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:	D5.8
Deliverable Title:	Benchmarks and KPIs
Type:	Report
Dissemination Level:	Public
Authors:	Hamid Sagedhian, Kübra Karacan, Mihael Simonič, Boris Kuster, Matija Mavsar, Sebastian Ruiz, Manuel G. Catalano, Riccardo Persichini, Aleš Ude, and Jürgen Schulz
Contributing Partners:	All

Estimated Date of Delivery to the EC:	31-07-2024
Actual Date of Delivery to the EC:	11-09-2024

Contents

1 Executive summary

This deliverable reports the key performance indicators (KPIs) developed for robotic operations and processes dismantling electronic devices, considering the latest relevant standards. We successfully adapted and applied ISO 22400 for KPIs in manufacturing operations management and MRED (Multi-Relationship Evaluation Design) developed by NIST (National Institute of Standards and Technology) to the recycling domain. Based on these standards, we derived specific KPIs for different use cases and established their target values. The integrated use cases were implemented and tested. The project's performance was thoroughly assessed against the KPIs outlined in Task 5.3. The results demonstrate that the project successfully met the established KPIs, with all targets achieved. Furthermore, recommendations for enhancing the work were provided where necessary, ensuring that the project met its objectives and paved the way for potential future improvements.

2 Introduction and Background

The diversity and model variety of electronic waste necessitates more adaptable methods for recycling, driving the demand for innovative automation concepts. Despite these needs, electronic waste recycling remains largely unautomated due to high setup costs, lengthy setup times, and lack of adaptive flexibility. The primary goal of the ReconCycle project is to introduce this flexibility into a cost-effective robot workcell concept, enabling automated robotic disassembly of various electronic devices.

Performance benchmarks are typically structured as modular sets of task-based tests. For example, the NIST task boards are designed for assembly operations (e.g., simple insertions, threading, snap-fitting, meshing, routing), as shown in Fig.1 and Fig.2. These benchmarks focus on assembly tasks by analyzing them through measurement science to identify key metrics and potential test methods. Reconfigurable tests incorporate various assembly tasks using standard components like screws, nuts, washers, gears, and electrical connectors, drawing from factors identified by Boothroyd-Dewhurst (B-D) design for assembly (DFA) studies.

However, while these NIST task boards provide a robust framework for assessing assembly operations, they fall short when applied to disassembly operations, which present unique challenges not adequately covered by existing benchmarks. Disassembly involves processes inherently more complex and less predictable than assembly, such as dealing with the variability and degradation of materials, safely handling hazardous components, and efficiently separating mixed materials. These tasks require more extensive experimental setups and specialized metrics beyond what is provided by the NIST boards or any other standard assembly-focused benchmarking tools [9].

For example, while assembly tasks often focus on precision and repeatability, disassembly must account for the disorderly and often destructive nature of breaking down devices. These processes can vary widely depending on the condition and design of the electronic waste. This variability requires a benchmarking system that can handle various scenarios, including dynamically adapting to different disassembly processes. Consequently, the ReconCycle project emphasizes the development of tailored benchmarks and experimental setups that specifically address the complexities of robotic disassembly, ensuring that the KPIs and performance metrics are comprehensive and relevant to the unique demands of this field.

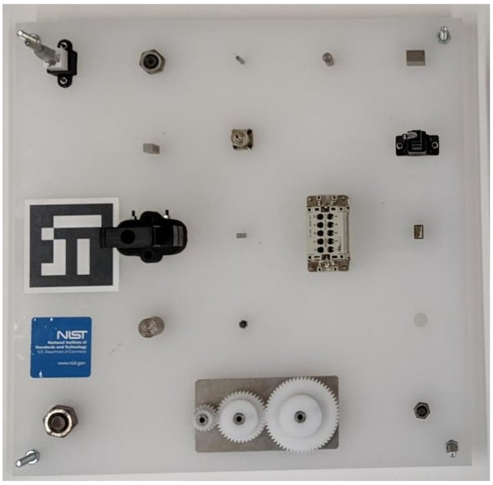


Figure 1: NIST assembly task board.

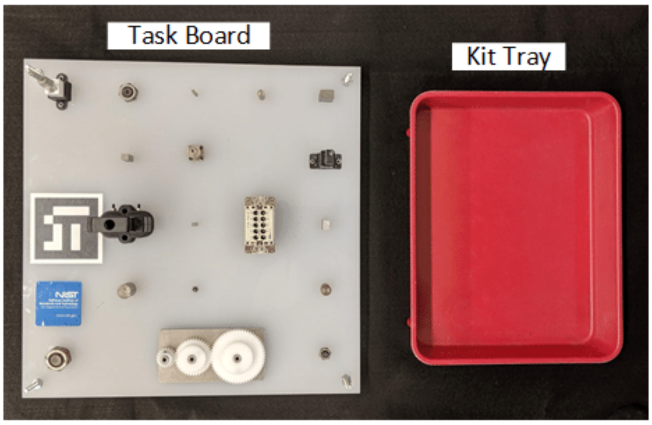


Figure 2: An example of disassembly task board proposed in [7].

3 Reconcycle KPIs and use-cases

The robotic work cell for dismantling the electronic devices, which was developed in the ReconCycle project, was benchmarked by measuring:

- The success rates of action choices and parameterizations made by the ReconCycle system in specific numbers of different exemplars from different device types and available actions arising from the use cases.
- The number of suitable actions planned and executed by the system to recover from specific failures.
- The number of suitable actions generated and executed for situations distinct, in specific ways, from those the system had encountered while creating the archetypical solution.
- The execution times of the individual steps required to address the use cases. Here, we quantify the work cell's performance relative to human performance.

Furthermore, we grouped our KPIs based on the work packages in the project.

3.1 KPIs in WP1 for Hardware and Software Setup and Dismantling Processes

The target KPIs for these processes are:

- Robustness and reliability of inserting electronic devices into the work-holding devices: over 95%
- Robustness and reliability of battery extraction from different electronic devices: over 90%.
- Cycle time of the dismantling process of a device involving self-reconfiguration or process adaptation < 90 seconds.
- Time of hardware reconfiguration and software changes when switching the dismantling process from one known device to another known device within the same family of devices: less than 2 minutes.
- Time of hardware reconfiguration and software changes when switching the dismantling process from a known device of one family to a known device of another family: less than 5 minutes.
- Time of hardware reconfiguration and software policy adaptation when switching the dismantling process from a known device to an unknown device within the same family of devices, provided that the same sequence of operations can be applied (albeit with changed parameters): less than 15 minutes.
- Reconfigurability/adaptation index (RAI): the percentage of devices from a device family that can be dismantled without manual intervention when switching from one device to another: over 33%.
- Exception detection index (EDI): the percentage of detected workpieces that can't be dismantled but are correctly classified as such: over 70%.

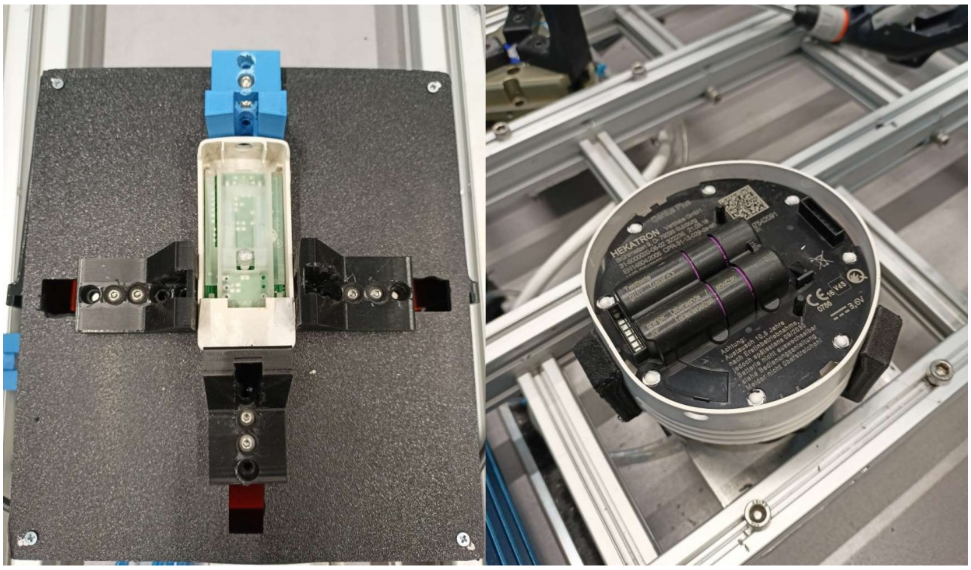


Figure 3: Examples of successful clamping operations for an HCA (left) and a smoke detector (right).

3.1.1 Robustness and reliability of inserting electronic devices into the work-holding devices

We utilize various work-holding devices (fixtures) at different stages of disassembly for electronic devices. Successful insertion into the work-holding device is achieved when the electronic device is firmly clamped, allowing us to proceed with subsequent operations.

For disassembling heat cost allocators (HCAs), we use a pneumatic vise equipped with two pairs of jaws: one pair aligned perpendicular to the narrow side of the device, and the other aligned perpendicular to the longer side.

We first evaluated the insertion of a short-type HCA with an intact plastic housing (no missing exterior walls). The robot, after correctly grasping the short-type HCA, approaches a predefined position above the fully opened vise and releases the device. The pneumatic mechanism then triggers the vise to close both pairs of jaws. Under these conditions, we performed 50 trials and successful clamping was achieved in every instance.

The procedure was similar for the long-type HCA with no damage to its plastic housing. However, when dropped into the open vise, the HCA occasionally rotated, causing the mechanism to jam. To prevent this, we first closed the side jaws, followed by the jaws along the principal axis. With this approach, we conducted 50 trials and achieved successful clamping in 49 cases.

For disassembling smoke detectors, we use a three-jaw chuck to secure the devices before cutting them open with a CNC mill. This procedure was tested on circular smoke detectors with diameters between 100 and 116 mm. Once dropped into the opened chuck, a pneumatic mechanism activates to close the jaws. Under these conditions, we conducted 50 trials and achieved successful clamping in every case.

