



**HAL**  
open science

## **OSCAR: a valuable asset for cost-effective automation**

Shruti Chakraborty, Lucas Labarussiat, Ferreol Gagey, Matteo Morelli,  
Raphaël Lallement, Zacharie Laloux, Francois Keith, Fabrice Mayran de  
Chamisso, Raphaël Gerin, Baptiste Gradussoff, et al.

### ► To cite this version:

Shruti Chakraborty, Lucas Labarussiat, Ferreol Gagey, Matteo Morelli, Raphaël Lallement, et al..  
OSCAR: a valuable asset for cost-effective automation. LoDiSA 2024: 2nd Workshop on Low-Cost Digital Solutions for Industrial Automation, Oct 2024, Cambridge, United Kingdom. pp.67-72, 10.1049/icp.2024.3487 . cea-04852481

**HAL Id: cea-04852481**

**<https://cea.hal.science/cea-04852481v1>**

Submitted on 20 Dec 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

## OSCAR: a valuable asset for cost-effective automation

S. Chakraborty<sup>1</sup>, L. Labarussiat<sup>1</sup>, F. Gagey<sup>1</sup>, M. Morelli<sup>1</sup>, R. Lallement<sup>1</sup>, Z. Laloux<sup>1</sup>, F. Keith<sup>1</sup>, F. Mayran de Chamisso<sup>1</sup>, R. Gerin<sup>1</sup>, B. Gradussoff<sup>1</sup>, F. Gosselin<sup>1</sup>, C. Vienne<sup>1\*</sup>

<sup>1</sup> Université Paris Saclay, CEA, LIST, Palaiseau, France

\*caroline.vienne@cea.fr

**Abstract:** In this work, we present OSCAR (Outstanding Self-adjusting Cognitive Autonomous Robot), an agile robotic platform that can be reconfigured on the fly and is capable of adapting to variations in the environment. OSCAR incorporates advanced perception, planning and action functionalities, coupled with an intuitive task-programming interface, making it a valuable asset for cost-effective automation. Its performances are demonstrated on two use cases corresponding respectively to a fine manipulation task inspired by the Robothon® Grand Challenge and an industrial application related to the dismantling of electric vehicle batteries, where we illustrate the system's ability to automatically detect and locate screws of a cover, to unscrew them, change tool and finally remove the cover.

### 1. Introduction

The growing customization of products in response to the customer's demand for personalized goods, alongside increasingly shorter product development and life cycles, as well as global crises, present manufacturing companies with the challenge of enhancing the agility and resilience of their production processes, without compromising productivity. Achieving agility and resilience in production primarily requires a high degree of flexibility and adaptability at all levels. To address this challenge, we observe the development of a new category of intelligent robotic systems, also known as autonomous industrial mobile manipulators, which integrate perception, planning, action, and adaptation capabilities required to perform various tasks in close proximity to operators. These robots can be considered strong candidates for assisting in the agile automation of manual tasks in industrial environments [1]. The Little Helper system [2], for instance, is a robot manipulator mounted on a non-holonomic mobile platform designed for tasks such as navigation, pick-and-place, and quality control in semi-structured industrial settings. This system employs a distributed software architecture with servers dedicated to each hardware subsystem, and it has been used for machine tending and interaction with visitors at trade fairs. An updated version of the Little Helper [3] incorporates an intuitive GUI and a screwing skill to assist workers in maintaining injection molds, though its capabilities are still limited for broader industrial applications. Recently, the OMNIVIL mobile manipulator [1] was deployed in a factory mock-up, performing tasks like transporting goods and preassembled parts between workstations. It features autonomous navigation, workspace monitoring, and adaptive manipulation using visual servoing, managed by a software architecture that differentiates between lower-level and higher-level tasks. Despite its ability to facilitate robot-machine collaboration, it is constrained to simple manipulation tasks involving augmented reality markers. Another noteworthy example is BAZAR, a dual-arm mobile manipulator designed for the factory of the future [4], which integrates advanced sensing and actuation technologies to support efficient navigation, dual-arm manipulation, and

human interaction. This system's modular design addresses the needs of the automotive and aeronautics industries but involves a complex, heterogeneous architecture that necessitated a bespoke standardization framework, complicating further development.

