A Distributed Control Architecture for a Multi-Agent Robotic Cell: A Battery Pack Disassembly Case Study

Michele Ferrari[a,b,†] and Roberto Fausti[b,†] and Cesare Tonola[b,*] and Michele Delledonne[c] and Samuele Sandrini[b] and Manuel Beschi[a,b] and Enrico Villagrossi[b]

[a]Department of Mechanical and Industrial Engineering, University of Brescia, Via Branze 42, 25123, Brescia, Email: name.surname@unibs.it
[b]Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council of Italy, Via A. Corti 12, 20133, Milan, Email: name.surname@cnr.it
[c]Inxpect S.p.A., Via Serpente 91, 25131 Brescia, Email: name.surname@inxpect.com

ABSTRACT
Disassembly in industrial settings is a complex process that requires diverse hardware, robots, and sensors. Effective collaboration between robots and human operators is crucial for enhancing flexibility. Although research has explored aspects like motion planning and multi-robot coordination, a comprehensive architectural solution is still lacking. This article proposes a decentralised software architecture that integrates various devices, robots, tools, sensors, vision systems, and safety mechanisms. It supports multi-robot operations and is compatible with different hardware and protocols. The architecture allows real-time control operations to coexist with more intensive computational tasks. Additionally, the paper offers a qualitative comparison with existing solutions. The proposed architecture has been validated in a case study focused on dismantling end-of-life automotive battery packs, demonstrating its adaptability, safety, and low-code programming in disassembly processes.

KEYWORDS
Distributed Control Architecture, Robotic Disassembly, Battery Pack Disassembly, Multi-Robot Cooperation, Multi-Sensor Integration, Hybrid Digital Twin

1. Introduction

Despite an increased interest in reuse, remanufacturing, and recycling, disassembly has not been a primary focus of mainstream robotics research. Disassembly processes, often involving the largest number of employees and the highest complexity in End-of-Life (EoL) treatments, are inherently more challenging than general assembly tasks due to product complexity, design issues[1], and variability. These complications are particularly evident when non-destructive disassembly is required to preserve component integrity for remanufacturing and second-life applications purposes (Priyono, Ijomah, and Bititci 2015). At the same time, even in the case of recycling, the extraction of value-added parts requires dismantling activities. The drive to automate disassembly processes stems from their current inefficiency, economic sustainability, and the operators' safety and

---

† Authors contributed equally.

* Corresponding author: C. Tonola. Email: cesare.tonola@cnr.it.

[1] Devices not designed according to the *Intelligent Design for Disassembly* paradigm (Lander et al. 2023; Meng et al. 2022).

well-being (Lu, Pei, and Peng 2023). Researchers have attempted to automate the disassembly of highly complex products, such as cars and mechanical components, but none have achieved industrial-scale implementation yet. Current automated disassembly plants are highly specialised and tailored to specific devices and models (Lu, Pei, and Peng 2023); despite this, some activities remain manual, and fully automated systems with current technologies are still far from reaching industrial maturity (Poschmann, Brüggemann, and Goldmann 2020; Tan et al. 2021). This situation underscores the need to advance robotic disassembly technologies to enhance the efficiency and economic viability of reusing, remanufacturing, and recycling EoL products.

A particularly relevant and emerging use case in EoL product management is the dismantling of Electric Vehicle Batteries (EVBs), which has become a critical issue following the rapid expansion of the electric vehicle market. In 2023, 18% of new cars sold globally were electric[2] bringing the total number on the road to 40 million (IEA 2024; Ritchie 2024). This impressive growth results in a substantial burden of EVBs, posing significant challenges in Battery Pack (BP) manual sorting, testing, and disassembly for reuse, remanufacturing, and recycling (Kaarlela et al. 2024).

Addressing challenges in robotic battery pack disassembly

BP disassembly presents hazards for operators, yet it still heavily relies on manual labour. This process is industrially challenging due to the complexity of the systems and the precision required to handle various mechanical components. Full automation is costly and technically demanding, especially when dealing with tasks that require manipulation dexterity and variable workpieces. Additionally, automated systems for pack-to-module[3] disassembly are still far from being introduced. Vehicle design priorities like crash safety and space optimisation often conflict with remanufacturability considerations. As a result, designs are not optimised for disassembly, posing a major barrier to robotic automation.

Traditional automation systems are optimised for structured manufacturing environments, where tasks are highly repetitive and objects are fixed in well-defined positions. In contrast, disassembly tasks are inherently unstructured and variable, requiring robotic systems to adapt to uncertain conditions and non-uniform components (Harper et al. 2019; Villagrossi et al. 2023). Moreover, as automation levels increase, managing external uncertainties in EoL product disassembly becomes more complex (Foo, Kara, and Pagnucco 2022).

Addressing these challenges requires a flexible system capable of handling arbitrarily shaped objects in semi-structured environments, supporting fully and semi-automatic operations. This demands the development of advanced control algorithms that enable robotic hardware to operate intelligently and adaptively. Moreover, it necessitates integrating cutting-edge perception technologies, such as three-dimensional RGB-D imaging and specialised sensors, to manage the dynamic interactions and complexities involved. Multi-robot cooperation is crucial, as each robot has different payloads and mounts different sensors and tools. Thus, it could specialise in a subset of tasks. However, while multi-robot systems can handle the complexity of demanding disassembly tasks (*e.g.,* removing a battery pack cover or monitoring the removal of battery cells), some tasks still require human intervention. Several studies suggest exploring human-robot co-working

---

[2] Including battery electric vehicles and plug-in hybrid vehicles.

[3] EVB packs are arranged into multiple modules consisting of individual battery cells with the so-called pack-module-cell architecture. The pack-cell architecture has recently emerged, where cells are placed directly into the battery pack without using modules.

using force-sensitive lightweight robots (Wegener et al. 2015; Li et al. 2024). Thus, ensuring a safe and efficient workspace sharing between humans and robots is essential for harnessing both strengths. This requires the heterogeneous use of different sensors and vision systems, not only for the localisation and inspection of mechanical parts but also to ensure operator safety. In this highly uncertain setting, simulation becomes invaluable for offline testing and online monitoring within a replicated manufacturing environment (Adams et al. 2022; Kousi et al. 2019). The ability to test procedures and control algorithms before executing them on real hardware can save significant time and, most importantly, ensure the safety of operators and equipment. Integrating sensor readings into the simulated replica allows for feedback that can be used to fine-tune control and perform error correction.

Contribution

This article presents a modular and distributed software architecture designed to address the inherent complexity of industrial disassembly processes, focusing on enabling flexible disassembly applications. The proposed architecture allows effective coordination among heterogeneous components, including robot manipulators, mobile platforms, vision systems, sensors, and safety devices through an abstraction layer that supports integration across various industrial protocols and hardware interfaces. It enables atomic task decomposition through a Behavior Tree-based plan execution model (Iovino et al. 2022), supporting real-time motion planning and execution for industrial robots using *open controllers*[4] and micro-interpolated trajectories. Moreover, essential safety modules are integrated to ensure safe human-robot coexistence in complex environments.

The architecture is validated on a real-world application of a battery pack disassembly. This use case demonstrates the architecture's versatility and robustness by integrating industrial hardware and protocols to enable state-of-the-art motion planning, control, vision-based localisation and servoing, and coordination algorithms. Notably, the architecture enables the following:

- Integration with industrial servo drives and PLC I/O modules using widespread industrial communication protocols such as EtherCAT, CANopen over EtherCAT (CoE), and Modbus.
- Sensor integration through various communication protocols, including MQTT and OPC UA.
- Force-feedback control via wrist-mounted sensors.
- Vision-based localisation and visual servoing using AI-powered 3D vision systems.
- Support for hybrid digital twin simulation environments for early-stage validation (Rasheed, San, and Kvamsdal 2020).

Beyond the considered use case, the proposed architecture allows straightforward adaptation to other scenarios involving similar assembly or disassembly processes.

The paper is organised as follows. Section 2 presents a brief literature review on semi-automatic disassembly. Section 3 introduces a general control architecture and highlights design choices that make it stand out compared to other solutions. Section 4 details the specific implementation tailored to battery pack disassembly regarding hardware components, software libraries, communication protocols, and safety modules. Section 5 provides a critical analysis, discussing key insights related to network infrastructure,

---

[4] Open controllers allow direct control of the robot by sending position, speed, or force references to low-level control layers.

industrial hardware integration, fault management strategies, and safety considerations for human workers. Finally, Section 6 summarises the main findings, presents conclusions, and outlines directions for future work.

2. Related works

Remanufacturing processes promote non-destructive disassembly, and robotics is one of the most suitable technologies to address product variability and perform such tasks effectively. Robotic disassembly systems often rely on multi-robot configurations, Human-Robot Collaboration (HRC) (Hjorth and Chrysostomou 2022), AI (Meng et al. 2022), perception systems or a combination of these to effectively address external factors and achieve the required flexibility (Yin, Xiao, and Wang 2022). The typical flow for automated dismantling is device data acquisition (*i.e.,* acquiring information from the database, capturing 2D/3D images, etc.), detecting components, identifying them, determining their positions, defining a disassembly plan, and removing the components (Beghi, Braghin, and Roveda 2023).

Integrating diverse software modules and a heterogeneous network of sensors into a unified industrial control architecture to accomplish each task lacks out-of-the-box solutions. This challenge is particularly evident in low-level robot control. Most industrial robots are still managed by PLCs (Nakagawa et al. 2021), which are unsuitable for integrating advanced control algorithms and sensors. Consequently, industrial applications remain rigid, relying on customised solutions tailored to specific control modules.

Robotic research has adopted the Robot Operating System (ROS) (Quigley et al. 2009; Macenski et al. 2022) as the go-to development platform, quickly becoming a *de facto* standard to overcome the already mentioned limitations (Simonič et al. 2021). With its ecosystem, ROS made possible a fast deployment of complex applications combining low-level motion control (including real-time), multi-robot motion planning, advanced algorithms, easy sensor integration, and simulation environments (Malavolta et al. 2020; Albonico et al. 2023). ROS also allows easy deployment of simulations and digital twins (Mattila et al. 2022), providing seamless integration with the RViz visualisation software (OSRF 2024c) and several simulators such as Gazebo (OSRF 2024a). Despite its success in research, ROS has not yet been widely adopted in the industry. ROS 2 aims to address this by providing industrial-grade real-time middleware with Quality of Service (QoS) and security features.

Concerning BP robotic disassembly, the literature proposes several works on the topic (Meng et al. 2022; Li et al. 2024). However, existing research primarily concentrates on laboratory experiments without concrete real-world applications and sufficient safety investigation (Tan et al. 2021; Kaarlela et al. 2024). Previous research proposed semi-autonomous solutions conceptualising BP disassembly, either HRC applications or a multi-robot approach (Wegener et al. 2014; Blankemeyer et al. 2021; Hellmuth, DiFilippo, and Jouaneh 2021), demonstrating the technical and economic feasibility (Yuan et al. 2023). Lightweight robots were used intensively without considering the actual size and weight of the BP modules, which are not manipulable with low-payload robots (Villagrossi and Dinon 2023). As laboratories lack the required safety monitoring and fire prevention systems, and the hardware used is not industrial grade, the presented solutions remain between TRLs 2–4 (Kaarlela et al. 2024), leaving room for research activities and technology transfer to industry (Tan et al. 2021).