Examples of successful HCA and smoke detector clamping are depicted in Fig. 3. Depending on the device type, the sucess rate of clamping was betweem 98-100%, exceeding the target KPI of 95%.

3.1.2 Robustness and reliability of battery extraction from different electronic devices

This KPI varies based on the type of device, whether it is a known model or an unknown variant, and the success of preceding operations. We evaluated two different types of HCAs, assuming correct ex-

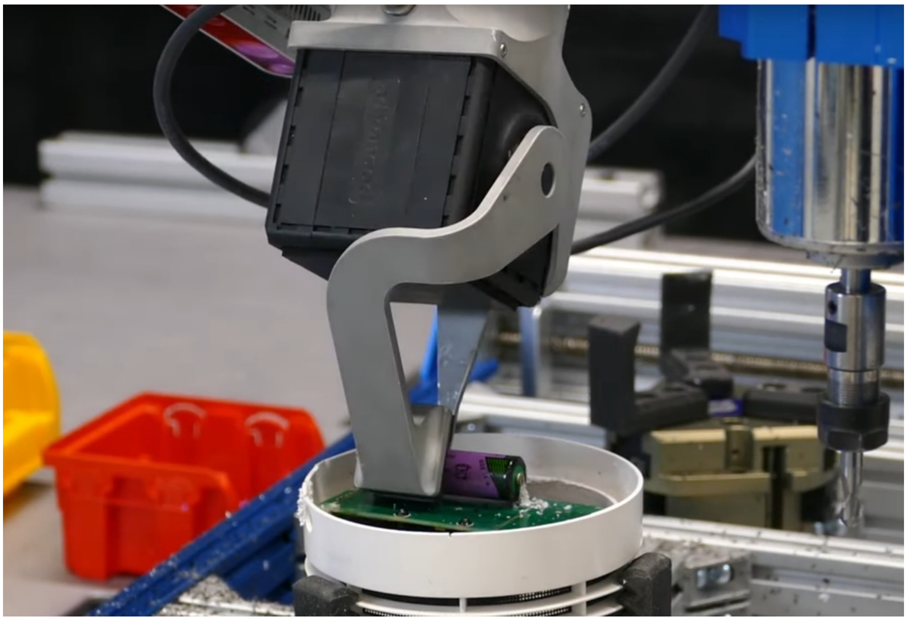


Figure 4: Battery removal by rocking

ecution of the process – specifically, that the PCB with the attached battery was accurately positioned within the pneumatic cutter, with the cutting blade aligned parallel to the battery. The robustness of this process depends on the placement of the PCB, requiring adjustments for non-standard positioning. The proposed solution assumes that the battery is located at the end of the PCB, which was true for both HCA types we assessed. For devices where the PCB is positioned differently (e.g., in the center), multiple cuts may be needed to ensure successful extraction.

Reliability was tested in 20 trials, all of which resulted in successful cuts, effectively separating the battery from the PCB.

There are various methods for extracting batteries from the PCB in smoke detectors, depending on the type of detector. Battery contacts can be severed by using a rocking motion, milling away the contacts, or cutting the wires. We evaluated a smoke detector type where the battery contacts were broken using a rocking motion, as shown in Fig. 4. We assumed that the smoke detector was securely clamped in the fixture, the plastic housing was precisely cut open, and the battery was correctly located for the VS gripper to grasp it. While holding the battery, the robot performed a rocking motion to apply force on the contacts, pulling them apart. The process was deemed successful when no counter-force was detected, at which point the robot transported the still-grasped battery to the designated bin. The maximum duration for the rocking process was set to 25 seconds. If counter-force was still detected after this time, the process was considered a failure, triggering an exception strategy for handling devices with overly strong battery contacts.

Out of 20 battery removal attempts, 18 were successful. The failures primarily occurred during the contact-handling step. In one case, the contacts remained attached to the PCB, causing the entire PCB to be dislodged from the smoke detector case. In the second failure, the extraction process immediately broke one of the two contacts, causing the remaining part to rotate around the intact contact instead of breaking it. This contact was not removed within the maximum process duration, and these cases were left for manual disassembly.

We observed that, for the device types used in our experiments, the remaining contacts could be removed by extending the rocking motion for a longer duration. However, this significantly increases the disassembly time.

Thus, depending on the device type, the reliability of battery extraction process was between 90-100%, achiveing the target KPI of 90%. These findings highlight the importance of precise and adaptable handling mechanisms tailored to specific device types to ensure high reliability in battery extraction across a range of electronic devices.

3.1.3 Cycle time of the dismantling process

The cycle time for the entire disassembly process varies depending on the device. Even when dismantling the same type of device, individual variations can arise, significantly impacting the forces required for tactile operations. To manage these variabilities, we employ different adaptation techniques. Since the duration of these adaptations can fluctuate, we provide time intervals within which specific disassembly operations are typically completed. Additionally, we define parameters to determine when an operation is considered unsuccessful, preventing the process from becoming stalled. The devices handled by our disassembly processes are shown in Fig. 5.

Below we list the typical disassembly operations involved in the dismantling process. They often involve adaptation or self-reconfiguration and therefore vary in duration.

- **PCB levering:** 15-30 sec (parameter determination for unknown or deformed parts), 5 sec (for known device models).

- **Pin pushing:** 3-9 sec (depends on the number of attempts to remove the pin).

- **CNC contact breaking:** 5 sec initialization + 2–5 sec execution (depends on the cut size).

- **Breaking battery contacts by a rocking motion:** 5-25 sec.

- **Cutting of housing by CNC milling:** 5-sec initialization + 10–30 sec execution (depends on the cut size).

- **Opening housing by unscrewing:** typically 10-40 sec (depending on the contact determination time).

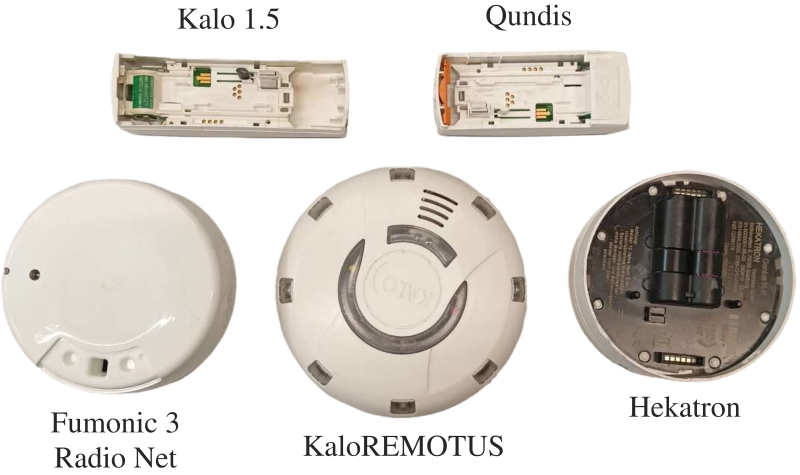


Figure 5: Devices used in our disassembly experiments.

- **Opening housing by levering using a variable stiffness gripper:** 60 sec.

Scene analysis and action prediction queries also vary in duration:

- **Queries to the vision system** (depends on camera resolution, camera frame rate, graphic card, and task): typically 0.1 sec (instance segmentation), up to 3 sec (gap determination) – see WP2.
- **Queries to the VLM system:** typically 2–3 sec using the GPT-4-turbo model via OpenAI API, limited to 5 sec (response time depends on the number of input and output tokens generated by the model).

Other operations include:

- **Point-to-point movements** (transferring objects, approaching positions): typically 1-5 sec.
- **Tool changing operation:** 15 sec.
- **Grasping using pneumatic grippers:** typically 1 sec.
- **Grasping using soft grippers:** typically 3 sec.
- **Flipping of incorrectly placed devices:** 10 sec.
- **Cutting using a linear pneumatic cutter:** typically 2 sec.

The cycle times for the dismantling of a complete device varied depending on the type of the device. The average cycle times for different devices were as follows:

- **80 sec** for long-type HCA disassembly, including tool changing (D5.2) and adaptation (D5.4).
- **75 sec** for short-type HCA disassembly, including adaptation (D5.4).
- **110 sec** for a smoke detector disassembly with breaking battery contacts by a rocking motion (D5.5).
- **70 sec** for a smoke detector disassembly with CNC cutting of the housing (D5.5).

Although the disassembly cycle duration occasionally exceeds the target KPI of 90 seconds, most devices can be disassembled within 90 seconds, which is a technical requirement for the reference software and hardware platform.

3.1.4 Reconfiguration time from a known device to another known device within the same family of devices

Switching from one defined process to another at the software level only requires the operator to load a new task sequence. The process may also involve changing device-specific robot tools, which the operator is instructed to manually place in the correct fixtures. Since the tools are not stored elsewhere, the time required for this is negligible. In any case, in all our use cases the tools needed for different devices within the same family were housed in a dedicated tool-changing station, eliminating the need for manual intervention. The time for this, including program initialization and tool change, is negligible and does not exceed the target KPI of 2 minutes.

3.1.5 Reconfiguration time when switching the dismantling process from a known device of one family to a known device of another family

Changing from one device family to another typically involves exchanging modules in the work cell (e.g., adding a cutter module for cutting PCBs in HCA disassembly). Detaching and attaching a module typically takes around 4 minutes. The modules are automatically initialized within 50 seconds and connected to the ROS network. The operator then needs to load another task sequence. The time required for this is negligible, as already discussed in Sec. 3.1.4. Thus the achieved reconfiguration time is always below the target KPI of 5 minutes, usually significantly less.

3.1.6 Reconfiguration and software policy adaptation time when switching the dismantling process from a known device to an unknown device within the same family of devices

The target KPI for this was 15 minutes because we have initially foreseen more complex adaptation processes. However, for less complex adaptation processes implemented within the workcell such as levering, unscrewing and rocking motion, the achieved adaptation time is significantly lower, i.e. less than 40 seconds.

3.1.7 Reconfigurability/adaptation index (RAI)

This metric measures the percentage of devices within the same family that can be dismantled without manual intervention when switching from one device to another. This applies to all devices we considered, as we have accommodated all necessary tools for the considered devices on a dedicated tool-changing station. For example, the same workcell setup was used to disassemble two different HCA models, and a different configuration managed three distinct smoke detector models without manual intervention. For these devices, the RAI is 100%, exceeding the target value of 50%. However, as additional devices enter the recycling process, it cannot be guaranteed that they can be disassembled within the existing reconfigurable hardware. This limitation underscores the importance of optimal hardware design.