An industrial sector that particularly emphasizes this need for agility is the disassembly of end-of-life products, especially the disassembly of Electric Vehicle Batteries (EVBs) for repair or recycling. EVBs automatic dismantling has been the subject of a growing interest these last years, evidenced by a rapidly increasing number of projects and publications [5-7]. The current state of the art primarily focuses on the validation of partial functions in laboratory settings with a still low Technology Readiness Level (TRL) as reported in the latest review from 2024 [7]. Among these functions, the unscrewing and removal of small fasteners are identified as the most feasible and relevant actions to automate, while more delicate manipulation tasks generally still require human intervention.

In this work, we present OSCAR (Outstanding Self-adjusting Cognitive Autonomous Robot), an agile robotic platform tailored to industrial settings that can be reconfigured on the fly and is capable of adapting to variations in the environment. Based on the software architecture described in [8], OSCAR incorporates advanced perception, planning and action functionalities, coupled with an intuitive task-programming interface, making it a valuable asset for cost-effective automation. The structure of our paper is as follows: we describe the hardware and software architecture of the platform in Section 2 and detail the different technological bricks in Section 3. Section 4 illustrates the use of this system for two use cases dedicated to fine manipulation (UC1) and automatic unscrewing and EVB cover removing (UC2).

### 2. OSCAR platform description

#### 2.1. Hardware architecture

Featuring an UR10e robotic arm mounted on a MiR200 mobile base, the OSCAR platform combines mobility and precision for efficient handling of various parts in a large workplace (see Figure 1). For an increased

versatility, the platform is equipped with a tool changer comprising a range of essential items, such as:

- a Hand-E gripper for grasping small objects,
- a Robotiq 2F-140 gripper with enlarged jaws for grabbing larger objects (from 10 to 20 cm width),
- an e-Pick vacuum gripper offering an effective solution for catching flat parts,
- a screwdriver with a bit changer for unscrewing different types of screws and nuts, coupled with a recovery box for efficient management of spare parts.

In addition to these end effectors, the system is equipped with force and visual sensors. Thanks to a Photoneo Phoxi M camera (not visible in the picture), mounted on a dedicated pole, fast and accurate 3D scanning of the whole surrounding scene can be performed while a wrist mounted Intel RealSense D435 RGB-D camera provides advanced visual perception for detecting and locating small objects.



Figure 1: Display of the OSCAR platform equipped with a tool changer allowing the use of a screwdriver (left) or gripper (right).

## 2.2. Software architecture

The software architecture is based on the ROS2 middleware. It ensures the communication between an intuitive programming interface, the components (computer vision and robot actions) and the task planning tools. The instantiation of the general software architecture is based on a Model-Based Software Engineering platform called Papyrus for Robotic [9]. The software system uses a modular, components-based and skills-oriented architecture. Its structure comprises robotic algorithms encapsulated in modules which range from simple software blocks to more complex components, they all interact via publish/subscribe and action/client mechanisms thanks to the robotic middleware. The functionalities provided by these algorithms, called skills, are exposed to higher-level software for task and mission specifications. The Task Plan Sequencer is a critical component at the core of our architecture (see Figure 2). It takes a high-level comprehensive plan produced by the user and orchestrates it by activating the required skills in the specified sequence (see Figure 3).

Skills within our system are divided into two types: 'atomic' and 'composite'. Atomic skills bind directly to the algorithmic modules while composite skills implement a sequence of lower-level skills, which can be composite and/or atomic as well.

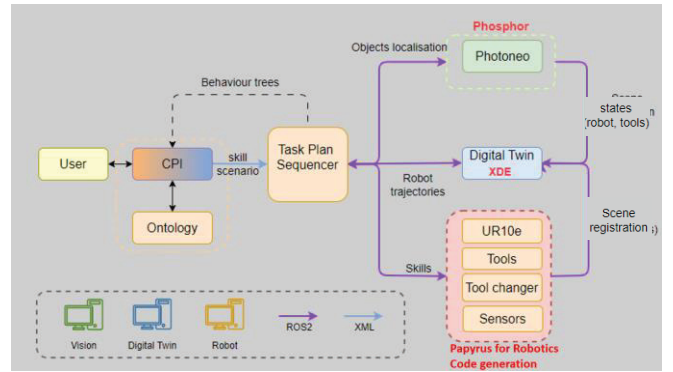


Figure 2: Simplified view of the components integrated in the software architecture.