A comparison of control architectures for robotic disassembly found in the literature is presented in Table 1. The classification draws inspiration from the taxonomy pro-

Table 1. Comparison of robotic control architectures for disassembly tasks, considering robot setup (single/multi-robot), composition (same or different robot models), robot types (manipulator, mobile robot, or other actuated systems), coordination strategies, architecture type (distributed or centralised), module integration (*e.g.,* sensing, planning), inter-agent communication, collaboration level, human involvement, real-world implementation, and application.

Classification	Robot Setup		Composition		Robot Type				Coordination		Architecture Type		Module Integration				Inter-Agent Communication		Collaboration		Human Involvement		Implementation		Application	
Reference	Single Robot	Multi Robot	Homogeneous	Heterogeneous	Manipulator	Mobile Robot	Mobile Manipulator	Other	Coordinated	Non-Coordinated	Distributed	Centralised	Tools/Sensing	Control	Motion Planning	Task Planning	Communicating	Non-Communicating	Cooperative	Competitive	Human-in-the-loop	Fully autonomous	Real-World	Simulation	Battery Disassembly	Other
Tan et al. (2021)	✓		✓		✓							✓	✓								✓			✓	✓	
Wang et al. (2023)	✓		✓		✓							✓	✓	✓	✓	✓						✓		✓	✓	
Erdogan et al. (2024)		✓	✓		✓			✓	✓			✓	✓		✓	✓	✓		✓			✓		✓	✓	
Qu et al. (2024)		✓		✓	✓				✓			✓	✓	✓			✓		✓			✓	✓	✓	✓	
Viso et al. (2024)	✓			✓	✓							✓	✓	✓	✓							✓	✓	✓		✓
Dimosthenopoulos et al. (2024)		✓		✓	✓				✓			✓	✓	✓	✓	✓	✓		✓			✓	✓	✓		✓
Yumbla et al. (2025)		✓		✓		✓			✓			✓			✓	✓	✓		✓			✓		✓		✓
Our work		✓		✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓		✓		✓	✓	✓	

posed in (Verma and Ranga 2021). Specifically, Table 1 categorises the reviewed works according to the following criteria:

- **Robot Setup**: refers to the number of robots employed in the disassembly process. Multi-robot systems can better handle the variability and complexity of BP disassembly and perform multiple parallel tasks. However, they are more complex to manage. In contrast, single-robot setups are simpler and more cost-effective to deploy, but may be constrained by the size and weight of the BP, and cannot parallelise tasks.
- **Composition**: indicates whether the architecture supports robots from different brands and models. Heterogeneous architectures are designed to integrate diverse hardware and software, enabling task specialisation by leveraging each robot's unique capabilities, such as payload or dexterity. In contrast, homogeneous architectures typically support only identical robots, often limited to the same brand or model, simplifying integration but reducing flexibility and adaptability.
- **Robot Type**: specifies the categories of robots employed, including manipulators, mobile platforms, and other robotic systems (*e.g.,* external linear axes or custom-built platforms). While manipulators are typically essential for executing disassembly tasks, auxiliary operations such as inspection or object recognition via vision systems can be enhanced by integrating mobile robots or specialised equipment.
- **Coordination**: robots can operate in a coordinated or uncoordinated manner. Coordination does not necessarily require robots to be aware of each other; it can also be achieved through a central system or by sharing information via shared memory or variables. However, coordination is not always required—some flexible systems allow multiple robots to access existing resources independently. While uncoordinated systems are generally simpler to implement, they may lead to inefficient resource usage or conflicts.
- **Architecture Type**: distributed architectures share computational load across multiple processing units, enabling modular handling of distinct functional tasks.

While this increases scalability and fault tolerance, it also introduces added complexity in system design, deployment, and maintenance.

- **Module integration**: highlights whether the architecture includes dedicated modules or approaches for advanced robot control, task execution, motion and task planning, or incorporates specialised tools and sensors to enhance the disassembly process. These features are critical for leveraging sophisticated algorithms to improve performance and to ensure flexibility in adapting to unpredictable or complex scenarios.
- **Communication**: refers to the ability of robots to share information. In the absence of communication, coordination is limited or nonexistent. Effective communication enables synchronised behaviour and more efficient task allocation among robots.
- **Collaboration**: assesses whether robots cooperate toward a shared goal or operate independently. In disassembly tasks, collaboration typically involves robots working together to maximise the overall efficiency and effectiveness of the process.
- **Human involvement**: despite significant automation, certain disassembly tasks still require human intervention due to their complexity or variability—tasks which may be too time-consuming or difficult for robots to perform autonomously. The architecture must incorporate appropriate safety measures for safe human-robot interaction in such cases. Fully autonomous systems, by contrast, restrict human access to the operational cell during execution.
- **Implementation**: indicates whether the architecture has been developed solely in simulation or also deployed on real hardware.
- **Application**: indicates whether the architecture has been implemented for battery pack disassembly processes or other operations (*e.g.,* pick-and-place, assembly/disassembly of different products).

As shown in Table 1, to the best of the authors' knowledge, the literature lacks a comprehensive decentralised multi-robot architecture that simultaneously integrates heterogeneous robots, supports advanced planning and control algorithms, incorporates custom sensing and tooling, and enables seamless collaboration between robots and human operators during disassembly. To address this gap, the following sections outline the requirements for such an architecture and introduce the proposed solution.

3. Proposed architecture

The following requirements outline the key attributes that a control architecture must possess based on the previous discussion to ensure both flexibility and efficiency in disassembly operations:

- *Decentralised architecture*: given the potential computational demands that could overwhelm a single machine, the architecture should enable multiple processing units to collaborate within a distributed framework.
- *Support for multi-robot configurations*: the disassembly task can demand multiple robots (*e.g.,* many different operations can be executed on the same object simultaneously).
- *Support for real-time control*: *open control* requires real-time communication with the low-level controllers of actuated systems, such as robots and linear axes.
- *Robot brand agnosticism*: the architecture should seamlessly integrate robots from

any manufacturer, enabling unified programming across different systems from a single platform. This eliminates the need for proprietary programming languages and simplifies development.

- *Support for a wide variety of sensors*: to address all operational aspects, including safety, vision systems, and hazard control, the architecture should support a broad spectrum of sensors.
- *Advanced motion planning capabilities*: fixed trajectories lack the flexibility needed to accommodate different disassembly items and positions. Therefore, advanced motion planning is crucial for enabling real-time, collision-free trajectory generation that adheres to both the robot's kinematics and the task-related requirements.
- *Plan execution model*: the system must provide robust high-level task execution strategies to enable effective concurrent agent management and effective fault recovery.
- *Digital twin and simulation*: employing a digital twin of the entire work cell facilitates design and testing of control algorithms, motion, and task planning in a safe, virtual environment. Additionally, a hybrid digital twin can integrate real-time data from the physical cell, enabling more accurate testing and optimisation of the control architecture.

Based on the requirements mentioned above, the proposed architecture comprises four primary modules: a Primary Unit, a Supervision Unit, a Plan Execution Model, and the Hybrid Digital Twin. Figure 1 shows a schematic representation of the architecture. The Primary Unit executes the control algorithms for the physical disassembly of components, such as motion planning routines and mid-level control. Robots acting on the process are connected to this unit, as well as industrial servo drives, I/O modules, and different kinds of tools (*e.g.,* electric, pneumatic, hydraulic). The most commonly adopted communication strategies are industrial field-buses (*e.g.,* EtherCAT, ProfiNet, Modbus, etc.). The Supervision Unit performs side and non-physical tasks to support the disassembly, such as localisation and inspection. The Plan Execution Model orchestrates the whole cell, organising the order of operations, raising alarms on bad states, and restoring faults. Finally, the Hybrid Digital Twin reflects the cell state and enables different levels of virtualisation: from a fully simulated environment, to a mixed real-virtual setup, up to a complete mirroring of real-world operation. Information exchange among network nodes relies on inter-process communication based on the Data Distribution Service (DDS) (Object Management Group 2015). DDS is an open standard implementing a publisher-subscriber model where participants publish data structures into a global space and subscribe to them, knowing only identifiers (called *topics*) and types. Also, it supports QoS policies, allowing extensive tuning of communication details such as reliability, lifespan and security. As each DDS participant is not interested in the presence of others but only in the data published, this model encourages modularity and simplifies data exchange. Each hardware device is connected to the DDS through a software interface that acts as an abstraction of the physical layer. The following sections report a detailed description of the four primary modules.

3.1. *Plan execution model*

This module orchestrates the operations within the work cell, equipping the architecture with the necessary flexibility and intelligence to manage multiple tasks and actors simultaneously. Tasks can be executed sequentially or in parallel, with their order determined by predefined logic and sensor data from the field. The plans must be adaptable