3.1.8 Exception detection index

This index measures the effectiveness of our system in handling exceptions, specifically its ability to identify objects that cannot be dismantled. Correctly classifying such objects is important to prevent the application of incorrect procedures and to avoid damaging hazardous components.

We have incorporated Vision-Language Model (VLM) based action prediction to implement exception handling for unknown devices. In our testing environment, which included nine different device models (HCAs and smoke detectors), the system successfully predicted in 85% when none of the available actions were applicable. For instance, it accurately identified situations where wire cutting was not possible. This ensures that the dismantling process is halted before any inappropriate or potentially damaging actions are taken.

The accuracy of the VLM in predicting the correct next disassembly action (or halting of the process if needed) was tested in various disassembly steps and with varying conditions (i.e., locations and orientations of the smoke detectors, states of the smoke detectors and heat-cost allocators). Within the limited set of nine devices, we have thus achieved and exceeded the target KPI of 70%.

3.2 KPIs in WP2 for Vision Processes

The vision system is responsible for the recognition and pose estimation of the devices, including their parts. Actions can be performed in the workcell based on these detections. We examine the KPIs defined at the project's beginning and determine if they have been met.

3.2.1 Instance Segmentation Accuracy for Recognition and Pose Estimation

We use instance segmentation to find devices and their components from images. Instance segmentation is a computer vision task that identifies and delineates each object in an image at the pixel level, distinguishing between different instances of the same object category. It combines object detection and semantic segmentation to classify objects and provide precise boundaries for each instance.

While these methods do not provide direct object pose estimation, they can serve as a first step in the process. If we have certain information about the object, such as its height and the camera's orientation relative to the object – by, for example, restricting the camera to a top-down view – we can estimate the object's pose using the segmentation mask. Though this approach has limitations, it also offers advantages. The limitation is that the pose estimation is less detailed compared to 6D pose estimation methods. However, the advantage is that object labeling is faster, and labeling and training on internal device components are much simpler than with 6D pose estimation methods.

Mean Average Precision (mAP) is a metric used to evaluate the accuracy of instance segmentation models. It calculates the average precision for each class at various Intersections over Union (IoU) thresholds, considering both precision (how many detected objects are correct) and recall (how many true objects are detected). Essentially, mAP represents the area under the precision-recall curve. We denote mAP with an IoU threshold of 50% as mAP@50. For evaluation we use the average of mAP@50, mAP55, . . . , mAP95, denoted as mAP@50-95. For our model, we examine the mAP@50-95 for both box and mask tasks. The box is the bounding box of the mask aligned with the image.

We use the Yolov8 [2] segmentation model, which is currently one of the most performant segmentation models. It outperforms YOLACT [1], which we initially used at the start of the project. While the standard image size for Yolov8 is 640x640, we increased this to 1280x1280 for improved resolution. Additionally, we opted for the P2 model over the standard model due to its superior performance in detecting small details, such as screws and wires.

We created a dataset of 4458 images containing all the classes shown in Table 1. The dataset was augmented to 20,000 images by changing the background of the images, varying the lighting, and putting multiple devices on the same image. These augmentations enhance the model's robustness against irrelevant background objects and changing lighting conditions.

The dataset also includes separate labels to indicate whether a device is empty. This labeling choice allows the model to determine if a device has been fully disassembled. For HCAs, the possible classes are front, back, and back empty. The "back" and "back empty" classes indicate whether the back of the device is shown and whether it is empty of components. For the smoke detector, the possible classes are front, back, insides, and empty. It is not always clear from an image whether the front or back of the smoke detector has been removed to reveal the components. herefore, the "insides" class is used when the internal components are visible. The "empty" class is applied when the housing of the smoke detector has been emptied of its components..

The device components are categorized into the following classes: battery, PCB, plastic clip, wires, and screw. For battery and PCB, we also have an additional "covered" class, representing instances where the battery or PCB is partially visible but obstructed by plastic housings. The "internals" class corresponds to miscellaneous internal parts that do not fit into the other categories. It often consists of plastic components used in the device assembly.

The results are shown in Table 1. The HCA (front, back, back empty) box mAP@50-95 is around 95%. For the mask, we score mAP@50-95 of 75%-96%.

KPI: HCA box mAP@50-95 larger than 95% satisfied.

The pose of the HCA is determined directly from the mask segmentation up to its rotational invariances when the battery is not detected. On a test of 60 HCAs, front and back, we score 100% accuracy for the correct rotation.

KPI: orientation of device correct 95% accuracy satisfied.

The smoke detector (front, back, insides, empty) box mAP@50-95 is around 96% and the mask mAP@50-95 score is between 83% and 95%. Since the device's orientation cannot be determined from the mask when internal components are not visible because of the circular design of the devices, we use a different method for rotation estimation (see Section 3.2.2).

KPI: smoke detector box mAP@50-95 larger than 95% satisfied.

The battery has box mAP@50-95 of 88% and mask mAP@50-95 of 82%.

KPI: battery box mAP@50-95 larger than 95% NOT satisfied.

The other components of the device: internals, PCB, PCB covered, plastic clip, score a box mAP@50-95 of 89%-95% and a mask mAP@50-95 of 85%-93%.

KPI: Remaining device internal components box mAP@50-95 larger than 85% satisfied.

The wires achieve a box mAP@50-95 of 96% and a mask mAP@50-95 of 60%, while the screws achieve a box mAP@50-95 of 58% and a mask mAP@50-95 of 47%. The challenge with detecting both wires and screws lies in their small size, as they often occupy only a few pixels in large images, making detection difficult. To address this, we switched to the YOLOv8 p2 model, which enhances the performance in detecting small features. For optimal results in screw and wire detection, the object should occupy a significant portion of the image.

Class	Instances	**Box(mAP@50-95)**	**Mask(mAP@50-95)**
all	7209	**0.903**	**0.84**
HCA front	429	**0.978**	**0.958**
HCA back	414	**0.962**	**0.749**
HCA back empty	342	**0.94**	**0.909**
smoke det. front	396	**0.958**	**0.947**
smoke det. back	416	**0.936**	**0.915**
smoke det. insides	393	**0.947**	**0.829**
smoke det. insides empty	371	**0.945**	**0.927**
internals	530	**0.952**	**0.927**
battery	995	**0.88**	**0.823**
battery covered	302	**0.917**	**0.875**
PCB	862	**0.893**	**0.853**
PCB covered	398	**0.935**	**0.917**
plastic clip	366	**0.946**	**0.869**
wires	593	**0.822**	**0.598**
screw	378	**0.576**	**0.468**

Table 1: YOLOv8 results on test set.

The pose of the battery is also determined directly from the mask segmentation up to its rotational invariances.

3.2.2 Rotation Estimation Accuracy

The pose consists of position and rotation. The position is found using the results of segmentation. The rotation can also be found from the segmentation up to its rotation invariances. The correct rotation from the invariances can be chosen if other features are found by the segmentation method.

When this is not the case, a device's rotation can be found by finding local features in the device's image and matching them with the template definition. The device's rotation can be found if enough local features are matched. To achieve this, we use SuperPoint [12] for keypoint and descriptor extraction and SuperGlue [11] for feature matching. We extend the augmentations for the of training SuperGlue and achieve the results shown in Table 2. We find that we can estimate the rotation correctly to a degree on the test set, containing both devices seen in training and devices not seen in training.

	Seen Test			Unseen Test		
	Med	Mean	Std.	Med	Mean	Std.
L1 difference in degrees	$0.19°$	$0.40°$	$0.73°$	$0.26°$	$0.46°$	$0.95°$

Table 2: Performance of SuperGlue on the test set for rotation estimation

These results show that SuperGlue can estimate the rotation to within a degree. However, the images used to test the model are synthetically generated rotations of the same device. Therefore, this task is easier than with real-world differing perspectives of devices.

We use the SuperGlue-based method to determine the rotation of smoke detectors. On a test set of 60 images of smoke detectors, we achieve 83% accuracy in correct rotation. The poorly performing cases are where the device has many symmetries and no clear features to guide SuperGlue to the correct orientation. On a subset of those, ignoring devices with little to no features to show the correct orientation, we achieve 92% accuracy. On images of smoke detectors showing the back of the device, we achieve 100% accuracy. The backs make it a lot easier to determine the rotation because there are many features to guide it.

KPI: orientation of device correct 95% accuracy is satisfied under the constraint that the device must have clear features.

3.2.3 Device Classification Accuracy

The dataset for device classification consists of images of HCAs and smoke detectors. The images are of the front and back of the device and of the device in various states of disassembly. We have categorized the devices as shown in Figure 6. We chose this categorization based on a few observations. Some devices are physically the same, except for a different brand written on them, in which case we say they are the same. Some devices are almost the same except for some small variations. This can be when the front of the device is the same, but the back has some variation, like for example missing an element, a different tab layout, or a color variation. In these cases, we say they are a derivative of the same device, and we give it a name of x.y, where x is the main class number (see Figure 6).