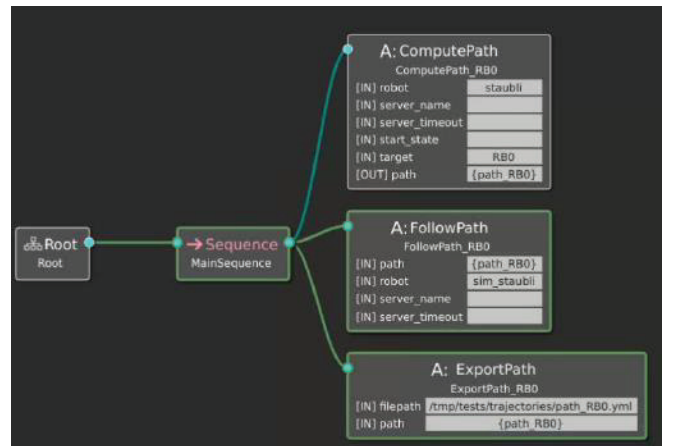


Figure 3: Behaviour tree (BT) of a trajectory-following skill.

## 3. Main components

In this section we present three of the technological bricks integrated in the software architecture as depicted in Figure 2, namely the user interface, the vision modules and the digital twin.

### 3.1. Cognitive Programming Interface

To allow for a fast adaptation to novel tasks, we have developed an intelligent task scripting tool also called Cognitive Programming Interface (CPI) [10], capable of reasoning about objects by taking into account their properties, affordances and afferences (ontologies and Semantic Web Rule Language). The tool proposes a list of available actions based on the objects and robots present in the workspace (scene), enabling the user to easily construct a sequence of actions to be carried out. The list of actions is updated after each selection, and management of the scene ontology guarantees the feasibility of the scenario. This reasoning layer is integrated into an intuitive graphical user interface (GUI), which relies on the digital twin of the scene.

This scenario writing functionality (see Figure 4) exploits the semantic representation of the scene. When the user selects an object from the 3D view or the Scene tab (top), the GUI automatically provides the available skills associated with the selected object, allowing for easy scenario building. The Scenario tab (bottom) allows the user to view and edit the complete sequence.

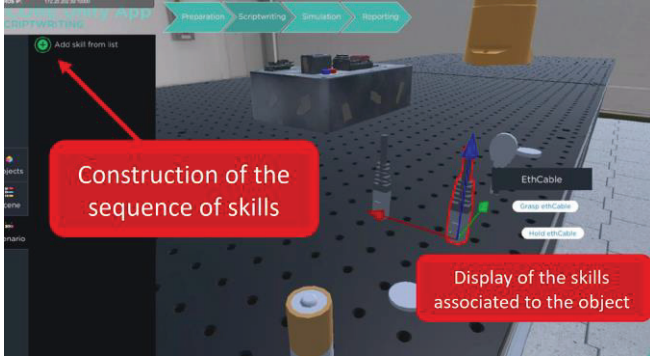
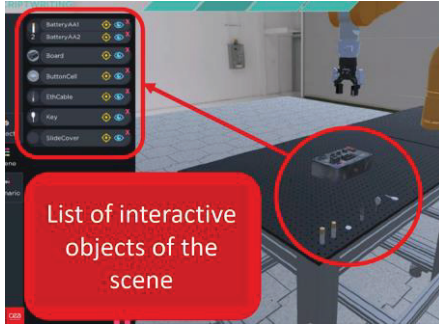


Figure 4: Display of the GUI allowing the user to define the sequence of actions and ensuring the consistency of the final sequence.

### 3.2. Visual Perception

The Photoneo sensor provides a 3D point cloud of the scanned area (see Figure 5), which is used to compute the pose of objects whose CAD models are known, relative to a fixed reference frame. The 6D registration algorithm is a point cloud descriptor-based registration pipeline [11] with special emphasis on ambiguous and symmetrical industrial objects. This approach does not use deep learning and does not require a prior dataset or training. Pose hypotheses are selected in Hough space, refined using a point-to-point iterative closest point approach, and scored using the proximity between a render of the registered object and the observed scene point cloud. This whole process is GPU-accelerated and requires less than 0.5s to localize a given object in the point cloud with a submillimetric precision.