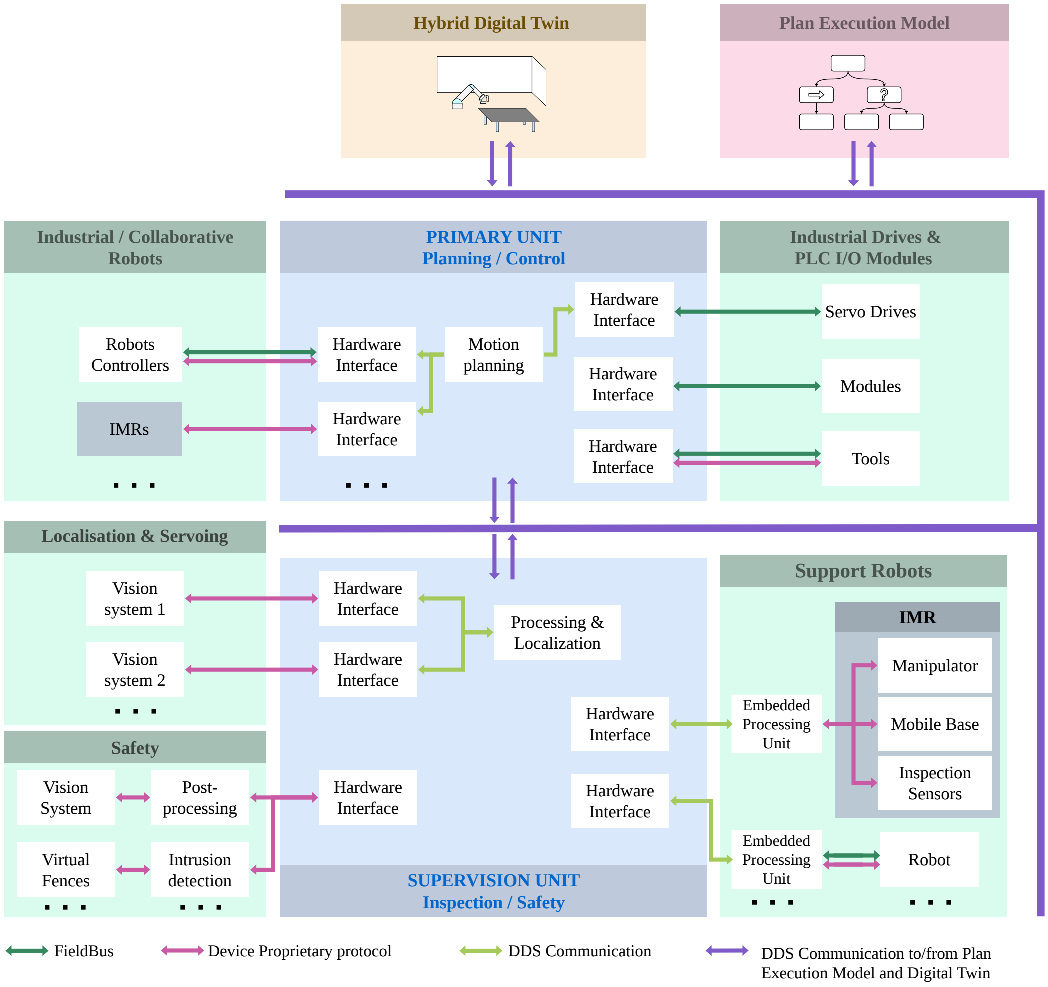

Figure 1. Schematic representation of the control architecture comprising the following modules: a Plan Execution Model (Section 3.1), a Planning & Control Computational Unit (Section 3.2), an Inspection & Safety Supervision Unit (Section 3.3), and a Hybrid Digital Twin (Section 3.4). One or more Industrial Mobile Robots (IMRs) can be integrated to provide additional inspection and manipulation functionalities.

to unforeseen failures, incorporating fail-safe mechanisms to ensure continuous operation. This adaptability is crucial for completing complex tasks and guiding the system toward a safe state when needed.

In task planning literature, various techniques have been proposed to organise the flow of operations, the most well-known being *State Machines* (González-Santmarta et al. 2023), *Behavior Trees* (Kokotinis et al. 2024), *Petri Nets* (Casalino et al. 2021), *AI-based planners* (Jin et al. 2024; Martín et al. 2021), *Mixed Integer Linear Programming (MILP) planners* (Monguzzi et al. 2022; Faccio, Granata, and Minto 2023), and recently *LLM-based planners* (Singh et al. 2023; Rana et al. 2023). Regardless of the specific plan execution model, a prerequisite is the definition of a set of generic, atomic operations that can be easily combined into more complex tasks. Parameters for each operation must be defined so that general skills can be tailored to the specific needs (*e.g.,* an atomic task could be moving the robot's end effector from the current position to a given target position; required parameters are the goal, the name of the joints to move, the speed scaling factor, which controller to use, and so forth).

3.2. *Primary Unit: planning and control*

The standard industry practice is writing brand-specific programs using proprietary programming languages. This approach has several drawbacks. First, the available functionalities regarding language and libraries are restricted to those provided by the manufacturer. Second, there are portability issues, as the resulting program cannot be used on robots from other manufacturers and sometimes even on different models from the same supplier. Each change in the robotic brand introduces a learning barrier (Villagrossi et al. 2023). Robot manufacturers typically provide communication systems that enable direct interaction with robot controllers, allowing position, velocity, and force control (also called *open controllers*). This communication also provides the robot's status as feedback, such as joint configuration and currents, allowing interfacing between the robot and high-level software components. This setup enables external control units to leverage state-of-the-art path planners and replanners for collision-free trajectory planning in static and dynamic environments to adapt the robot's movement to various situations (Karaman and Frazzoli 2011; Tonola et al. 2023). Moreover, external-unit control simplifies the integration of advanced controllers, enabling functionalities not typically found in standard industrial robot controllers. For example, impedance control algorithms (Hogan 1985) allow for the robot's movements to be adjusted based on interaction forces with the environment. Additionally, high-level programming offers the flexibility to create new types of controllers, opening up virtually limitless development possibilities. Finally, the control module interfaces with low-level automation via the fieldbus, enabling I/O operations and controlling external axes (*e.g.,* those positioning the device under disassembly).

3.3. *Supervision Unit: inspection and safety*

3.3.1. Inspection

Component disassembly requires constant supervision to control the process and monitor hazards like fire, gases or radiation. The placement of inspection sensors is crucial for accurately representing the system's current state. Fixed sensors mounted in the cell can have their view obstructed by manipulators performing the task. One potential solution is to mount the sensors on the manipulators themselves; however, this can be impractical due to pose limitations and disturbances from the process environment (*e.g.,* vibrations, heat, dust). Using mobile manipulators for inspection solves most of these problems, as it allows the sensors to remain at a safe distance from the process while selecting the best point of observation. This approach is particularly effective when the device being disassembled is cumbersome (*e.g.,* mechanical parts, automotive BPs, etc.).

3.3.2. Localisation and servoing

Traditionally, workpieces are mechanically constrained in fixed, known positions within tight tolerances to support repetitive operations. Vision systems eliminate the need for such mechanical placement by identifying and localising target objects. This enhances cell flexibility, allowing greater freedom in object positioning and orientation. It also enables the detection of physical damage that may hinder disassembly, as EoL

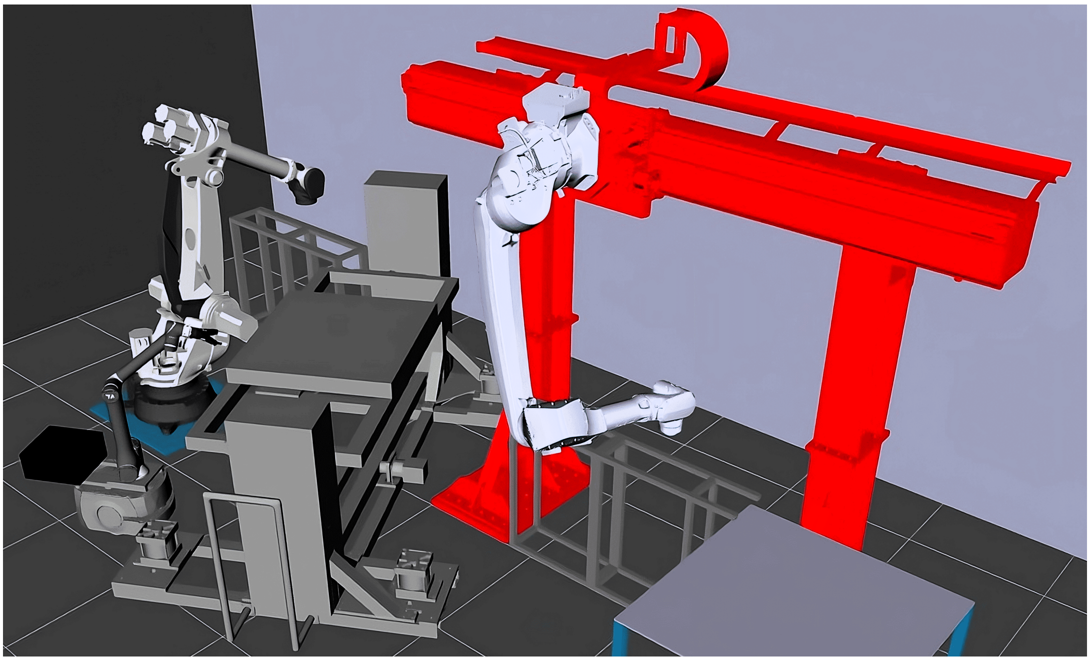

Figure 2. View of the Hybrid Digital Twin of the multi-robot cell.

products often deviate from their original shape and dimensions. Additionally, vision systems support visual servoing, enabling tightly integrated control and sensing strategies (Chaumette, Hutchinson, and Corke 2016).

3.3.3. Safety

Smart HRC can significantly enhance the efficiency of complex disassembly tasks (Li et al. 2024). However, to fully realise its benefits, mechanical safety features — such as fences — should be minimised to reduce obstructions and facilitate human intervention during disassembly. Integrating advanced safety sensors within the work cell can further enhance HRC by enabling automatic alarm reset and eliminating the need for manual intervention. Additionally, new safety measures must be introduced to ensure a safe and efficient working environment (Dobra and Dhir 2020). Detection systems could be integrated alongside existing industrial safety hardware. Sensor data from these devices would be processed on the edge by a dedicated unit to improve response time.

3.4. *Hybrid Digital Twin*

Digital twins are a pillar of the Industry 4.0 paradigm, essential tools for various software components. Digital twins provide planning algorithms with all the necessary information by creating a detailed replica of the work environment. They also function as debugging tools, allowing testing tasks in a virtual environment. Implementing a hybrid digital twin enables full and partial system simulations for complex systems with multiple agents operating simultaneously. During commissioning or maintenance, some elements may be missing from the setup; however, simulating their interactions with other physical devices can significantly accelerate testing and deployment.

4. Architecture deployment: robotic battery disassembly use case

The architecture introduced in Section 3 was applied to control a robotic cell developed for automotive battery pack disassembly[5]. The control architecture was deployed to control the robotic cell presented in (Villagrossi and Dinon 2023). The authors analysed the disassembly tasks in that paper and proposed a semi-automatic, partially destructive disassembly. The integrity was maintained for components with potential reuse and remanufacturing, while low-value components were removed with destructive techniques such as cutting. The following disassembly steps refer to the Stellantis 500e "low range" BP, but the tasks are easily portable to other models. The disassembly begins with thoroughly inspecting the battery pack to identify potential damage. Furthermore, the process requires monitoring continuous temperature and gas emissions to detect possible explosions. The battery is then discharged to the lower limit that the battery management system allows, and the cooling liquid is drained. Following the initial preparation, the process moves to removing the fuses. This involves opening a plate and carefully unscrewing and manipulating the metallic components to extract the fuses safely. The next step is the removal of the upper cover, which is both screwed and sealed with mastic. After unscrewing the cover, a combination of tools is required to separate it from the frame while maintaining an appropriate pulling force to prevent the parts from resticking. The disassembly continues with the removal of wires. This task involves cutting plastic ties, unplugging connectors, and removing the fastening systems. Subsequently, the battery management system is dismantled by unscrewing and extracting its components. Attention then shifts to the lower cover, which undergoes a similar procedure to the upper cover. The battery pack is rotated 180 degrees, the screws are removed, and the cover is separated from the frame using the same set of tools. Finally, the extraction of the battery modules is carried out. This requires another 180-degree rotation of the battery pack, fastening release, and removing cooling hoses to free the modules. The modules are then separated from the frame by breaking the sealing silicone, involving careful unscrewing and manipulation of the hoses and modules. Throughout this process, every step must be conducted with the utmost care to ensure that the components are handled safely and preserved for future reuse and remanufacturing. This facilitates an efficient and sustainable approach to electric vehicle battery pack disassembly.