There are 13 unique HCAs and 10 unique smoke detectors. Accounting for the variations in models, there are altogether 60 classes:

- 14 HCA fronts
- 16 HCA backs

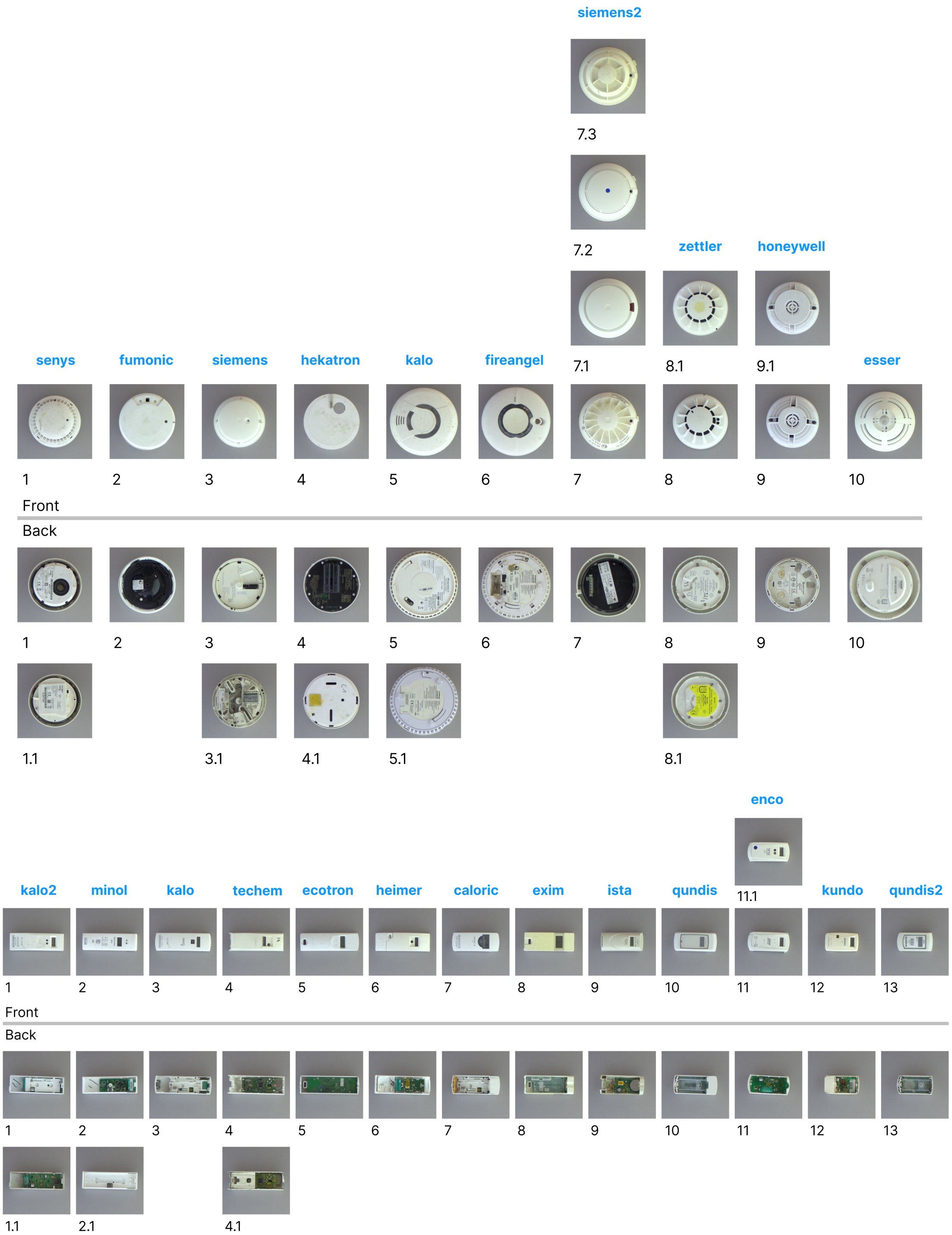


Figure 6: Chart of devices. The back and front are shown for each device, including the variations.

- 15 smoke detector fronts
- 15 smoke detector backs

The train/val/test split was 0.8/0.1/0.1, with the number of images in each class being 1328/183/183. The split has been constructed such that the validation and test sets contain uniform images over the classes.

We obtained 99.1% overall accuracy on the test set. HCA back two scores 75% accuracy, confusing it with HCA back 4. HCA back seven scores 75%, confusing it with HCA back 1.1

KPI: the device should be labelled correctly 95% of the time, satisfied.

3.2.4 Accuracy of Pixel to 3-D Position Mapping

We implemented a mapping that determines the position of objects in real-world coordinates using an RGB-D camera mounted to the robot end-effector and the RGB camera mounted above the work surface. First we determine the location of objects in an image in pixel coordinates. We If available, depth information is then used to determine the real-world object pose. For an RGB image without depth information, we use the work surface's physical dimensions to determine the objects' real-world coordinates.

Depth-based Estimation

We use the Realsense D435 RGB-D camera. To evaluate the depth-based estimation, we have a grid of 3x3 dots on paper that are spaced evenly 50mm apart. The camera is positioned above the paper

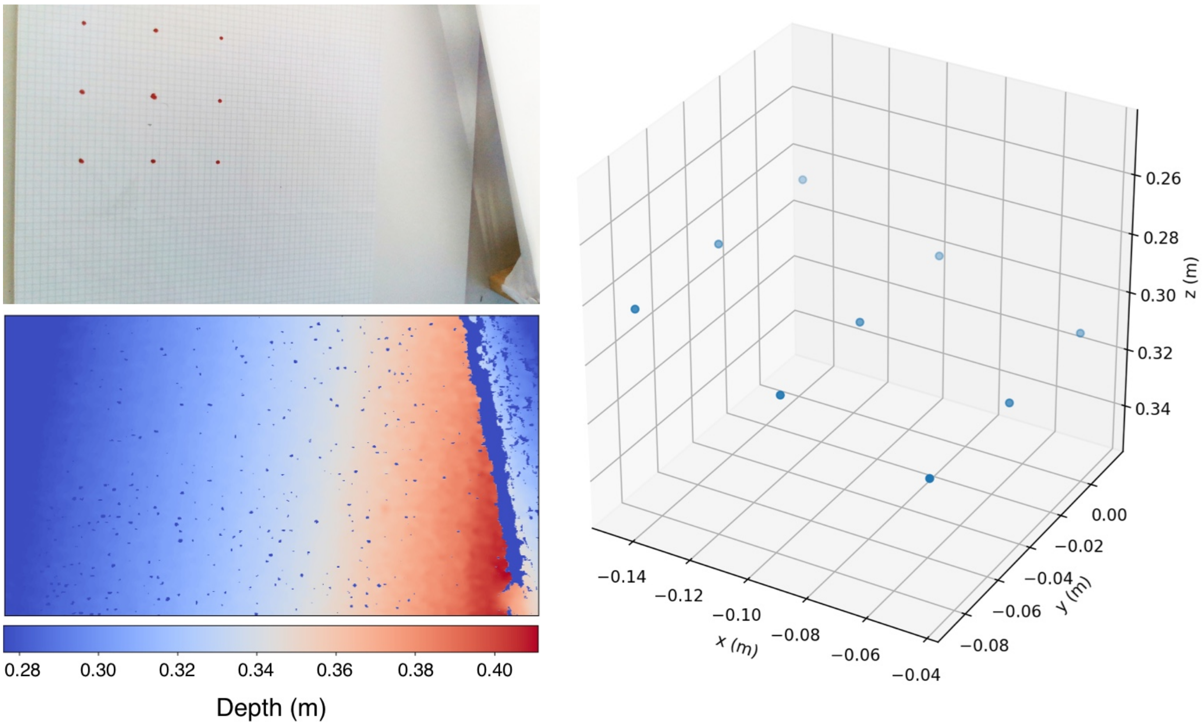


Figure 7: Example of a 3x3 grid of dots spaced 50mm apart. The RGB image is on the top-left, with the accompanying depth image on the bottom-left. The Real-world dot deprojection is on the right, w.r.t. the camera center.

at $\sim$320mm. We detect the dots using OpenCV and deproject them using the depth information. An example of an image of dots and its deprojection is shown in Figure 7. We position the camera in 9 different poses on the paper.

The mean distance between dots is 49.9mm, with a standard deviation of 0.5mm. From this, we determine that the depth is accurate to 0.5mm, at the approximate height of $\sim$320mm, when considering the depth of a smooth surface.

Estimating the position of a device is more difficult than estimating the position of a dot on a paper because the device can have a varying depth profile. Furthermore, the position of the device w.r.t. the camera changes the error in the position. If the device is centered below the camera, the error in the x, y position is 0mm. If the device is on the edge of the camera image, the error in the x, y position is 3mm. This error comes from the center of the segmentation mask not being the center of the object because of the perspective of the object w.r.t. the camera. The object is viewed slightly from the side instead of directly above, as is the case when the object is in the center of the image.

Work Surface-based Estimation

The Basler camera is mounted above the work surface. The work surface comprises a square metal plate with bolt holes around the edge. We use the bolts to find the position of the work surface. The camera is undistorted, and intrinsics are found via camera calibration.

The bolts are found using the Hough circle transform, with the parameters found using trial and error. We know the locations of the bolts in real-world coordinates. We use a 2D affine point cloud registration method [10], followed by a nearest neighbor search to match the found bolts with the ground truth bolts. The point set registration method finds the affine transformation between two unordered sets of points.

Given the matching, we compute an affine transform from pixel bolt positions to the real-world bolt positions. This affine transform gives us the required coordinate transform.

We use the affine transform to transform the bolt positions from pixels to meters. The Euclidean distance between the ground truth bolt positions and the affine transform has a mean error of 1.2mm and a maximum error of 2.8mm.

The error of the position of a device depends on its position on the work surface. If the object is in the middle of the work surface, such that it is in the middle of the camera image, the position is more accurate than if the device is on the edge of the work surface and, therefore, on the edge of the image.

For a device in the center of the image, the error in the x, y position is 4mm. For a device on the edge of the image, the error in the x, y position is 10mm. This error is larger than the error when the device is in the center of the image, and this is the same as for the Realsense camera and comes from the perspective not being centered below the camera.

KPI: The position of the objects should be accurate to ±5mm in the x and y direction. This KPI is satisfied when the objects are viewed using the Realsense camera and the Basler camera under the constraint that the device is near the center of the image.

3.2.5 System Speed

We run the vision pipeline on an Nvidia 1080Ti. The images from the Realsense camera are processed in an average time of 200ms. The images from the Basler camera can be processed in an average time of 250ms.