Figure 5: Illustration of a Photoneo scan used to locate objects whose CAD models are known.

Visual perception is also used in a dedicated skill for the detection and localization of small objects using a RGB-D camera mounted on the robot arm and whose extrinsic and intrinsic parameters have been calibrated. This visual skill has been deployed in our case more specifically for screw detection, where we fine-tuned a YOLOv5 model [12], pre-trained on the COCO dataset, to extract screw locations in an image. Triangulation is then performed using multiple 2D

detections in frames acquired from different points of view. A database of 3D positions of screws together with their type (hex, torx, Philips, etc.) is maintained and shared with the digital twin for robot movement planning.

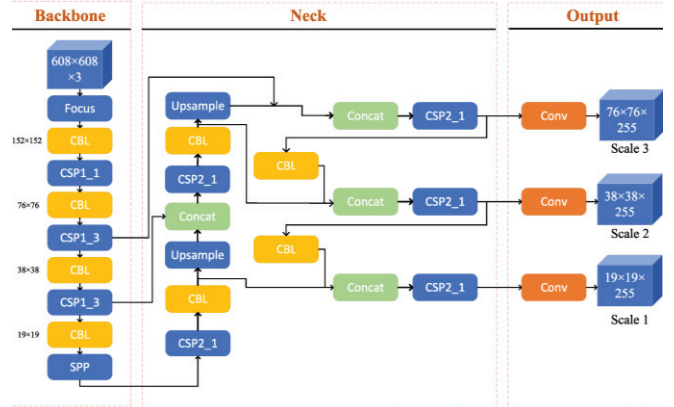


Figure 6: YOLOv5 network consists of three parts: backbone part for feature extraction, neck part for feature fusion and output part for object detection [12].

### 3.3. Digital twin

The Digital Twin uses Unity3D as graphics engine and XDE (eXtended Dynamics Engine [13]) as physics engine. XDE is computationally efficient in the physically realistic simulation of the dynamics of rigid multibody mechanical systems, such as robot arms, in interaction with their environment (e.g. gravity, contact, friction forces...). It allows computing the evolution of the system in interactive time, with access to a large range of data relative to the robots (velocities, torques, forces, and swept volumes). The models are imported from Solidworks with CAD plugins such as PIXYZ or CAD Exchanger, allowing to use them easily for graphical rendering as well as collision computation. The physical properties of the robots can be extracted from these CAD data, or from URDF (Unified Robot Description Format) files. XDE can also work with point clouds, allowing to update the world model at each step with the data provided by the 3D camera.

Beyond allowing dynamical simulation of the behaviour of the robot using numerical models, our Digital Twin makes it possible to directly link the robot model with its real controller. This functionality is used to run concurrently the robot and its model for monitoring purposes. The path planning is computed through this Digital Twin, using a RRT (i.e. Rapidly-Exploring Random Tree) algorithm with collision avoidance in the simulator, based on the Open Motion Planning Library [14]. The computations are made in the joint space and XDE is used to compute the pose of the robot for each configuration proposed and detect potential interferences with its environment.

## 4. Results

In this Section, we illustrate the use of OSCAR on two use cases. In the first example (UC1), we show how the graphical interface is used to write a task sequence and how this sequence is carried out with the real robot. In a second industrial use case (UC2), we illustrate the system's ability to handle a complex disassembly task involving automatic detection and recognition of screws, unscrewing and removing a cover.

#### 4.1. Application to a fine manipulation task (UC1)

The robotic system was first deployed to address a use-case inspired from the Robothon® Grand Challenge, an international competition and benchmarking event for measuring state-of-the-art performance in robot manipulation on tasks representative of electronics waste management [15]. The challenge consisted in carrying out a series of robotic tasks on a board provided by the organisers [16]. After randomly positioning this board in the robot's workspace, the robotic system had to act autonomously to perform a set of six different tasks.

In the 2022 edition of the challenge, that supported the development of our interface, the proposed tasks were: locate the board and press a button (T1), pick up a key on the task board and activate a keyswitch (T2), move an Ethernet plug from one port to another (T3), open a battery box, remove 2 AA batteries and place them in dedicated holes (T4), remove a CR2032 battery from its receptacle (T5), press a red button on the board to stop the chronometer (T6).