The analysis of the disassembly steps made in (Villagrossi and Dinon 2023) brought to the decision to automate the following tasks: unscrewing, sealed part separation, cutting (liquid hoses and wires), and parts handling (due to the high variability of BP components, part handling imposes multiple grippers to address all the cases). The remaining disassembly tasks (*i.e.,* plastic ties removal, electric connectors removal, and cooling hoses removal) were left in charge of the operator because automation can be less effective than human intervention in terms of flexibility, time spent on the task, and economic sustainability of the application. The choice of semi-automatic disassembly imposes the design of a cell, which involves sharing human-robot workspaces and performing collaborative tasks. The cell has two industrial robots: a medium-payload robot (R1) mounted on a linear axis to extend its workspace and a medium-high payload robot (R2). An Industrial Mobile Manipulator (IMM) inspects the BP and picks small parts. Given the numerous tools required for disassembly, R1 and R2 are fitted with a quick tool change system. Figure 3 illustrates the cell layout, highlighting the key elements. Detailed descriptions of all hardware components, tools, and sensors are

---

[5] All the developed code is available in the GitHub repository: `https://github.com/JRL-CARI-CNR-UNIBS/battery_cell.git`. A video of a disassembly demo can be found here: `https://youtu.be/gLpW7oX3Q-c`.

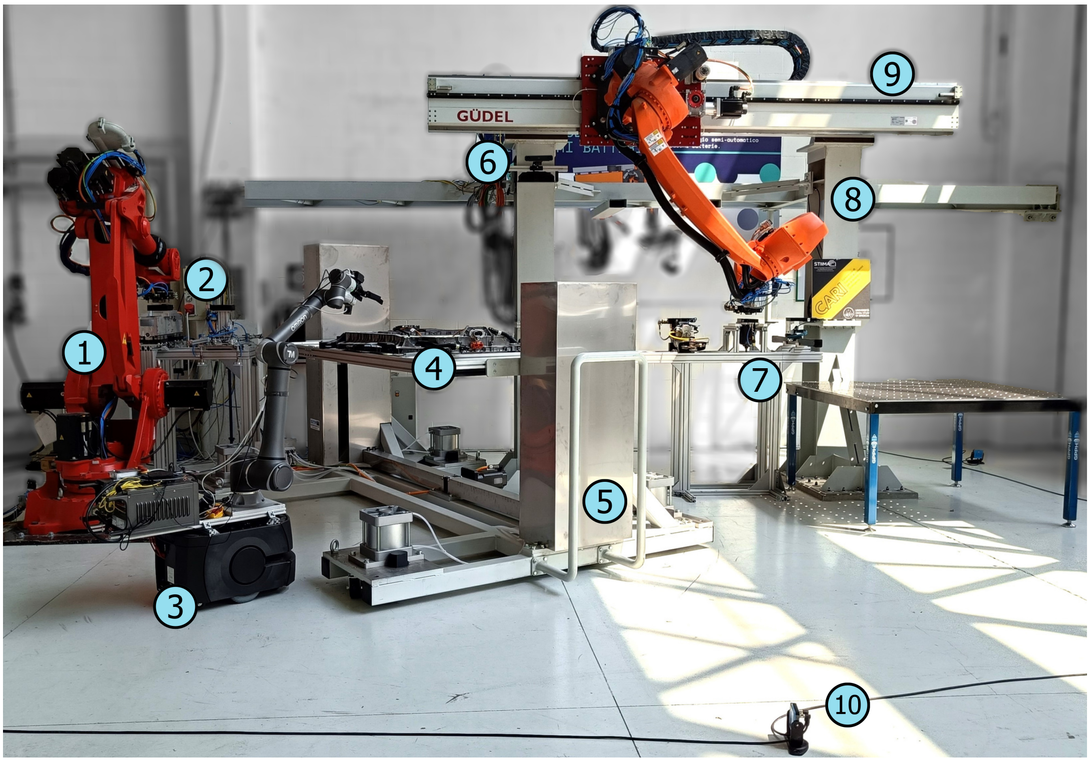

Figure 3. The multi-robot cell for battery disassembly. The main components are tagged as follows: (1) Robot 2 (COMAU), (2) Robot 2 tools, (3) Industrial Mobile Manipulator (OMRON + TECHMAN), (4) Battery Pack, (5) Battery Handling System, (6) ZED Camera, (7) Robot 1 tools, (8) Robot 1 (KUKA), (9) Linear Axis, and (10) Inxpect Radar.

provided in Section 4.1.

R1 is responsible for unscrewing screws and bolts (impact wrench tool), separating sealed parts (wedge spreader tool), such as the upper and the lower cover, and cutting wires, busbars, and hoses when the user decides not to preserve their integrity (shear tool). R2 is responsible for parts handling; the robot has two different handling tools: a two-finger gripper to manipulate heavy parts, such as the modules, and a vacuum gripper to pick lighter planar elements, such as covers. R1 and R2 must work together for part separation, as the upper and lower covers are removed, using the wedge spreader (R1) and the vacuum gripper (R2) simultaneously. The IMM has a single two-finger gripper to pick small parts, such as fuses, plates, hoses, wires, etc. Additionally, the IMM functions as a mobile inspection system, using its end-effector to bring an infrared camera and gas sensors close to the disassembly area for real-time monitoring. The battery pack is placed on a Battery Handling System (BHS), which can tilt and lift the pack during the disassembly with two actuated axes. The BHS has wheels to be moved outside the cell in case of emergency (*i.e.,* thermal runaway and gas emission). Additional 2D and 3D vision systems are mounted on the R1 and R2 tools to detect elements in the battery pack.

4.1. ***Hardware Setup***

The architecture of the integrated system, illustrated in Figure 4, features five primary subsystems positioned around the battery pack, which is mounted on a BHS with a controllable orientation and height to facilitate disassembly operations. Each element is designated to perform specific tasks during the disassembly sequence and interacts dynamically with the other subsystems throughout the process.

Overall, the system utilises the following robots:

- KUKA KR 50 R2500 (R1): a 6-DOF industrial manipulator used for battery localisation and disassembly. The robot is mounted on a linear axis to improve the workspace;
- COMAU NJ-220-2.7 (R2): a 6-DOF industrial manipulator dedicated to battery disassembly;
- OMRON LD60 and TECHMAN TM12 (IMM): an industrial mobile manipulator employed for support and inspection purposes;
- BHS: custom-made handling system able to support the BP during the disassembly. The BHS can lift and rotate the BP through two electro-actuated axes.

Below, we delve into the functionalities and interactions of these components.

4.1.1. The Core unit

The Core unit is the operational core of our system, comprising the two computation units of Figure 1: a workstation U1 as Primary Unit and a workstation U2 as Supervision Unit that orchestrates and synchronises the robotic actions throughout the disassembly process. The system's software architecture, which governs these operations, is based on ROS 2 (Robot Operating System 2) (Macenski et al. 2022) and it will be outlined in Section 4.2. U1 mounts a Linux Kernel with the PREEMPT_RT patchset that enables controlling hardware devices with real-time requirements. On the other hand, U2 should be able to exploit GPU features unsupported by PREEMPT_RT, thus a common stable Linux Kernel has been used.

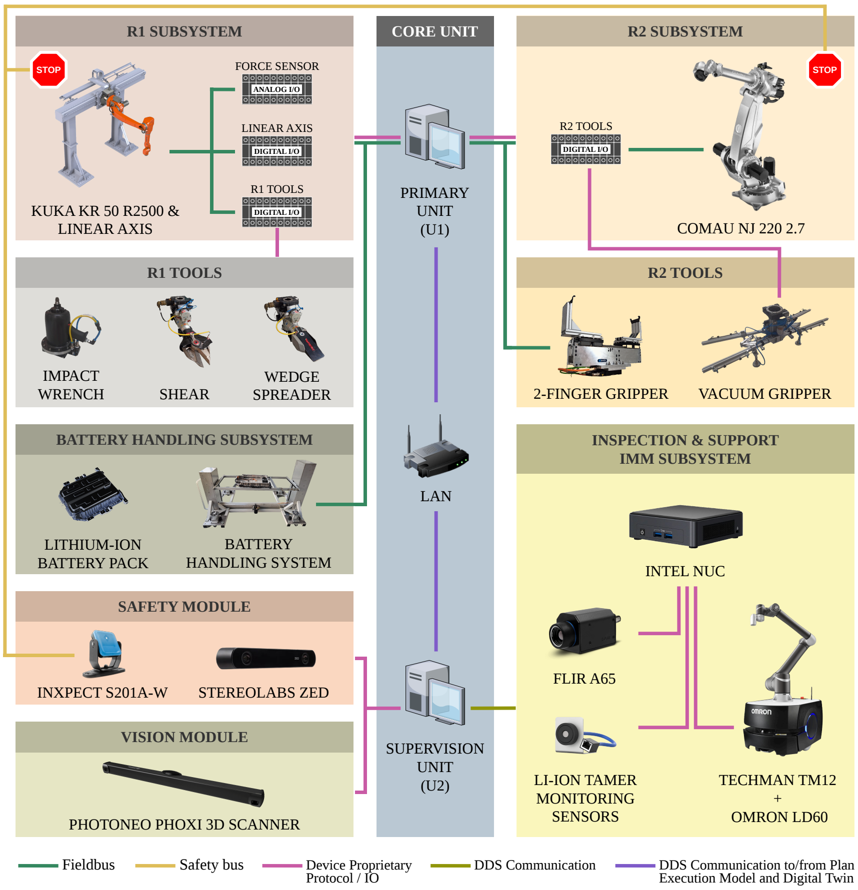

Figure 4. System architecture consisting of seven primary components: the *Core unit*, the *R1 subsystem* with its tools, the *R2 subsystem* with its tools, the *Inspection & support IMM subsystem*, the *Battery Handling Subsystem* (including the EVB under disassembly), the *Safety module*, and the *Vision module*.