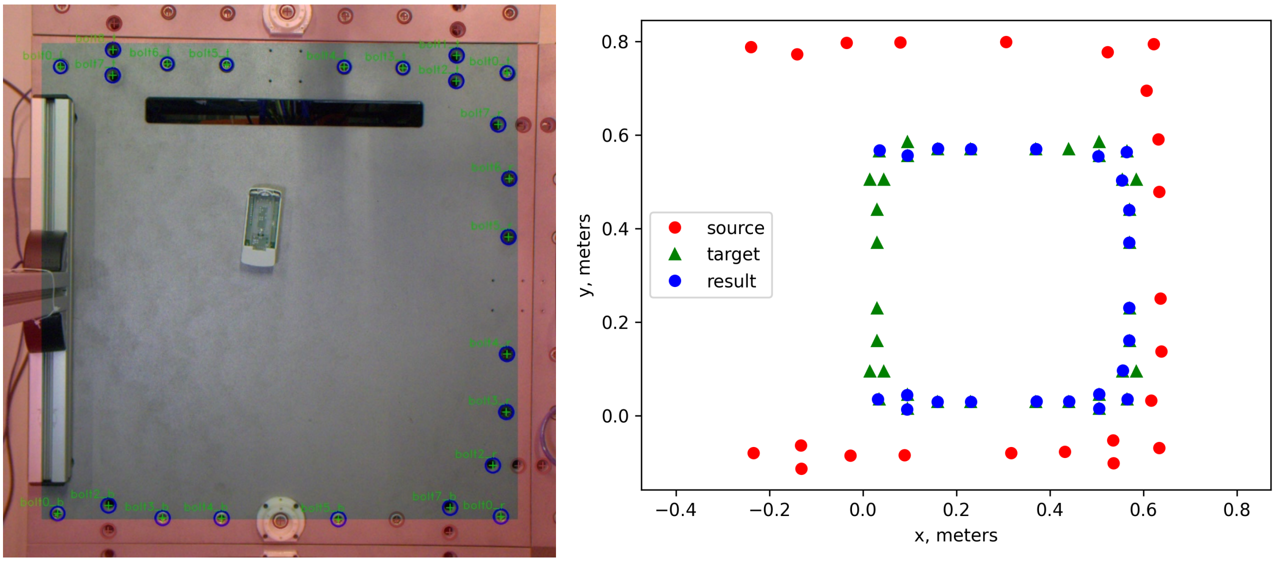


Figure 8: Point set registration for bolt matching. The ‘source’ is the input bolts, scaled up. The ‘target’ is the ground truth bolt locations. The ‘result’ is the source bolts fitted onto the target bolts.

KPI satisfied: The system should be able to process an image in under 500ms when running on modern computer hardware coupled with an Nvidia 1080Ti graphics card or better.

3.3 KPIs in WP3 for Tactile Robots and their Control

Preliminary KPIs introduced in the previous deliverable D5.3:

- Three main disassembly tactile skills were developed, covering the required operations for our use cases: levering, unscrew-driving, and tactile exploration. Each includes four basic custom operations: going to pose, establishing contact, applying force, and manipulating the workpiece.
- Success rate for implementing tactile skills under various conditions: 100% due to the tactile exploration skill.
- Force and displacement error from motion generator and unified force-impedance control: < 3 N, < 4mm discussed in detail later [8].
- Improvement of controller efficiency by adaption and learning parameters [5], [6], [13]
- Handling faults due to unforeseen external wrench: 100% [5], [6], [13].
- Stability of the controller: 100% [4].

However, the robot’s skill performance should be analyzed more broadly, and a comprehensive benchmarking system that can handle various scenarios, including dynamically adapting to different disassembly processes, is required. For example, in the context of disassembly, removing a battery from an electronic device requires several coordinated actions: first, the robot must *Go To Pose* itself accurately to align with the battery; second, it needs to *Establish Contact* with the battery carefully to avoid damaging it; third, the robot must *Manipulate Workpiece* to lever the battery; and finally, it should *Apply Force* appropriately to detach the battery without causing harm. These operations highlight the need for precision, adaptability, and control in disassembly, ensuring tasks are completed efficiently and safely.

Thus, we developed experimental setups that specifically address the complexities of robotic disassembly, ensuring that the KPIs and performance metrics are comprehensive and relevant to the unique demands of this field. Additionally, we extended our approach to benchmark and standardize all the relevant industrial tasks. This extended version has also been submitted as a publication and is in review now [8].

3.3.1 Process-Based Tactile Performance Metrics

Twenty-five robot tactile performance metrics are derived for disassembly to determine the robot control performance requiring physical interaction, namely, tactile performance metrics [9] [8]. In summary, the five main groups of tactile performance metrics are as follows.

1. **Force Sensing Metrics** measure the system's capability to sense forces accurately, precisely, and consistently over time, using the maximum force deviations from a reference mass attached to the robot end-effector. The definitions of measurement accuracy, resolution, precision, and sensing drift after $1\,\mathrm{min}$, $10\,\mathrm{min}$, $1\,\mathrm{h}$ and $8\,\mathrm{hrs}$ are defined.

2. **Force Control Metrics** measure the quality of controlling end-effector interaction forces, which is essential for manipulating objects and the environment. They quantify the accuracy of the applied forces compared to a desired force, the applied force precision, resolution, settling time, overshoot forces, minimal force that the system can apply, and the stability under changing environmental conditions such as materials and impact forces need to be considered.

3. **Force Reaction Metrics** measure the robot's capability to detect and react to contacts. They quantify force-sensitive detection of obstacles or contacts that can be desired or accidental during motion. The two metrics describing this capability are *contact sensitivity* and *tactile contact sensitivity*. They differentiate in the considered robot contact speeds and, thus, also in the resulting contact force.

4. **Human Safety Metrics** measure the physical contact safety concerning the current state of safety regulations [3], especially in constrained contact scenarios. While the initial contact forces are primarily influenced by the robot's dynamics and are affected by effective mass and impact velocity, the quasi-static clamping forces remaining after first contact can be controlled via appropriate collision handling. Current industrial standards provide references for acceptable safety thresholds. Therefore, using a standardized test device, we measure the conformance to transient and quasi-static contact force thresholds according to [3], and the human safety metrics are defined.

5. **Teaching Metrics** define, measure, and rate the quality and smoothness of the hand-guiding functionality, i.e., manual maneuverability, of a robot. The *minimum motion force*, *guiding force*, *guiding force deviation*, required *guiding energy*, and *maneuver effort* are defined.

For all metrics, a reproducible experimental standard protocol and an according experimental setup were developed, respectively. These may serve future robot standards as templates. The twenty-five metrics depicted in Fig.9 are crucial in defining the performance feature vector of robot tactility. These metrics, including force profile error, load/force estimation error, and force tolerance, collectively determine tactile performance.