#### 4.2. Implementation and results of UC1

Unlike the traditional approach to the challenge, which involves performing actions as quickly as possible in a given order, we aim to provide the operator with a graphical interface that allows the definition of the sequence of actions. Once this sequence is defined, it is transmitted to the Task Plan Sequencer, which will then request the digital twin of the scene to determine the robotic trajectories needed to perform the various actions without collision. Since all tasks are physically linked to the board, an initial step of locating the board using the Photoneo sensor is sufficient to register the reference frame of the board in the digital twin.

**4.2.1. Sequence definition:** Based on the scene ontology, the graphical interface displays all the objects with which the robot can interact, such as the key, the Ethernet cable, the batteries, and the buttons on the board. By clicking on these various objects, the user can see the possible actions (see Figure 7) and thereby select the appropriate one to create the sequence.

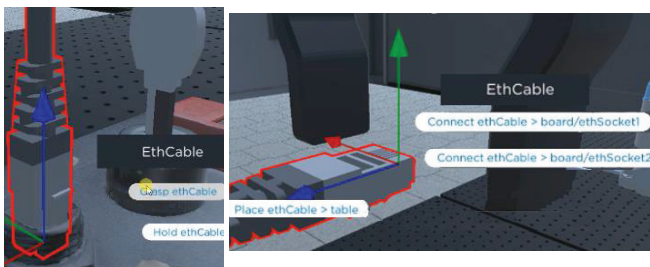


Figure 7: Illustration of the GUI integrating the ontology to guide the user in the definition of the sequence and ensure its feasibility.

The reasoning module integrated into the CPI ensures the coherence of the sequence defined by the user, which can be visualized in the simulation environment before being transferred to the orchestrator, for execution with the real robot (see Figure 8).

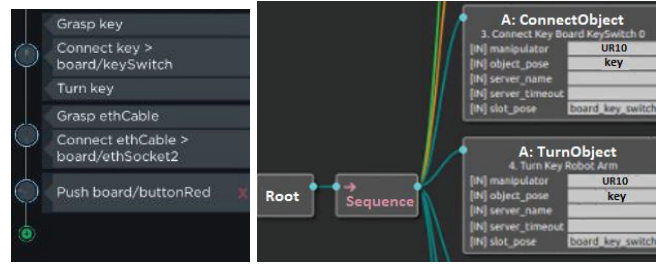


Figure 8: Display of the sequence in the GUI (left) and extract of one part of the corresponding BT (right).

**4.2.2. Sequence realization:** To compute the trajectories using the system's digital twin, it is first necessary to locate the various elements of the scene relative to the robot's reference frame. This localization step is systematically carried out at the start of the sequence using a scan from the Photoneo sensor, which has been previously calibrated with respect to the robotic end-effector through an eye-to-hand calibration procedure. In our case, only the base frame of the board is located, from which the positions of all interaction elements are inferred. Finally, the sequence is smoothly realized by the robot (see Figure 9).



Figure 9: Illustration of the system performing the defined tasks: inserting the key (left), grasping the Ethernet cable (centre) and pushing the red button (right).

#### 4.3. Application to a disassembly task (UC2)

As mentioned in the introduction, one emblematic use case we target is the repair or recycling of EVBs, where the versatility of the OSCAR platform allows it to cope with the diversity of products and disassembly operations. The strategy we consider is twofold. When the battery pack is still closed, the system automatically detects, recognizes, and locates the screws positioned on the cover. Then, it computes the robot's positioning relative to the pack and autonomously performs the unscrewing tasks under the supervision of an operator. After the cover is removed, the operator will be able to load a standard battery scene into the intuitive programming interface and load or create the disassembly sequence.

In the current work, we focus only on the first part of the disassembly scenario and illustrate the behaviour of the robotic platform on a box with a screwed lid (see Figure 11, left). The results are presented for the following three steps:

- 1) the detection and localization of screws,
- 2) the unscrewing, and
- 3) the lid removal.

#### 4.4. Implementation and results of UC2

In this section we present the dataset we have constituted to fine tune the YOLO architecture described in the methodology section, and the general strategy we consider to achieve the unscrewing and lid removal tasks.