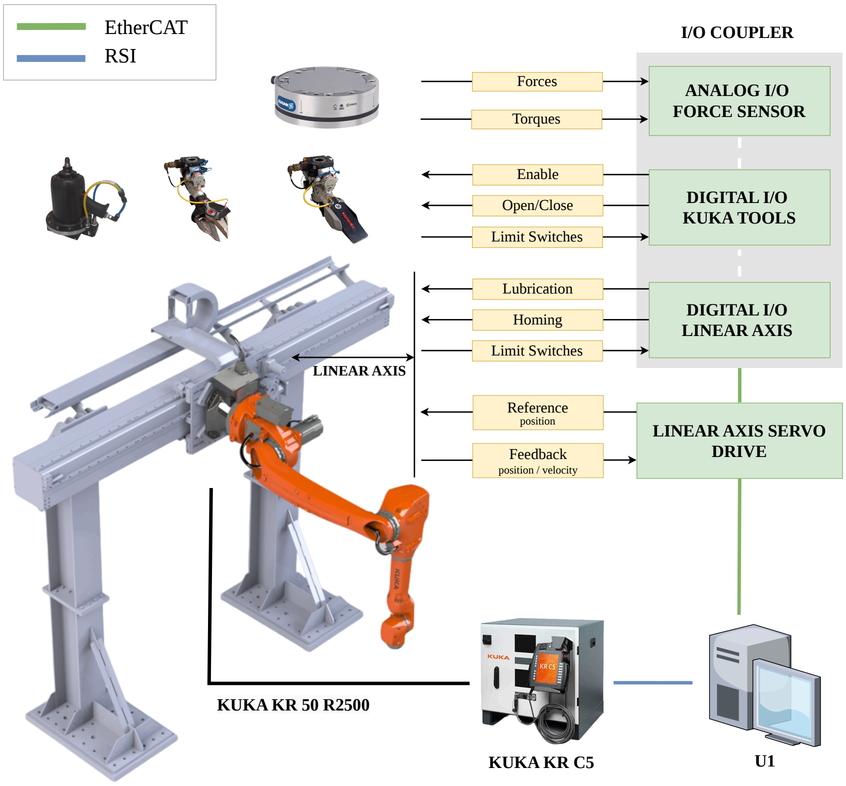

Figure 5. EtherCAT and RSI (Robot Sensor Interface) connection diagram within the R1 subsystem. Green lines represent EtherCAT-based communications, and blue lines indicate RSI connections.

4.1.2. The R1 subsystem

The robot R1, featuring a linear axis, is a 6-DoF KUKA KR 50 R2500 robotic manipulator. This robot performs various disassembly operations on battery packs, including identifying components, unscrewing fasteners, separating sealed parts like protective covers, and cutting wires, bars, and tubes. The system uses several tools to support these functions, including a camera holder, a shear, a wedge spreader, and an impact wrench. These tools are organised on a rack, and a quick change system (SCHUNK SWS) is integrated into the robot's end-effector to change the tool easily.

Control over the entire module is managed by U1, which handles tasks such as reading and writing digital signals, monitoring sensors, and robot operations. Communication within the system happens through two protocols: EtherCAT and RSI (Robot Sensor Interface), with EtherCAT being the dominant protocol across the setup and RSI used specifically for interactions with the KUKA robot. The connections are detailed in Figure 5. The EtherCAT communication chain begins at U1 and first connects to the servo drive that controls the linear axis's motion. From there, it links to the I/O modules responsible for the axis, the tools, and the robot's force sensor. Within this chain, a specific digital I/O module is dedicated to critical functions for the linear axis, such as initiating guide lubrication, executing homing sequences, and processing signals from limit switches. Another module handles tool management, including enabling and disabling tools, operating their opening and closing mechanisms, and monitoring their limit switches. An analogue module follows, reading force and torque measurements from the force sensor on the robot's wrist.

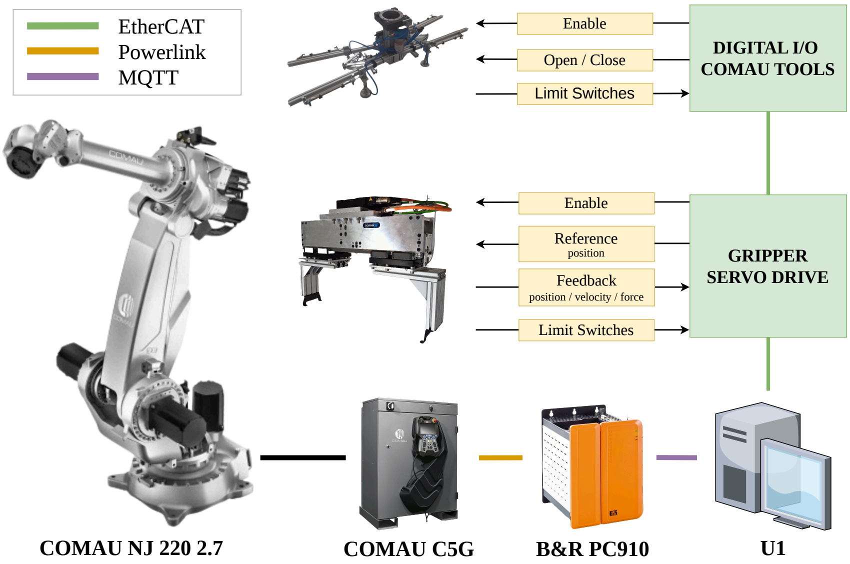

Figure 6. Communication scheme within the R2 subsystem. Green lines represent EtherCAT communications, while pink lines indicate MQTT connections, and orange lines represent Powerlink connections.

4.1.3. The R2 subsystem

The R2 subsystem is based on a COMAU NJ-220-2.7 manipulator. This robot primarily handles heavy parts using two specialised tools: a vacuum gripper for lifting planar, bulky objects like battery covers, and a two-fingered gripper for manipulating heavier components.

Figure 6 illustrates the communication architecture within the system. U1 controls the robot via an industrial PC (integrated into the robot controller). Communication between the industrial PC and the robot controller is established through Powerlink, while MQTT is used to connect the industrial PC and U1. The EtherCAT chain extends to a digital I/O module that controls tool operations and ends at the servo drive, managing the two-fingered gripper.

4.1.4. The Safety module

During certain stages of the disassembly process, specific areas must be restricted for operators to prevent collisions with the robots and to avoid performing potentially hazardous operations on the batteries in the presence of human workers. On the other hand, some phases require human-robot collaboration, as tasks like disconnecting connectors and handling cables are complex to automate fully. In these cases, it is essential to continuously monitor the operators to ensure safe and efficient interaction between humans and robots. The primary safety system comprises four Inxpect S201A-W safety radar (Inxpect. 2024) sensors connected to the robot controllers with dual-channel safety output; the radars also communicate to U2 via the Modbus protocol, which allows access to the distance and angle information from the closest person to the sensor in the monitored area. These sensors establish virtual barriers around hazardous areas, detecting moving objects within their field of view. Upon crossing these barriers, an alarm triggers, stopping the robot's operations nearby. This system offers advantages over standard laser gates by automatically defining discrete safety zones with varying levels of risk. Indeed, the safety radar can be configured to autonomously reset the

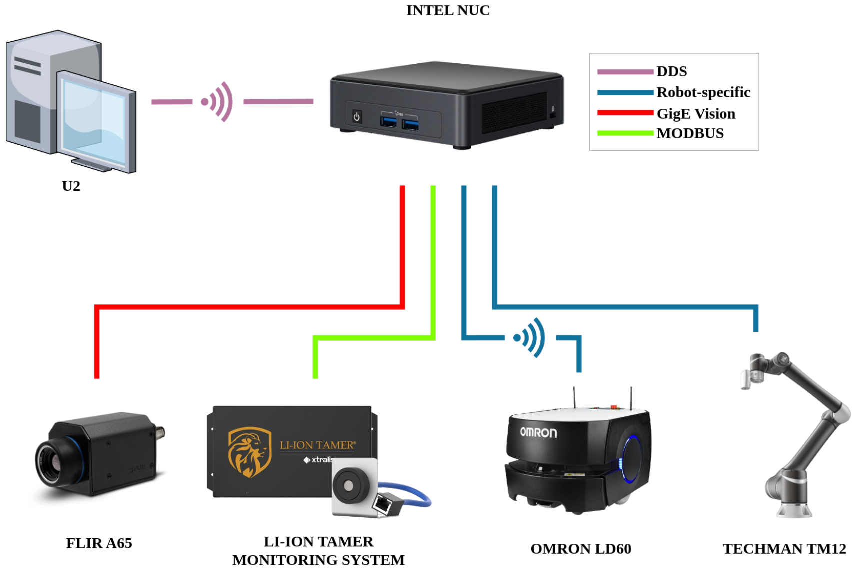

Figure 7. Communication scheme within the Inspection & support IMR subsystem. The Violet line is the DDS connection, blue lines represent robot-specific communication protocols, red lines indicate a connection based on the GigE Vision protocol, and green lines are the ones based on Modbus.

emergency stop once the operator goes outside the monitored area after waiting for a predefined timeout. The second system employs a Stereolabs ZED2 (Stereolabs. 2024) camera equipped with a multi-body skeletonisation library, communicating with U2 via the USB protocol. Unlike the radar network, this system is susceptible to dust and varying light conditions but provides detailed information on operator behaviour within the scene. This capability is crucial for human-robot collaboration tasks, enabling the robot to adjust its speed relative to the user's distance following ISO/TS 15066 guidelines (ISO Central Secretary 2016; Byner, Matthias, and Ding 2019). Furthermore, the camera's ability to track human poses over time allows for inferring current activities and, in some cases, predicting future movements. This capability supports the implementation of proactive safety measures (Li et al. 2023), such as adjusting the sequence of operations based on anticipated hazardous situations (Pupa and Secchi 2021; Sandrini, Faroni, and Pedrocchi 2025). This proactive approach helps prevent abrupt interruptions that could compromise operations. It is worth noting that ZED2 is not a certified safety sensor; it was used to track the human position and compute the human-robot distance for the implementation of the Speed and Separation Monitoring mode (ISO/TC 299 Robotics. 2011a,b; ISO Central Secretary 2016).

4.1.5. The Inspection & support IMM subsystem

Battery disassembly, particularly lithium-ion types, poses significant safety challenges due to the risk of thermal runaway, which can lead to cell failure, rapid temperature spikes, and the emission of toxic and flammable gases. The sensors on the IMM detect these hazardous conditions and trigger appropriate safety protocols. This surveillance enables the activation of safety protocols, such as relocating the battery to a secure area. The IMM features an OMRON LD60 mobile base and a TECHMAN TM12 manipulator, equipped with a FLIR A65 thermal imaging camera to track temperature changes and a Li-Ion Tamer gas sensor to detect hazardous and flammable gas leaks. An Intel NUC

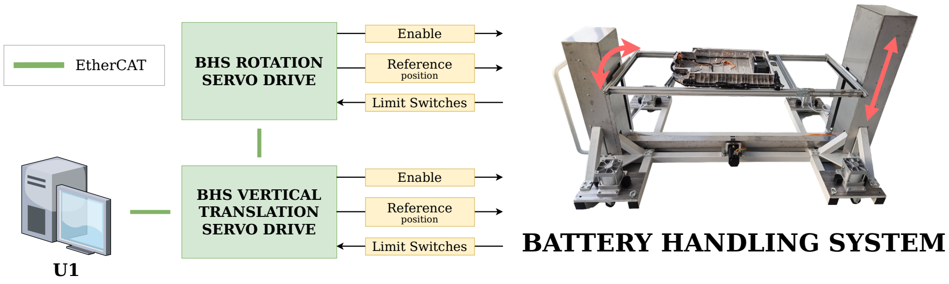

Figure 8. Communication scheme within the BHS subsystem. The EtherCAT chain from U1 reaches the servo drives that control the rotation and vertical translation of the BHS.

mini-computer installed on the IMM coordinates the operations of these devices, which receive instructions from U2. The NUC manages communications with the mobile base and the manipulator through wireless and wired TCP/IP, respectively, interfaces with the thermal imaging camera using the GigE Vision protocol, and with the gas monitoring system through Modbus. The connectivity configuration is illustrated in Figure 7.