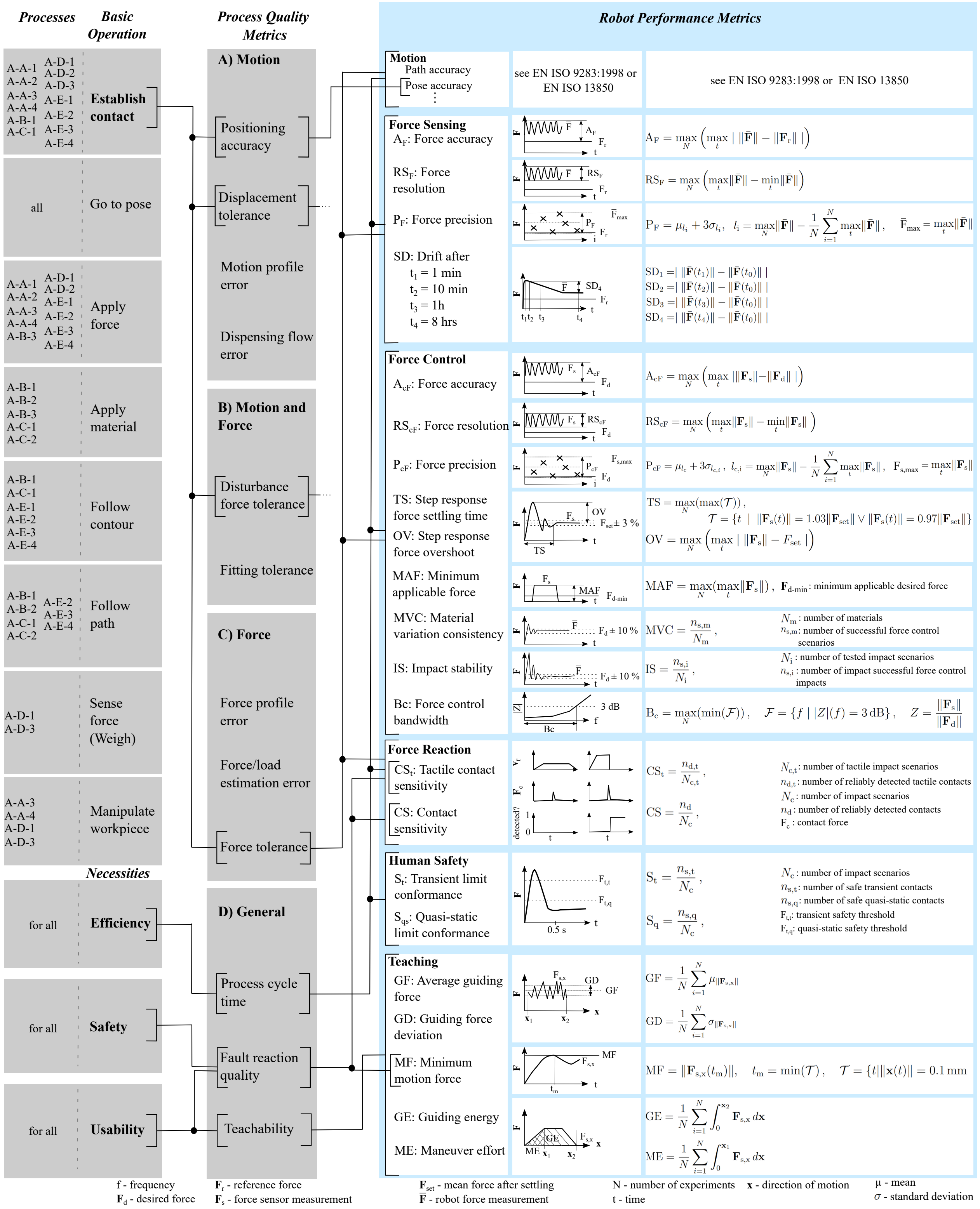

Processes
Basic Operation
Process Quality Metrics
Robot Performance Metrics
A-A-1 A-A-2 A-A-3 A-A-4 A-B-1 A-C-1 A-D-1 A-D-2 A-D-3 A-E-1 A-E-2 A-E-3 A-E-4
Establish contact
all
Go to pose
A-A-1 A-A-2 A-A-3 A-A-4 A-B-3 A-D-1 A-D-2 A-E-1 A-E-2 A-E-3 A-E-4
Apply force
A-B-1 A-B-2 A-B-3 A-C-1 A-C-2
Apply material
A-B-1 A-C-1 A-E-1 A-E-2 A-E-3 A-E-4
Follow contour
A-B-1 A-B-2 A-C-1 A-C-2 A-E-2 A-E-3 A-E-4
Follow path
A-D-1 A-D-3
Sense force (Weigh)
A-A-3 A-A-4 A-D-1 A-D-3
Manipulate workpiece
Necessities
for all
Efficiency
for all
Safety
for all
Usability
A) Motion
Positioning accuracy
Displacement tolerance
Motion profile error
Dispensing flow error
B) Motion and Force
Disturbance force tolerance
Fitting tolerance
C) Force
Force profile error
Force/load estimation error
Force tolerance
D) General
Process cycle time
Fault reaction quality
Teachability
Motion
Path accuracy
Pose accuracy
see EN ISO 9283:1998 or EN ISO 13850
see EN ISO 9283:1998 or EN ISO 13850
Force Sensing
A_F: Force accuracy
$A_F = \max_N \left(\max_t \mid \|\bar{\mathbf{F}}\| - \|\mathbf{F}_r\| \mid \right)$
RS_F: Force resolution
$RS_F = \max_N \left(\max_t \|\bar{\mathbf{F}}\| - \min_t \|\bar{\mathbf{F}}\| \right)$
P_F: Force precision
$P_F = \mu_{l_i} + 3\sigma_{l_i}, \quad l_i = \max_N \|\bar{\mathbf{F}}\| - \frac{1}{N}\sum_{i=1}^{N} \max_t \|\bar{\mathbf{F}}\|, \quad \bar{F}_{max} = \max_t \|\bar{\mathbf{F}}\|$
SD: Drift after
t_1 = 1 min
t_2 = 10 min
t_3 = 1h
t_4 = 8 hrs
$SD_1 = \mid \|\bar{\mathbf{F}}(t_1)\| - \|\bar{\mathbf{F}}(t_0)\| \mid$
$SD_2 = \mid \|\bar{\mathbf{F}}(t_2)\| - \|\bar{\mathbf{F}}(t_0)\| \mid$
$SD_3 = \mid \|\bar{\mathbf{F}}(t_3)\| - \|\bar{\mathbf{F}}(t_0)\| \mid$
$SD_4 = \mid \|\bar{\mathbf{F}}(t_4)\| - \|\bar{\mathbf{F}}(t_0)\| \mid$
Force Control
A_{cF}: Force accuracy
$A_{cF} = \max_N \left(\max_t \mid \|\mathbf{F}_s\| - \|\mathbf{F}_d\| \mid \right)$
RS_{cF}: Force resolution
$RS_{cF} = \max_N \left(\max_t \|\mathbf{F}_s\| - \min_t \|\mathbf{F}_s\| \right)$
P_{cF}: Force precision
$P_{cF} = \mu_{l_c} + 3\sigma_{l_{c,i}}, \quad l_{c,i} = \max_N \|\mathbf{F}_s\| - \frac{1}{N}\sum_{i=1}^{N} \max_t \|\mathbf{F}_s\|, \quad F_{s,max} = \max_t \|\mathbf{F}_s\|$
TS: Step response force settling time
OV: Step response force overshoot
$TS = \max_N(\max(\mathcal{T})), \quad \mathcal{T} = \{t \mid \|\mathbf{F}_s(t)\| = 1.03\|\mathbf{F}_{set}\| \vee \|\mathbf{F}_s(t)\| = 0.97\|\mathbf{F}_{set}\|\}$
$OV = \max_N \left(\max_t \mid \|\mathbf{F}_s\| - F_{set} \mid \right)$
MAF: Minimum applicable force
$MAF = \max_N(\max_t \|\mathbf{F}_s\|)$, $\mathbf{F}_{d\text{-}min}$: minimum applicable desired force
MVC: Material variation consistency
$MVC = \frac{n_{s,m}}{N_m}$,
N_m: number of materials
$n_{s,m}$: number of successful force control scenarios
IS: Impact stability
$IS = \frac{n_{s,i}}{N_i}$,
N_i: number of tested impact scenarios
$n_{s,i}$: number of impact successful force control impacts
Bc: Force control bandwidth
$B_c = \max_N(\min(\mathcal{F})), \quad \mathcal{F} = \{f \mid |Z|(f) = 3\,dB\}, \quad Z = \frac{\|\mathbf{F}_s\|}{\|\mathbf{F}_d\|}$
Force Reaction
CS_t: Tactile contact sensitivity
CS: Contact sensitivity
$CS_t = \frac{n_{d,t}}{N_{c,t}}$,
$CS = \frac{n_d}{N_c}$,
$N_{c,t}$: number of tactile impact scenarios
$n_{d,t}$: number of reliably detected tactile contacts
N_c: number of impact scenarios
n_d: number of reliably detected contacts
F_c: contact force
Human Safety
S_t: Transient limit conformance
S_{qs}: Quasi-static limit conformance
$S_t = \frac{n_{s,t}}{N_c}$,
$S_q = \frac{n_{s,q}}{N_c}$,
N_c: number of impact scenarios
$n_{s,t}$: number of safe transient contacts
$n_{s,q}$: number of safe quasi-static contacts
$F_{t,t}$: transient safety threshold
$F_{t,q}$: quasi-static safety threshold
Teaching
GF: Average guiding force
GD: Guiding force deviation
$GF = \frac{1}{N}\sum_{i=1}^{N} \mu_{\|\mathbf{F}_{s,x}\|}$
$GD = \frac{1}{N}\sum_{i=1}^{N} \sigma_{\|\mathbf{F}_{s,x}\|}$
MF: Minimum motion force
$MF = \|\mathbf{F}_{s,x}(t_m)\|, \quad t_m = \min(\mathcal{T}), \quad \mathcal{T} = \{t \mid \|\mathbf{x}(t)\| = 0.1\,mm$
GE: Guiding energy
ME: Maneuver effort
$GE = \frac{1}{N}\sum_{i=1}^{N} \int_0^{\mathbf{x}_2} \mathbf{F}_{s,x}\,dx$
$ME = \frac{1}{N}\sum_{i=1}^{N} \int_0^{\mathbf{x}_1} \mathbf{F}_{s,x}\,dx$
F
t
i
$\bar{F}$
F_r
A_F
RS_F
P_F
$\bar{F}_{max}$
SD_4
$t_1 t_2$ t_3 t_4
F_s
F_d
A_{cF}
RS_{cF}
P_{cF}
$F_{s,max}$
OV
$F_{set} \pm 3\,\%$
TS
MAF
$F_{d\text{-}min}$
$F_d \pm 10\,\%$
|Z|
3 dB
f
Bc
v_r
F_c
detected?
1
0
$F_{t,t}$
$F_{t,q}$
0.5 s
$F_{s,x}$
GD
GF
x
x_1
x_2
MF
t_m
GE
ME
f - frequency
F_d - desired force
F_r - reference force
F_s - force sensor measurement
F_{set} - mean force after settling
$\bar{F}$ - robot force measurement
N - number of experiments
t - time
x - direction of motion
μ - mean
σ - standard deviation

Figure 9: The robot motion and tactile performance metrics for industrial process, including disassembly. This version was submitted as a publication and is in review now [8].

3.4 KPIs in WP4 for Soft Grippers and Tools

The foreseen KPIs are:

- A. Pinch and power grasp: standard dynamo meters can be used to measure the force of the hand's grasp. This value is deemed the main performance indicator for a gripper. Pinch Grasp: [1 N –5N], Power Grasp: [1N – 50N].

- B. Capability of the hand to hold objects with different shapes and orientations. Pinch Grasp: [> 0,2 kg], Power Grasp: [> 2 kg], Hanging: [> 5 kg].

- C. Opening and closing time: Since the opening and closing commands are executed many times in a given task, their running time strongly impacts the whole duration of the dismounting process. Closing time: [1,5 sec].

- D. Possibility to reconfigure the gripper. End-effector reconfiguration time: [< 20 sec].

- E. Quickly engage and disconnect wrist would also contribute to speeding up the disassembly task – End-effector exchange time: [< 60 sec].

- F. Electromagnetic compatibility for the final robotic end-effector integrated into the cell measured according to EN ISO 6100-6-2 / 6-4.

- G. Estimated price of end-effector for the final robotic cell. Two fingers: [< 3 k€], Five fingers: [< 10k€].

- H. Successful grasp repeatability for different orientations of objects relevant to the ReconCycle Project. Parallelepiped shape: > 90%, disk shape: > 90%, cylindrical shape: > 90%.

3.4.1 KPI Evaluations

These tests have been carried out in qblab by certified force cells and customized tools to fit the grippers' shape. For force estimation, qbrobotics designed a handle with two force cells for hands and a custom pad for the gripper. Grippers grasped four samples from 2 to 45mm thick, 50 times each. The force values have been obtained by the GUI provided by the sensor supplier, and the reported KPI is the average value. The test consists of grasping samples of sphere shape (cylindrical shape for hanging approach) from 2 to 45mm diameter and a known load connected by a wire. A positive result consists of firmly keeping the load for 10 seconds.