**4.4.1. Visual detection:** The custom dataset consists of 178 images of hexagonal screws, which underwent several post-processing steps to enhance diversity. These steps include horizontal flipping, rotation, 5% zoom, and lighting adjustments of  $\pm 15\%$ . As a result of these augmentations, the dataset was expanded to include 744 images for training, 71 images for validation, and 35 images for testing (see Figure 10). The fine-tuned YOLOv5 model achieved a precision of 98.4% and recall of 99.3% on the validation set.

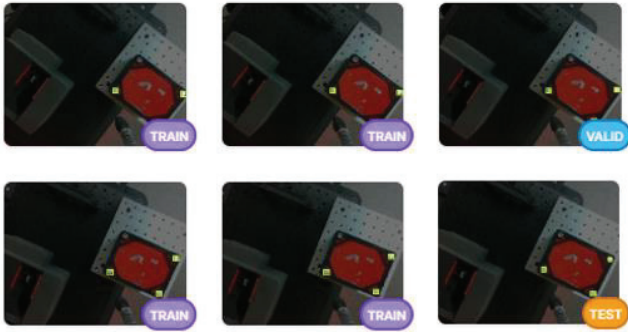


Figure 10: Illustration of the labelled dataset used for train / test and validation of the YOLO neural network.

**4.4.2. Unscrewing task:** For performing the unscrewing task, the robot begins by moving to a safe home position. Then, screws are detected via the embedded camera, calibrated in eye-in-hand configuration, and the trained YOLOv5 network (see Figure 11).

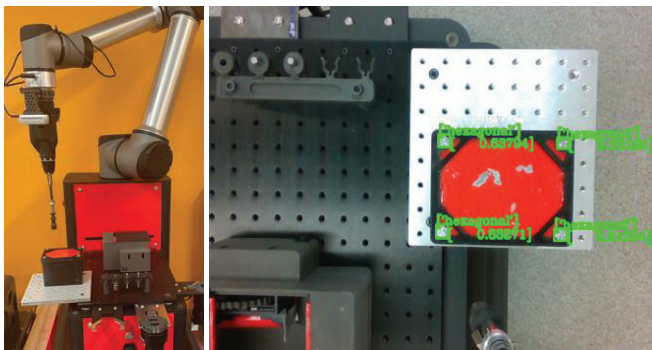


Figure 11: Left: overview of the robotic system with the box. Right: automatic screw detection in the embedded camera frame. The type of screw is also detected (“hexagonal”), allowing for the automatic selection of the corresponding bit for the screwdriver.

Detected screw centre positions are projected in the base frame of the robot and saved in a database. The BT of the unscrewing sequence is given in Figure 12. For each detected screw, the “unscrew” and “deposit screw” skill sequences are executed. These are composite skills that are in turn defined through specific BT. In the unscrew skill (see Figure 13), the 3D centre position of the screw is used to plan a Cartesian trajectory to place the screwdriver at a given offset above the screw. Force control along the z axis in the Cartesian space is then applied to engage with the screw head.

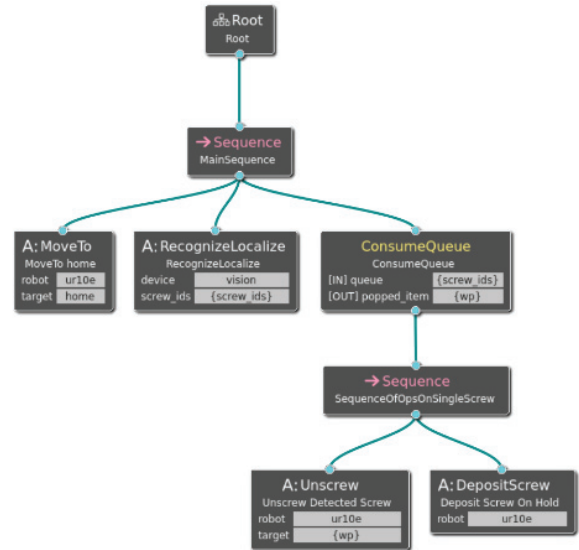


Figure 12: Behaviour Tree for the global unscrewing task.