4.1.6. The Vision module

Localising the battery in the workspace is a critical step at the beginning of the disassembly procedure. In this case, the vision system is used to localise the battery and detect features and parts to be disassembled. The KUKA robot mounts a Photoneo PhoXi XL 3D Scanner (Photoneo. 2024) to localise both the battery itself, but also screws, wires, modules, etc. Positioning the battery without tight constraints increases cell flexibility, granting freedom of movement for the battery holder, which, in turn, is of critical importance in case of damage to the battery and thermal runaway.

4.1.7. The BHS subsystem

The disassembly of a battery pack requires operations on both its upper and lower sections, necessitating the rotation of the pack. A custom BHS has been developed to move and securely hold the battery during the disassembly. This support allows for two types of movement: vertical and rotational. A vertical actuator adjusts the height of the battery, making different operations more accessible. A rotational actuator enables the support to rotate around a horizontal axis, exposing both sides of the battery for processing (see Figure 8). The BHS is also thought to support the operator during manual disassembly. The frame of the BHS is shaped to facilitate the operator's access to the BP. The regulation of the orientation and the height of the BP allows the operator to reach all the parts, improving ergonomics. The BHS actuators are controlled through two servo drives connected to the EtherCAT fieldbus starting at U1; see Figure 8.

The BHS is equipped with pneumatic pistons that securely lock it in place and can autonomously retract in hazardous situations to facilitate the safe evacuation of the battery.

4.2. ***Software modules***

The software structure, shown in Figure 9, highlights three principal blocks:

- **Plan Executer:** manager of the task execution plan.

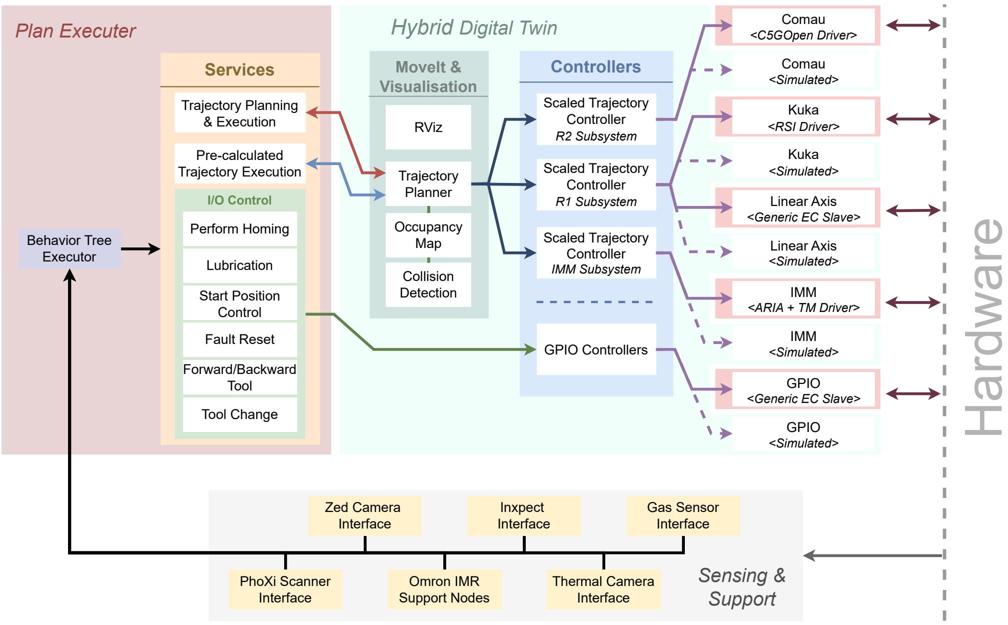

Figure 9. Software architecture consisting of three primary elements. (1) *Plan Executer* (red box): executes the sequence of tasks exploiting the ROS services in the yellow box (divided into trajectory and I/O control services); (2) *Hybrid Digital Twin* (green box): virtualises the environment integrating sensor feedback and computation of reference signals for the controllers in the blue box; (3) *Hardware Interfaces* (vivid red box): allow read/write communication with the real hardware. Moreover, a *Sensing & Support* module (grey box) encompasses all nodes responsible for sensor interfacing and interaction with the OMRON mobile platform.

- **Hybrid Digital Twin:** virtual-physical environment enabling offline and online supervision and control.
- **Hardware Interfaces:** software modules that send commands to hardware and receive sensor feedback.

Additionally, the **Sensing and Support** module comprises a set of nodes responsible for managing specific sensing devices to detect the cell state and for interacting with the OMRON mobile base.

All libraries used to implement the architecture are listed in Table 2.

4.2.1. Plan Executer

The process is governed using Behavior Trees (Florez-Puga et al. 2009; Mateas and Stern 2002; Colledanchise and Ögren 2018), a robust plan execution model that breaks down complex tasks into simpler sub-tasks. This modular approach enhances flexibility, making it easier to reconfigure and reuse sub-tasks across different contexts, simplifying scheduling and debugging. Moreover, Behavior Trees could handle faults and unexpected events effectively, ensuring robust process management. Our implementation leverages the BehaviorTree.CPP library and its ROS 2 integration. This library provides essential interfaces for incorporating ROS 2 actions and services as tree leaves, ensuring a robust and flexible process management system. The implementation manages multiple agents and their synchronisation, and the decomposition into subtrees for simplicity. The leaves rely on three specific services:

- the **I/O Control** service manages reading and writing various inputs and outputs to control the robot tools and support services.

Table 2. Overview of the software libraries utilised.

Library	Description	Authors
BehaviorTree.CPP	Behavior Tree implementation in C++.	Davide Faconti (2024b)
BehaviorTree.ROS	ROS integration for BehaviorTree.CPP.	Davide Faconti (2024a)
ZED ROS 2 Wrapper	ZED camera wrapper for ROS 2.	Stereolabs (2024)
ARIA	Communication with OMRON LD60.	Omron Adept (2016)
TM ROS Driver	ROS Driver for TECHMAN TM12.	TM ROS Driver (2024)
FLIR Camera Driver	ROS 2 integration for FLIR cameras based on FLIR's Spinnaker SDK.	Flir Camera Driver (2024)
KUKA Drivers	ROS 2 Driver for KUKA robot.	Kroshu (2024)
COMAU Drivers	ROS 2 Driver for COMAU robot.	STIIMA-CNR (2024)
IgH EtherCAT	EtherCAT Master for Linux.	Pose (2024)
EtherCAT ROS 2 Driver	Communication with generic EtherCAT slaves.	ICube Robotics (2024)
PyModBus	Full Modbus protocol implementation in Python.	PyModBus (2024)

- the **Pre-calculated Trajectory Execution** service loads and executes pre-calculated robot trajectories. This feature enables offline simulation of robot movements, which can then be precisely executed online. It is especially useful for tasks requiring high accuracy or those performed near objects, such as approaching screws or handling tools.
- the **Trajectory Planning** service computes and executes trajectories in real-time, moving the robot from its current configuration to a specific Cartesian pose. This leaf is ideal for free movements, typically used to transition between task execution areas.

4.2.2. Hybrid Digital Twin

The core of the robotic cell is a *hybrid digital twin*, *i.e.,* a virtual environment that replicates the physical setup, enabling both control and supervision in offline (*e.g.,* for debugging and testing) and online (*e.g.,* for real-time monitoring, planning, sensing, and control) modes. This architecture allows for the virtual closure of control loops by simulating direct interaction with the robot's physical hardware interfaces. To make this possible, each hardware component is complemented by a corresponding simulated version, emulating the execution and enabling seamless integration within the virtual environment. As a result, even in the absence of specific physical components, the entire system's behaviour can be tested, accelerating commissioning phases, enhancing maintenance safety, and enabling realistic testing in partially implemented configurations. The digital twin supports multiple levels of virtualisation, allowing seamless switching between real hardware and its simulated counterpart, without requiring extensive changes or reconfiguration at higher levels, which remain agnostic to the underlying implementation.

On top of this layer, various ROS 2 controllers are deployed to execute the Plan Executer's requests and manage motion and I/O interactions. For instance, the *Scaled Trajectory Controller* extends the standard *Joint Trajectory Controller* provided by *ros2_control* (OSRF 2024b), enabling dynamic modulation of robot movement speeds in real time (*e.g.,* due to the human-robot distance). Moreover, a dedicated *GPIOcontroller* handles the I/O communication between the Plan Executer and the hardware component responsible for binary signals.

Furthermore, at a higher level of abstraction, motion planning is handled by *MoveIt 2* (Ioan A. and Sachin 2024), which provides advanced capabilities for collision checking and trajectory generation. Additionally, *RViz* is used as a visualisation interface for the cell state and real-time trajectory monitoring and debugging.

4.2.3. Hardware interfaces

The actual communication with the hardware is done through a set of drivers called *Hardware Components* (as per *ros2_control* nomenclature). Each component is specific to the hardware it interacts with and translates ROS data into hardware-ready commands, using specific libraries (*e.g., KUKA RSI Driver* for KUKA, *C5GOpen Driver* for COMAU, *ARIA* for OMRON, *TM ROS Driver* for TECHMAN, *IgH EtherCAT* for EtherCAT-enabled devices, and *PyModBus* for ModBus-enabled devices). The set of libraries and drivers interfacing with the hardware components is summarised in Table 2, along with their respective references.

As discussed in Section 4.2.2, hardware interfaces support a simulation mode in which low-level communication with the physical hardware is bypassed. Instead, commands are sent to the simulated component, whose computed effects virtually close the control loops.

5. Critical Analysis and Lessons Learnt

This section revisits the cell requirements outlined in Section 3, detailing how the proposed solution meets these objectives and highlighting the challenges and new needs discovered during implementation.