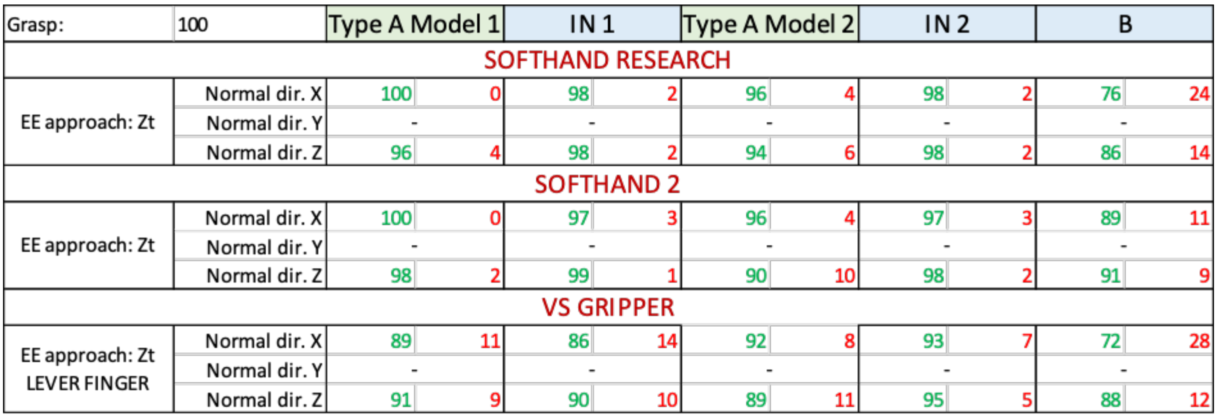
To estimate the closing time, we measured the interval from sending the closing reference to achieving the closing condition, where the motor speed and acceleration reached zero (static condition). The reconfiguration of SoftHand 2 can be considered continuous due to its dexterity. In contrast, for the VS gripper, the substitution time of both fingers was tested.

There were two Franka Emika robots in the workcell. The main phases involving end-effectors in the disassembly sequence are:

- grasping the device and positioning in the clamp;

- levering to remove the inner structure IN (device without external frame);

- tool changing for picking and positioning of IN in the cutter;



Grasp:	100	Type A Model 1		IN 1		Type A Model 2		IN 2		B	
SOFTHAND RESEARCH											
EE approach: Zt	Normal dir. X	100	0	98	2	96	4	98	2	76	24
	Normal dir. Y	-		-		-		-		-	
	Normal dir. Z	96	4	98	2	94	6	98	2	86	14
SOFTHAND 2											
EE approach: Zt	Normal dir. X	100	0	97	3	96	4	97	3	89	11
	Normal dir. Y	-		-		-		-		-	
	Normal dir. Z	98	2	99	1	90	10	98	2	91	9
VS GRIPPER											
EE approach: Zt LEVER FINGER	Normal dir. X	89	11	86	14	92	8	93	7	72	28
	Normal dir. Y	-		-		-		-		-	
	Normal dir. Z	91	9	90	10	89	11	95	5	88	12

Table 3: Data collection process for grasping tests with soft end-effectors

- grasping of critical component B (battery);
- grasping of external frame;

Our tests focused on the grasping of HCA and subcomponents (phases 1, 3, 4, and 5) with three devices: qb SoftHand Research, Research 2, and the VS gripper. Two heat cost allocator Type A models were used; they have similar battery locations, dimensions, and structures. Table 3 shows the main dimensions and direction of the reference system of devices and subcomponents.

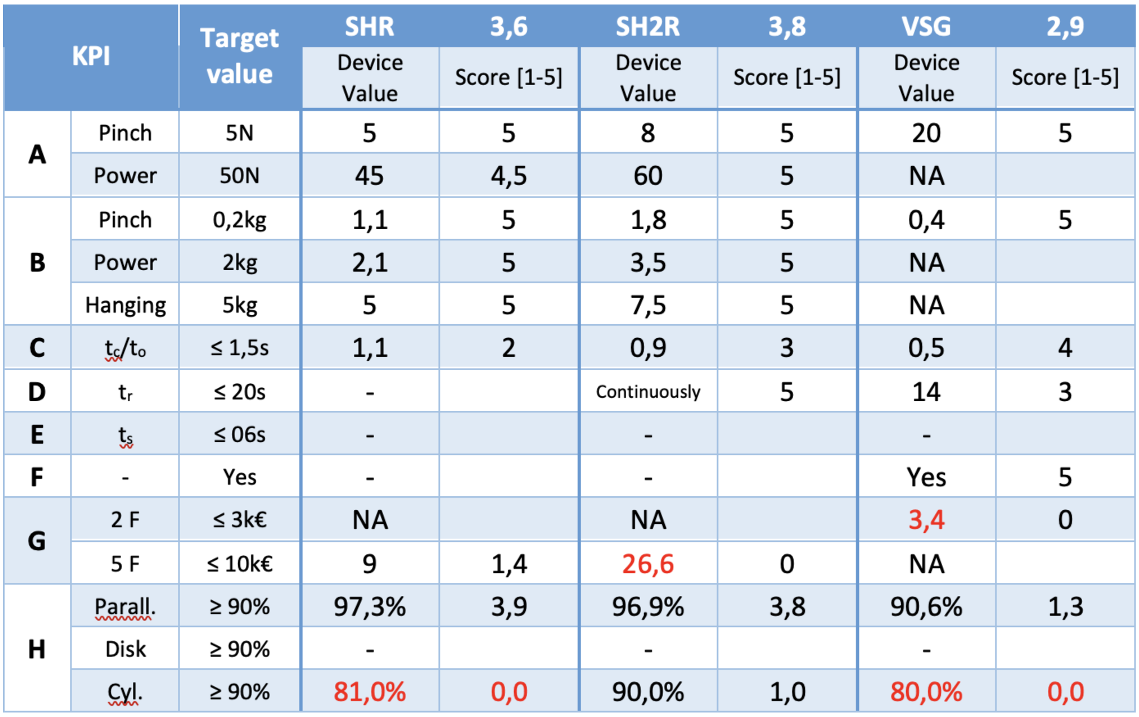
A pick-and-place open loop has been implemented in ROS. Each component is grasped fifty times to evaluate the repeatability of grasping. The end-effector approach direction is parallel to the normal $\mathbf{n}$ of the work-plane surface. We did not use a vision system, so the object's center was always placed in the same position but with random orientation to simulate the variability of its positioning in the work cell. As shown in Table 3, tests were scheduled by positioning the objects as follows: i) z direction parallel to normal $\mathbf{n}$ and ii) x direction parallel to normal $\mathbf{n}$. For the robotic hands, the wrist orientation allows one to grasp the object mainly by exploiting the opposition of the thumb with the forefinger. The object's x direction is approximately normal to the finger-pad surface for the VS gripper and parallel to the thumb-index closure direction for the hands. All tests were performed using ROS, running cyclically in the desired sequence. Particular attention was paid to the grasp of battery "B" due to its dimensions and warning. All test results have been collected in an Excel file to calculate the grasping repeatability.

The results for qb SoftHand Research, the qb SoftHand 2 Research, and the Variable Stiffness Gripper are shown in Tab. 4. Tests on disk shape samples have not been carried out because they are not attributable to the components of HCAs, and the smoke alarms are too big to be grasped from the circular surface.

For each KPI, the following scores were applied: Score (A - B): 1 for value = 0.1*target and 5 for value = target; Score (C - G): 1 for value = target and 5 for value = 0.1*target. We computed a median value for each device to evaluate performance.

From a technical point of view, the SH2R has a better performance than SHR, particularly for KPI "H", which is not fully satisfied by SHR. Moreover, the SH2R can continuously reconfigure the kinematic movements of fingers to achieve an excellent grasp versatility.

KPIs about costs for the new end-effectors SH2R and VSG were also evaluated. Considering the cost composition (CNC machining, injection, electronics, design, royalties, etc.) and discounts based on quantities, we can estimate the cost variation in these cases:



KPI		Target value	SHR	3,6	SH2R	3,8	VSG	2,9
			Device Value	Score [1-5]	Device Value	Score [1-5]	Device Value	Score [1-5]
A	Pinch	5N	5	5	8	5	20	5
	Power	50N	45	4,5	60	5	NA	
B	Pinch	0,2kg	1,1	5	1,8	5	0,4	5
	Power	2kg	2,1	5	3,5	5	NA	
	Hanging	5kg	5	5	7,5	5	NA	
C	t_c/t_o	≤ 1,5s	1,1	2	0,9	3	0,5	4
D	t_r	≤ 20s	-		Continuously	5	14	3
E	t_s	≤ 06s	-		-		-	
F	-	Yes	-		-		Yes	5
G	2 F	≤ 3k€	NA		NA		3,4	0
	5 F	≤ 10k€	9	1,4	26,6	0	NA	
H	Parall.	≥ 90%	97,3%	3,9	96,9%	3,8	90,6%	1,3
	Disk	≥ 90%	-		-		-	
	Cyl.	≥ 90%	81,0%	0,0	90,0%	1,0	80,0%	0,0

Table 4: KPIs evaluation for qb SoftHand Research (SHR), its evolution (SH2R), and the Variable Stiffness Gripper equipped with a levering tool (VSG lev). NA indicates "Not Applicable".

- SH2R – actual technology;
- SH2R inj – switching the FDM technology in injection molding;
- VSG – actual technology;
- VSG inj – switching the CNC technology for aluminum parts in injection molding

For this analysis, the considered quantity slots were 1 - 9; 10 - 49; 50 - 149; 150+, as depicted in Fig. 10. The cost of injection molds is equally divided into the first ten pieces for the first slot, 50 pieces for the second, and fully amortized for higher quantities.

Fixed costs, particularly for Intellectual Proprieties (SH2R), hardware and software design drastically decrease for the second bracket, satisfying the KPI target. Moreover, for more than 50 pieces, the injection molding investment allows a further reduction of costs. The cost analysis shown in Fig. 11 gave important information for the exploitation and business analysis of the innovation developed in the project.