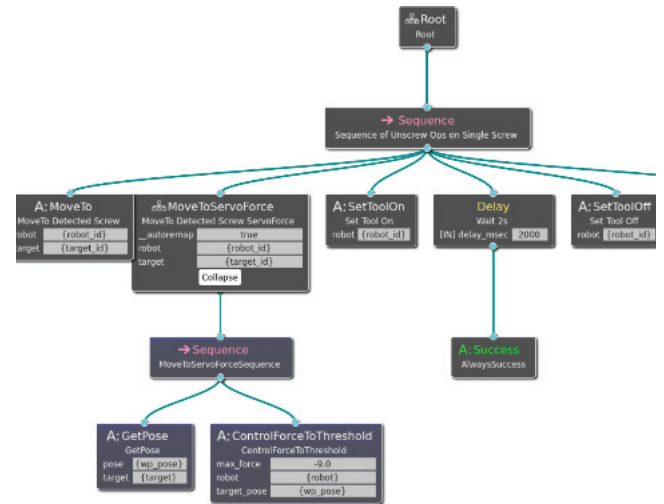


Figure 13: Behaviour Tree for the composite unscrewing skill. For display purpose we do not show the last atomic skill (on the right of the BT), which is another MoveTo skill allowing to reach the disengage position after unscrewing the screw.

To ensure the screw remains in the effector after unscrewing, we have integrated a magnet into the electric screwdriver. Consequently, it is necessary to remove the screw manually from the screwdriver before unscrewing a new screw. For this purpose, an actuated box has been made with a removable lid that clamps the screw and holds it when the tool is lifted, thus freeing it (see Figure 14).

**4.4.3. Lid removal task:** When all the detected screws have been unscrewed and removed, the robot changes tool for the adequate gripper, to grasp the box lid. The centre of the lid, used to define the gripping area, is estimated based on the location of the four screws in the initial image of the box (Figure 11). Figure 14 presents two views of the robotic system performing the task autonomously. In the left view, the electric screwdriver integrated into the robot is shown with an interchangeable bit allowing adaptation to different categories of screws. A spring mechanism provides additional compliance to the force control strategy, allowing the compensation of localization inaccuracies resulting from visual perception. This compliance ensures the efficiency of the unscrewing process (100% success rate over approximately 20 trials), with the bit clipping onto the screw

when the tool is rotated. In the right view, the robot is shown after the unscrewing and tool-changing phase, removing the lid from the box to place it beside the box.

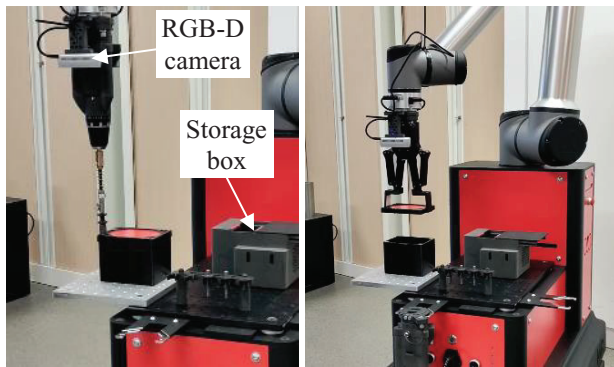


Figure 14: View of the robotic setup during the unscrewing task (left) and removing the cover of the box (right).

## 5. Conclusion

The potential benefits of automating EVBs disassembly are manifold. Automation can significantly reduce the time and cost associated with battery maintenance and recycling, improving overall efficiency. Moreover, it can enhance workers' safety by minimizing direct human interaction with potentially hazardous materials. However, the complexity of EVBs' designs, with their varied architectures and fastener types, presents a substantial challenge for developing a universally adaptable robotic solution. Recent advancements in robotics, artificial intelligence, and sensor technologies offer promising avenues to address this type of tasks. In this work, we have presented OSCAR, a robotic platform dedicated to these challenges, which integrates different tools and sensors, a user-friendly interface as well as perception, simulation and planning capabilities. OSCAR has already been used to address two different use cases of fine manipulation and dismantling and will be deployed in the future in industrial environments to address real cases, such as EVBs disassembly.

## 6. Acknowledgments

This work was supported by the European project euROBIN under Grant Agreement 101070596 and by the European project TraceBot under Grant Agreement 101017089.

