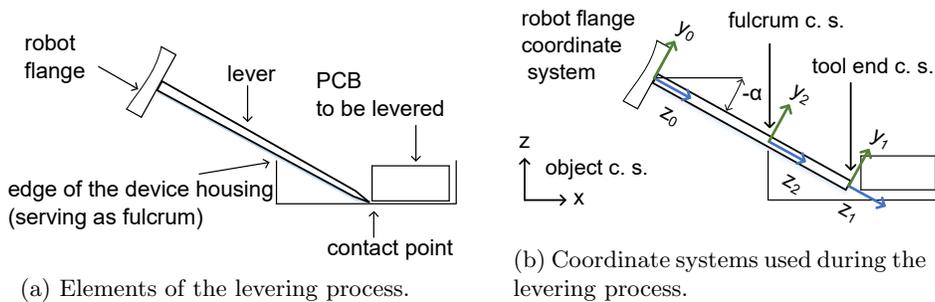
2 Methodology

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

2.1 Levering setup

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

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



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

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

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

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

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

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

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

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

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

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

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

$$\quad (10)$$

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

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

$$\quad (13)$$

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

$$\quad (15)$$

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

$$\quad (17)$$

$$\quad (18)$$

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

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

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

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

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

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

2.2 Fulcrum and part contact point search algorithm

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

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

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

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

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

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

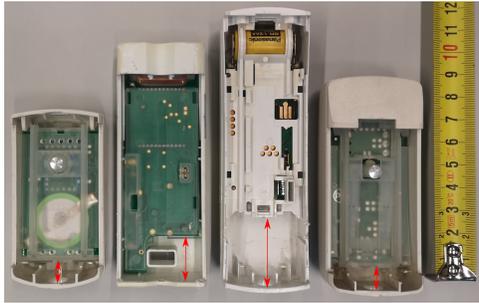


Fig. 2: Example electronic devices that need to be dismantled. Levering is used to remove the PCB from the housing.

2.3 Adaptive leveringing algorithm

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

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

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

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

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

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

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

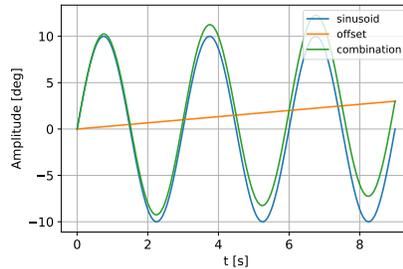


Fig. 3: Three periods of a sinusoid with an increasing offset g .

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

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

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

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

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

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

We can use force-torque measurements $[F_{x,0}, F_{y,0}, F_{z,0}, M_{x,0}, M_{y,0}, M_{z,0}]^\top$ at the robot’s flange to detect when the levering action succeeded at prying out the PCB. To determine when the PCB (part) has been pried out, we monitor the torque $M_{x,0}$ measured at the robot’s flange. During levering the torque increases, while a sharp torque drop-off is observed at the moment when the PCB is dislodged. The levering is successful if the difference between the maximum and minimum torque value within the width of the signal observation window is greater than a prespecified threshold M_{max}

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

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

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

2.4 Levering after adaptation

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

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

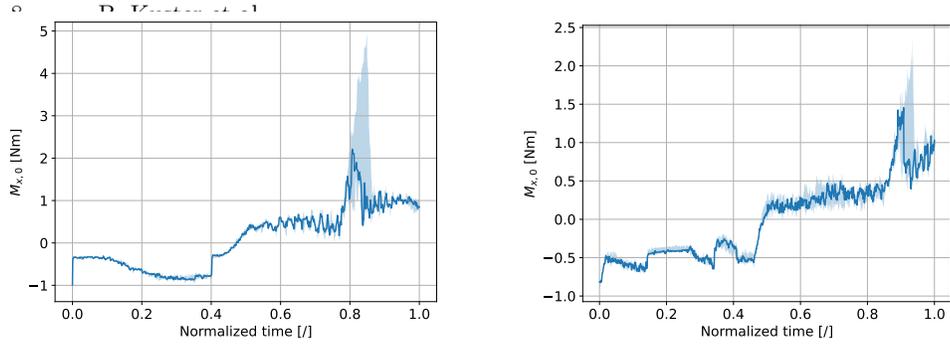
To speed up the levering process, we record the initial position at the start of the levering action and the robot pose reached after establishing the contact of the tool with the levered part (in our case, the PCB) and the fulcrum (edge of the device housing), both in object c.s. The highest amplitude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

3 Experimental evaluation

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

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

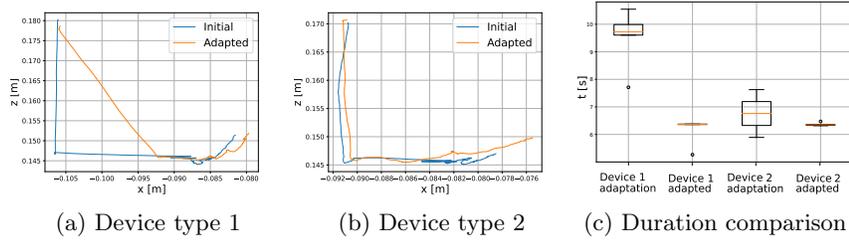
A comparison of the adaptation (initial) trajectory and the adapted trajectory is shown in Figs. 5a and 5b. Instead of searching for contact, the robot immediately moves to the previously learned contact point. Fig. 5c shows the comparison of average execution duration, as well as the standard deviation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



(a) Avg. torque and the standard deviation during 5 experiments - device type 1

(b) Avg. torque and the standard deviation during 5 experiments - device type 2

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



(a) Device type 1

(b) Device type 2

(c) Duration comparison

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

4 Conclusion and further work

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

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

References

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