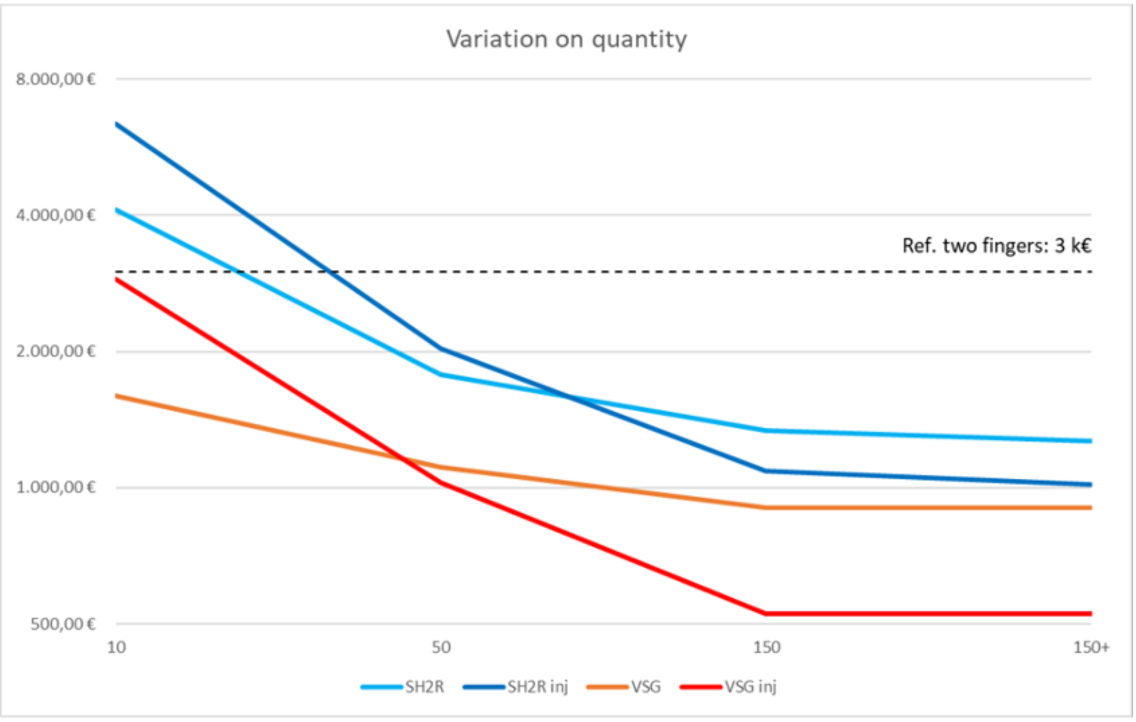
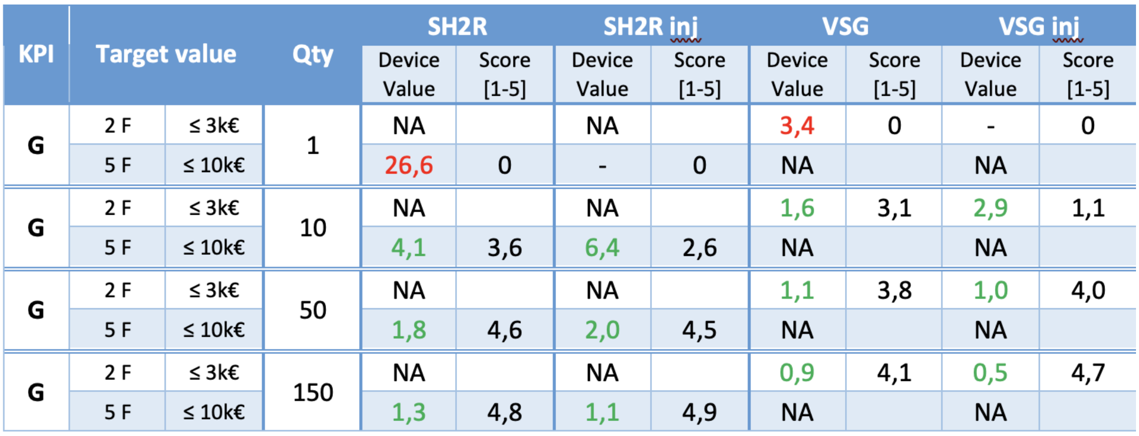


Figure 10: Cost variation depending on quantity



KPI	Target value		Qty	SH2R		SH2R inj		VSG		VSG inj	
				Device Value	Score [1-5]	Device Value	Score [1-5]	Device Value	Score [1-5]	Device Value	Score [1-5]
G	2 F	≤ 3k€	1	NA		NA		3,4	0	-	0
	5 F	≤ 10k€		26,6	0	-	0	NA		NA	
G	2 F	≤ 3k€	10	NA		NA		1,6	3,1	2,9	1,1
	5 F	≤ 10k€		4,1	3,6	6,4	2,6	NA		NA	
G	2 F	≤ 3k€	50	NA		NA		1,1	3,8	1,0	4,0
	5 F	≤ 10k€		1,8	4,6	2,0	4,5	NA		NA	
G	2 F	≤ 3k€	150	NA		NA		0,9	4,1	0,5	4,7
	5 F	≤ 10k€		1,3	4,8	1,1	4,9	NA		NA	

Figure 11: KPI (G) evaluation of two manufacturing options for each device

3.5 KPIs in WP5 for System Operation

The target KPIs for WP5 are:

- **Throughput time**: at least 100 pieces per hour.
- **Start-up time (boot time)**: less than 1 minute.
- **Set-up of a known work cell at a new location**: 90 minutes.
- **Set-up of a new work cell for new devices** (includes hardware design and manual programming): no more than two weeks.

3.5.1 Throughput time

To achieve the desired throughput of at least 100 pieces per hour in the recycling line, the cycle time should be below 36 seconds. However, the current disassembly cycle time for the devices in question, ranging from 70 to 110 seconds (see Sec. 3.1.3), falls short of the desired throughput KPI.

The implemented disassembly cycle was designed with a focus on accommodating different types of devices, rather than optimizing for large batches of similar devices with only minor variations due to damage (different conditions of devices after their end of life). In an industrial setting, where the process is expected to handle larger quantities of the same device type over a certain period, the KPI could be achieved by further optimizing the cycle time, tailoring it to the specific device example while allowing for only limited adaptation.

3.5.2 Start-up time

To start up the workcell and prepare it for use, the embedded computers as well as the external workstations and controllers must be booted. The computers are set to automatically boot when power supply is resumed. Booting can be done in parallel:

- Workstation: cca 50 sec including docker initialization.
- Embedded computers in the modules: cca. 45 sec including docker initialization.

The target startup time of less than 1 minute is thus met.

3.5.3 Set-up of a known work cell on a new location

Owing to the modular design of the software and hardware, setting up a known workcell at a new location is greatly simplified and requires only that the modules are connected to each other and integrated in the same network.

It is assumed that the modules are prepared in advance, i.e. fully assembled (the robot or peripherals are already mounted on the module). A special plug-and-play (PnP) connector module is used to connect compressed air, network, and power. The plugging procedure, along with locking the modules in place, takes about 1 minute per module. As discussed in the previous section, the modules automatically boot within 1 minute when power is applied.

After booting, various settings may need to be reconfigured, such as adjusting internet preferences to access resources outside the internal workcell network. These procedures typically take no longer than 10 minutes.

This means that the target value of the KPI (90 minutes) is met comfortably.

3.5.4 Set-up of a new work cell for new devices

This KPI depends on the workcell size, the complexity of the disassembly procedure and number of novel functionalities that have to be implemented. We assume that the set-up team includes at least one mechanical engineer, one software engineer, and one tutor for process operators.

Provided that all major hardware components are available, only smaller hardware components such as gripper fingers and fixtured need to be designed. Using rapid prototyping and 3-D printing or machining (milling), simple hardware elements can typically be designed and produced within two working days.

Another time-consuming part is the installation of the internal modules into archetypical modules with empty drawers for peripheral support. This includes setting up embedded computers, connecting GPIO pins to the solenoid valves to operate pneumatic devices, connecting air lines to peripherals, electrical wiring, installing network switches and network cables, and cable management. Deployment of the base software components to the modules with new peripheral equipment is facilitated by using docker containers. Most peripheral devices can be integrated by adjusting configuration files of ROS nodes. This process takes no longer than one working day per module.

To support new operations, skill templates and a library of convenience functions for operating peripheral devices and robots can be used as a boilerplate for programming new functionality that seamlessly integrates into the existing software architecture. Three working days should be allocated for the development, testing, and integration of each new functionality.

A medium-sized workcell with up to 6 modules and two new hardware components and two novel functionalities can be set up within a period of eleven days, exceeding the target KPI of two weeks.

References

[1] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee. *YOLACT: Real-time Instance Segmentation*. 2019. arXiv: 1904.02689 [cs.CV].

[2] *Brief summary of YOLOv8 model structure · Issue #189 · ultralytics/ultralytics*. GitHub. Jan. 10, 2023. URL: https://github.com/ultralytics/ultralytics/issues/189.

[3] DIN ISO/TS 15066:2016-02, Robots and robotic devices — Collaborative robots (ISO/TS 15066:2016).

[4] S. Haddadin and E. Shahriari. "Unified force-impedance control". In: *The International Journal of Robotics Research* (2024). DOI: 10.1177/02783649241249194.

[5] 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.

[6] 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*. 2024. arXiv: 2408.14219 [cs.RO].

[7] K. Kimble, K. Van Wyk, J. Falco, E. Messina, Y. Sun, M. Shibata, W. Uemura, and Y. Yokokohji. "Benchmarking Protocols for Evaluating Small Parts Robotic Assembly Systems". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 883–889. DOI: 10.1109/LRA.2020.2965869.

[8] R. Kirschner, K. Karacan, A. Melone, N. Mansfeld, S. Abdolshah, and S. Haddadin. "Process Fitness for the Metric-based Categorization of an Embodiment (Morphology + Control Algorithms) into the Tree of Robots". In: *under review*. 2024.

[9] R. J. Kirschner, A. Kurdas, K. Karacan, P. Junge, S. A. B. Birjandi, N. Mansfeld, S. Abdolshah, and S. Haddadin. "Towards a reference framework for tactile robot performance and safety benchmarking". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 4290–4297.

[10] *Probreg: probablistic point cloud registration library*. 2019. URL: https://probreg.readthedocs.io/en/latest/.

[11] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. *SuperGlue: Learning Feature Matching with Graph Neural Networks*. 2020. arXiv: 1911.11763 [cs.CV].

[12] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. 2022. arXiv: 2207.02696 [cs.CV].

[13] A. Zhang, K. Karacan, H. Sadeghian, Y. Wu, F. Wu, and S. Haddadin. *Tactile-Morph Skills: Energy-Based Control Meets Data-Driven Learning*. 2024. arXiv: 2408.12285 [cs.RO].