## References

- [1] H. Engemann, S. Du, S. Kallweit, P. Cönen, H. Dawar. 'OMNIVIL - An Autonomous Mobile Manipulator for Flexible Production', *Sensors*, 20(24), 2020.
- [2] M. Hvilshøj, S. Bøgh, 'Little Helper - An Autonomous Industrial Mobile Manipulator Concept', *Int. Journal of Advanced Robotic Systems*, 8, 2011, pp. 80–90.
- [3] P.J. Koch, M.K. van Amstel, P. Dębska, M. A. Thormann, A. J. Tetzlaff, S. Bøgh, D. Chrysostomou, 'A Skill-based Robot Co-worker for Industrial Maintenance Tasks', *Procedia Manufacturing*, 11, 2017, pp. 83-90.
- [4] A. Cherubini, R. Passama, B. Navarro, M. Sorour, A. Khelloufi, O. Mazhar, S. Tarbouriech, J. Zhu, O. Tempier, A. Crosnier, P. Fraisse, S. Ramdani, 'A collaborative robot for the factory of the future: BAZAR', *The International Journal of Advanced Manufacturing Technology*, 105(9), 2019, pp. 3643-3659.
- [5] Meng K, Xu G, Peng X, Youcef-Toumi K, Li J. Intelligent disassembly of electric-vehicle batteries: a forward-looking overview. *Resour Conserv Recy* 2022;182:106207.
- [6] M. Beghi, F. Braghin, L. Roveda, Enhancing Disassembly Practices for Electric Vehicle Battery Packs: A Narrative Comprehensive Review. *Designs* 2023, 7, 109.
- [7] T. Kaarlela, E. Villagrossi, A. Rastegarpanah, A. San-Miguel-Tello, T. Pitkäaho, Robotised disassembly of electric vehicle batteries: A systematic literature review. *Journal of Manufacturing Systems*, vol. 74, 2024, Pages 901–921.
- [8] F. Gosselin, G. Acher, B. Gradoussoff, S.Kchir, F. Keith, O. Lebec, C. Louison, B. Luvison, F. Mayran de Chamisso, B. Meden, V. Molina, M. Morelli, J. Rabarisoa, C. Vienne, G. Ameyugo, 'An Intelligent Robotics Modular Architecture for Easy Adaptation to Novel Tasks and Applications', in *IEEE 19th Int. Conf. on Automation Science and Engineering (CASE)*, Auckland, New Zealand, 2023.
- [9] 'Papyrus For Robotics Repository', <https://eclipse.dev/papyrus/components/robotics>, accessed 19 July 2024
- [10] R. Gerin, J. Dumora, O. David, B. Gradoussoff, 'Cognitive Programming Interface: from Task Level Programming to Coherent Task Level Programming', in *IEEE 20th Int. Conf. on Automation Science and Engineering (CASE)*, Bari, Italy, August 2024.
- [11] R.B. Rusu, N. Blodow, M. Beetz, 'Fast Point Feature Histograms (FPFH) for 3D registration', in *Proc. IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, May 2009.
- [12] L. Zhu, X. Geng, Z. Li, C. Liu, Chun. 'Improving YOLOv5 with Attention Mechanism for Detecting Boulders from Planetary Images'. *Remote Sensing*. 13. 3776, 2021.
- [13] X. Merlhlot, J. L. Garrec, G. Saupin et C. Andriot, 'The XDE Mechanical Kernel: Efficient and Robust Simulation of Multibody Dynamics with Intermittent Nonsmooth Contacts', *The 2nd Joint International Conference on Multibody System Dynamics*, 2012.
- [14] I.A. Şucan, M. Moll, L.E. Kavraki, 'The Open Motion Planning Library', *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. <https://ompl.kavrakilab.org>
- [15] 'Robothon Grand Challenge', <https://robothon-grand-challenge.com>, accessed 19 July 2024
- [16] P. So, A. Sarabakha, F. Wu, U. Culha, F. Abu-Dakka, S. Haddadin. 'Digital Robot Judge (DR.J): Building a Task-Centric Performance Database of Real-World Manipulation with Electronic Task Boards', *IEEE Robotics & Automation Magazine*, 2-14, 2023.