Network infrastructure

Building the software architecture on ROS 2 provided flexibility, modularity, and access to a wide range of open-source tools. ROS 2 relies on DDS (William Woodall 2019), a middleware suited for distributed systems thanks to its decentralised discovery and customizable Quality of Service (QoS) settings (Esteve Fernandez 2019). These features enhance modularity and robustness by avoiding single points of failure and allowing tailored communication, such as low-latency or high-reliability modes, without impacting the application-level design. Industrial applications require robust communication and must handle increasing numbers of heterogeneous devices (*e.g.,* thermal and RGB-D cameras, lidars, gas sensors), which can strain network and computational resources. DDS can be configured to address these challenges by disabling multicast discovery in favour of static peer lists, isolating data flows through logical partitions, and binding device types to specific network interfaces or partitions. System load can be balanced by distributing hardware management across multiple machines, such as dedicating units to AI processing, real-time robot control, mobile platforms, or auxiliary sensors. This approach limits communication to only the device pairs that need to exchange information, reducing unnecessary traffic. DDS QoS settings can also be fine-tuned per node to meet specific reliability or latency needs, ensuring consistent performance in real-time, resource-constrained environments. System designers must carefully tune these settings to achieve the desired behaviour regarding reliability, network load distribution, and computational efficiency.

Robotic Hardware Integration

By defining custom hardware interfaces and leveraging the open-controller architecture provided by the robot manufacturer, our integrated system enables direct control of robots from different brands. This alleviates operators from learning proprietary programming languages, facilitates real-time control, and manages potential stopping conditions. The modular structure of *ros2_control* allows specific controllers to be activated for each robot dynamically, enabling asynchronous control with centralised logic. However, this depends on the robot manufacturer's support for open control, which is not universally guaranteed. Additionally, direct communication via *ros2_control* necessitates developing specific hardware components, which can be challenging, and ensuring the availability of a library to manage the robot's communication protocol. Very often, the development of hardware components depends on open or known protocols. Thus, slightly different hardware manufacturer implementations can undermine such components' use. However, there has been a notable increase in support for open-controller architectures in recent years, possibly driven by the growing adoption of ROS 2 in academia and industry. PickNik Robotics (2024) provides a list of commercially available robots currently compatible with ROS 2.

Fault Identification and Recovery

Effective error handling involves two key steps: fault identification and recovery. Fault identification typically relies on additional dedicated sensors, while recovery often necessitates custom countermeasures, which can increase the system's complexity. This requires extra effort from the system designer to account for each potential failure, a challenge common to many robotic architectures in real-world applications.

The proposed architecture leverages Behavior Trees to manage execution logic, providing the flexibility to trigger recovery actions based on sensor signals detecting fault conditions. This allows system designers to implement custom fault detection and recovery strategies within the Behavior Tree structure, using a modular and maintainable framework. However, fault identification during complex task execution remains an open research area.

In some instances, recovery may require operator intervention. In such cases, it is crucial to establish clear procedures for safe access (as detailed in Sections 3.3.3 and 4.1.4) and to restore the system to a ready-to-restart state.

Safety in Human Detection

Deployed safety radar systems provide positional and angular information about detected obstacles within the cell. However, their accuracy falls short, and they cannot distinguish between moving objects (*e.g.,* moving IMM) and people. Consequently, to prevent false alarms and unnecessary system stoppages when a moving robot enters the area monitored by the radars, it is crucial to design the radar positioning based on the specific application and to place devices towards the cell periphery. The ability of radars to detect and distinguish humans remains an up-and-coming area of research (Scholz et al. 2024; Kianoush et al. 2021; Gelfert 2023; Buyukakkaslar, Erturk, and Aydin 2024). Integrating sensor fusion with a vision system could address this issue by accurately identifying humans. However, certified vision systems that can accurately track people (*e.g.,* human limbs and not only the body centre of mass) and meet safety standards for integration into the safety chain are currently unavailable. This aspect poses challenges for real-time data reception and timely execution of safety routines. Furthermore, ensuring safety requires robust measures to prevent operator entry into the cell during critical battery states, utilising a combination of safety devices and procedural controls. It is essential to restrict human access while maintaining the mobility of robots tasked with managing hazards, such as relocating batteries to designated safe zones away from infrastructure and personnel.

6. Conclusions and Future works

This paper explores the key features necessary for a robotic cell to perform disassembly operations effectively. It proposes a comprehensive framework to meet these requirements and illustrates its application in developing a robotic cell for disassembling electric vehicle battery packs. The proposed architecture integrates multiple robotic systems and sensors, allowing decentralised control and real-time processing capabilities. This integration enhances flexibility and modularity, which are essential for managing complex and variable disassembly tasks. The use of ROS 2 facilitates the integration of heterogeneous modules devoted to planning and control, supervision, localisation, servoing, and safety.

Despite these advancements, as outlined in Section 5, several areas still require further investigation and will be the focus of future work. Ensuring the safe entry of operators into the robot's working area is crucial in collaborative human-robot scenarios. Currently, the safety radars used in the robotic cell cannot differentiate between objects and humans. The authors are working on sensor fusion between cameras and radar sensors to address this issue. Robust recovery procedures will be implemented to handle execution errors efficiently. Lastly, authors are developing an intuitive low-code programming interface to promote industrial adoption.

Acknowledgement(s)

The CSMT Innovative Contamination Hub is acknowledged for hosting the joint laboratory between the National Research Council of Italy - Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing (CNR-STIIMA) and the University of Brescia, where the disassembly cell is located, and this work was carried out.

Funding

This research was supported by the Lombardy, Italy regional project EcoCirc [deliberation XI/4730 of the 17/05/2021].

Notes on contributor(s)

Michele Ferrari, Roberto Fausti, Cesare Tonola: Conceptualisation, Implementation, Methodology, Writing - original draft.
Michele Delledonne, Samuele Sandrini: Implementation, Methodology, Writing & review - original draft.
Manuel Beschi, Enrico Villagrossi: Conceptualisation, Investigation, Methodology, Writing - original draft, Writing - review & editing, Funding acquisition.

References

Adams, Matt, Xiao Li, Lucas Boucinha, Sameer S. Kher, Prith Banerjee, and Jose-Luis Gonzalez. 2022. "Hybrid Digital Twins: A Primer on Combining Physics-Based and Data Analytics Approaches." *IEEE Software* 39 (2): 47–52. https://doi.org/10.1109/MS.2021.3134042.

Albonico, Michel, Milica Đorđević, Engel Hamer, and Ivano Malavolta. 2023. "Software engineering research on the Robot Operating System: A systematic mapping study." *Journal of Systems and Software* 197: 111574. https://doi.org/10.1016/j.jss.2022.111574.

Beghi, Matteo, Francesco Braghin, and Loris Roveda. 2023. "Enhancing Disassembly Practices for Electric Vehicle Battery Packs: A Narrative Comprehensive Review." *Designs* 7 (5). https://doi.org/10.3390/designs7050109.

Blankemeyer, Sebastian, Denise Wiens, Tobias Wiese, Annika Raatz, and Sami Kara. 2021. "Investigation of the potential for an automated disassembly process of BEV batteries." *Procedia CIRP* 98: 559–564. The 28th CIRP Conference on Life Cycle Engineering, March 10 – 12, 2021, Jaipur, India, https://doi.org/10.1016/j.procir.2021.01.151.

Buyukakkaslar, Muhammet Talha, Mehmet Ali Erturk, and Muhammet Ali Aydin. 2024. "A Review on Radar-Based Human Detection Techniques." *Sensors* 24 (17). https://doi.org/10.3390/s24175709.

Byner, Christoph, Björn Matthias, and Hao Ding. 2019. "Dynamic speed and separation monitoring for collaborative robot applications – Concepts and performance." *Robotics and Computer-Integrated Manufacturing* 58: 239–252. https://doi.org/10.1016/j.rcim.2018.11.002.

Casalino, Andrea, Andrea Maria Zanchettin, Luigi Piroddi, and Paolo Rocco. 2021. "Optimal Scheduling of Human–Robot Collaborative Assembly Operations With Time Petri Nets." *IEEE Transactions on Automation Science and Engineering* 18 (1): 70–84. https://doi.org/10.1109/TASE.2019.2932150.

Chaumette, François, Seth Hutchinson, and Peter Corke. 2016. *Visual Servoing*, 841–866. Cham: Springer International Publishing.

Colledanchise, Michele, and Petter Ögren. 2018. *Behavior trees in robotics and AI: An introduction.* CRC Press.

Davide Faconti, Artem Voronov, Eric Dortmans. 2024a. "BehaviorTree.ROS." Accessed: July 2024, `https://github.com/BehaviorTree/BehaviorTree.ROS`.

Davide Faconti, Michele Colledanchise. 2024b. "BehaviorTree.CPP." Accessed: July 2024, `https://github.com/BehaviorTree/BehaviorTree.CPP/tree/master`.

Dimosthenopoulos, Dimosthenis, Fotios Panagiotis Basamakis, Christos Glykos, Angelos Christos Bavelos, Georgios Mountzouridis, and Sotiris Makris and. 2024. "A Digital Twin-based paradigm for programming and control of cooperating robots in reconfigurable production systems." *International Journal of Computer Integrated Manufacturing* 0 (0): 1–21. https://doi.org/10.1080/0951192X.2024.2428683.

Dobra, Zoltan, and Krishna S. Dhir. 2020. "Technology jump in the industry: human–robot cooperation in production." *Industrial Robot: the international journal of robotics research and application* 47 (5): 757–775. https://doi.org/10.1108/IR-02-2020-0039.

Erdogan, Cansu, Cesar Alan Contreras, Rustam Stolkin, and Alireza Rastegarpanah. 2024. "Multi-Robot Task Planning for Efficient Battery Disassembly in Electric Vehicles." *Robotics* 13 (5). https://doi.org/10.3390/robotics13050075.

Esteve Fernandez. 2019. "ROS 2 Quality of Service policies ." Available at: `https://design.ros2.org/articles/qos.html`.

Faccio, Maurizio, Irene Granata, and Riccardo Minto. 2023. "Task allocation model for human-robot collaboration with variable cobot speed." *Journal of Intelligent Manufacturing* 1–14.

Flir Camera Driver. 2024. "Flir Camera Ros 2 Integration." Accessed: July 2024, `https://github.com/ros-drivers/flir_camera_driver`.

Florez-Puga, Gonzalo, Marco Antonio Gomez-Martin, Pedro Pablo Gomez-Martin, BelÉn Diaz-Agudo, and Pedro Antonio Gonzalez-Calero. 2009. "Query-Enabled Behavior Trees." *IEEE Transactions on Computational Intelligence and AI in Games* 1 (4): 298–308. https://doi.org/10.1109/TCIAIG.2009.2036369.

Foo, Gwendolyn, Sami Kara, and Maurice Pagnucco. 2022. "Challenges of robotic disassembly in practice." *Procedia CIRP* 105: 513–518. The 29th CIRP Conference on Life Cycle Engineering, April 4 – 6, 2022, Leuven, Belgium., https://doi.org/10.1016/j.procir.2022.02.085.

Gelfert, Sebastian. 2023. "A sensor review for human detection with robotic systems in regular and smoky environments." *International Journal of Advanced Robotic Systems* 20 (3). https://doi.org/10.1177/17298806231175238.

González-Santmarta, Miguel Á., Francisco J. Rodríguez-Lera, Camino Fernández-Llamas, and Vicente Matellán-Olivera. 2023. "MERLIN2: MachinEd Ros 2 pLanINg." *Software Impacts* 15: 100477. https://doi.org/10.1016/j.simpa.2023.100477.

Harper, Gavin, Roberto Sommerville, Emma Kendrick, Laura Driscoll, Peter Slater, Rustam Stolkin, Allan Walton, et al. 2019. "Recycling lithium-ion batteries from electric vehicles." *Nature* 575 (7781): 75–86. https://doi.org/10.1038/s41586-019-1682-5.

Hellmuth, Jan F., Nicholas M. DiFilippo, and Musa K. Jouaneh. 2021. "Assessment of the automation potential of electric vehicle battery disassembly." *Journal of Manufacturing Systems* 59: 398–412. https://doi.org/10.1016/j.jmsy.2021.03.009.

Hjorth, Sebastian, and Dimitrios Chrysostomou. 2022. "Human–robot collaboration in industrial environments: A literature review on non-destructive disassembly." *Robotics and Computer-Integrated Manufacturing* 73: 102208. https://doi.org/10.1016/j.rcim.2021.102208.

Hogan, Neville. 1985. "Impedance control: An approach to manipulation: Part I-theory." *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME* 107 (1): 1–7. https://doi.org/10.1115/1.3140702.

ICube Robotics. 2024. "EtherCAT Driver for ROS2." Accessed: July 2024, `https://icube-robotics.github.io/ethercat_driver_ros2/`.

IEA. 2024. "Global EV Outlook 2023." Paris. Licence: CC BY 4.0, `https://www.iea.org/reports/global-ev-outlook-2024`.

Inxpect. 2024. "Inxpect safety radar equipment." `https://www.inxpect.com/en/`. Accessed: July 2024.

Ioan A., Sucan, and Chitta Sachin. 2024. "MoveIt." `https://moveit.ros.org/`. Accessed: July 2024.

Iovino, Matteo, Edvards Scukins, Jonathan Styrud, Petter Ögren, and Christian Smith. 2022. "A survey of Behavior Trees in robotics and AI." *Robotics and Autonomous Systems* 154: 104096. https://doi.org/https://doi.org/10.1016/j.robot.2022.104096.

ISO Central Secretary. 2016. *ISO/TS 15066:2016 Robots and robotic devices – Collaborative robots*. Standard. Geneva, CH: International Organization for Standardization.

ISO/TC 299 Robotics. 2011a. "ISO 10218-1:2011 Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots.” https://www.iso.org/standard/51330.html. Accessed: July 2024.

ISO/TC 299 Robotics. 2011b. “ISO 10218-2:2011 Robots and robotic devices — Safety requirements for industrial robots — Part 2: Robot systems and integration.” https://www.iso.org/standard/41571.html. Accessed: July 2024.

Jin, Yixiang, Dingzhe Li, Yong A, Jun Shi, Peng Hao, Fuchun Sun, Jianwei Zhang, and Bin Fang. 2024. “RobotGPT: Robot Manipulation Learning From ChatGPT.” *IEEE Robotics and Automation Letters* 9 (3): 2543–2550. https://doi.org/10.1109/LRA.2024.3357432.

Kaarlela, Tero, Enrico Villagrossi, Alireza Rastegarpanah, Alberto San-Miguel-Tello, and Tomi Pitkäaho. 2024. “Robotised disassembly of electric vehicle batteries: A systematic literature review.” *Journal of Manufacturing Systems* 74: 901–921. https://doi.org/https://doi.org/10.1016/j.jmsy.2024.05.013.

Karaman, Sertac, and Emilio Frazzoli. 2011. “Sampling-based algorithms for optimal motion planning.” *The International Journal of Robotics Research* 30 (7): 846–894. https://doi.org/10.1177/0278364911406761.

Kianoush, Sanaz, Stefano Savazzi, Manuel Beschi, Stephan Sigg, and Vittorio Rampa. 2021. “A Multisensory Edge-Cloud Platform for Opportunistic Radio Sensing in Cobot Environments.” *IEEE Internet of Things Journal* 8 (2): 1154–1168. https://doi.org/10.1109/JIOT.2020.3011809.

Kokotinis, George, George Michalos, Zoi Arkouli, and Sotiris Makris. 2024. “A Behavior Trees-based architecture towards operation planning in hybrid manufacturing.” *International Journal of Computer Integrated Manufacturing* 37 (3): 324–349. https://doi.org/10.1080/0951192X.2023.2228254.

Kousi, Niki, Christos Gkournelos, Sotiris Aivaliotis, Christos Giannoulis, George Michalos, and Sotiris Makris. 2019. “Digital twin for adaptation of robots’ behavior in flexible robotic assembly lines.” *Procedia Manufacturing* 28: 121–126. https://doi.org/https://doi.org/10.1016/j.promfg.2018.12.020.

Kroshu. 2024. “ROS 2 Kuka Drivers.” Accessed: July 2024, https://github.com/kroshu/kuka_drivers.

Lander, Laura, Chris Tagnon, Viet Nguyen-Tien, Emma Kendrick, Robert J.R. Elliott, Andrew P. Abbott, Jacqueline S. Edge, and Gregory J. Offer. 2023. “Breaking it down: A techno-economic assessment of the impact of battery pack design on disassembly costs.” *Applied Energy* 331: 120437. https://doi.org/https://doi.org/10.1016/j.apenergy.2022.120437.

Li, Shufei, Pai Zheng, Sichao Liu, Zuoxu Wang, Xi Vincent Wang, Lianyu Zheng, and Lihui Wang. 2023. “Proactive human–robot collaboration: Mutual-cognitive, predictable, and self-organising perspectives.” *Robotics and Computer-Integrated Manufacturing* 81: 102510. https://doi.org/https://doi.org/10.1016/j.rcim.2022.102510.

Li, Weidong, Yiqun Peng, Yu Zhu, Duc Truong Pham, A.Y.C. Nee, and S.K. Ong. 2024. “End-of-life electric vehicle battery disassembly enabled by intelligent and human-robot collaboration technologies: A review.” *Robotics and Computer-Integrated Manufacturing* 89: 102758. https://doi.org/https://doi.org/10.1016/j.rcim.2024.102758.

Lu, Yingqi, Weidi Pei, and Kaiyuan Peng. 2023. “State of the art of automatic disassembly of WEEE and perspective towards intelligent recycling in the era of Industry 4.0.” *The International Journal of Advanced Manufacturing Technology* 128: 2825–2843. https://doi.org/https://doi.org/10.1007/s00170-023-12043-3.

Macenski, Steven, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. 2022. “Robot Operating System 2: Design, architecture, and uses in the wild.” *Science Robotics* 7 (66): eabm6074. https://doi.org/10.1126/scirobotics.abm6074.

Malavolta, Ivano, Grace Lewis, Bradley Schmerl, Patricia Lago, and David Garlan. 2020. “How do you architect your robots? state of the practice and guidelines for ROS-based systems.” In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice*, ICSE-SEIP ’20, New York, NY, USA, 31–40. Association for Computing Machinery.

Martín, Francisco, Jonatan Ginés Clavero, Vicente Matellán, and Francisco J. Rodríguez. 2021.

"PlanSys2: A Planning System Framework for ROS2." In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9742–9749. IEEE Press. https://doi.org/10.1109/IROS51168.2021.9636544.

Mateas, M., and A. Stern. 2002. "A behavior language for story-based believable agents." *IEEE Intelligent Systems* 17 (4): 39–47. https://doi.org/10.1109/MIS.2002.1024751.

Mattila, Joel, Riku Ala-Laurinaho, Juuso Autiosalo, Pauli Salminen, and Kari Tammi. 2022. "Using Digital Twin Documents to Control a Smart Factory: Simulation Approach with ROS, Gazebo, and Twinbase." *Machines* 10 (4). https://doi.org/10.3390/machines10040225.

Meng, Kai, Guiyin Xu, Xianghui Peng, Kamal Youcef-Toumi, and Ju Li. 2022. "Intelligent disassembly of electric-vehicle batteries: A forward-looking overview." *Resources, Conservation and Recycling* 182. https://doi.org/https://doi.org/10.1016/j.resconrec.2022.106207.

Monguzzi, Andrea, Mahmoud Badawi, Andrea Maria Zanchettin, and Paolo Rocco. 2022. "A mixed capability-based and optimization methodology for human-robot task allocation and scheduling." In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 1271–1276.

Nakagawa, Elisa Yumi, Pablo Oliveira Antonino, Frank Schnicke, Rafael Capilla, Thomas Kuhn, and Peter Liggesmeyer. 2021. "Industry 4.0 reference architectures: State of the art and future trends." *Computers & Industrial Engineering* 156: 107241. https://doi.org/10.1016/j.cie.2021.107241.

Object Management Group. 2015. *OMG Data Distribution Service (DDS)*. Standard formal/2015-04-10. Object Management Group OMG. http://www.omg.org/spec/DDS.

Omron Adept. 2016. "Advanced Robotics Interface for Applications (ARIA)." Accessed: July 2024, https://github.com/cinvesrob/Aria.

OSRF. 2024a. "Gazebo." Accessed: July 2024, https://gazebosim.org/home.

OSRF. 2024b. "ros2_control documentation." https://control.ros.org/rolling/index.html. Accessed: July 2024.

OSRF. 2024c. "RVIZ." Accessed: July 2024, http://wiki.ros.org/rviz.

Photoneo. 2024. "PhoXi 3D scanner XL." https://www.photoneo.com/products/phoxi-scan-xl/. Accessed: July 2024.

PickNik Robotics. 2024. "ROS 2 Compatible Hardware." Available at: https://picknik.ai/hardware-ecosystem/#commercial.

Poschmann, Hendrik, Holger Brüggemann, and Daniel Goldmann. 2020. "Disassembly 4.0: A Review on Using Robotics in Disassembly Tasks as a Way of Automation." *Chemie Ingenieur Technik* 92 (4): 341–359. https://doi.org/10.1002/cite.201900107.

Pose, Florian. 2024. "EtherCAT master for Linux." Accessed: July 2024, https://gitlab.com/etherlab.org/ethercat.

Priyono, Anjar, Winifred L Ijomah, and Umit S Bititci. 2015. "Strategic operations framework for disassembly in remanufacturing." *Journal of Remanufacturing* 5: 1–16. https://doi.org/10.1186/s13243-015-0018-3.

Pupa, Andrea, and Cristian Secchi. 2021. "A Safety-Aware Architecture for Task Scheduling and Execution for Human-Robot Collaboration." In *IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS)*, 1895–1902.

PyModBus. 2024. "PyModBus - A Python Modbus Stack." https://pymodbus.readthedocs.io/en/latest/index.html.

Qu, Mo, D. T. Pham, Faraj Altumi, Adeyemisi Gbadebo, Natalia Hartono, Kaiwen Jiang, Mairi Kerin, et al. 2024. "Robotic Disassembly Platform for Disassembly of a Plug-In Hybrid Electric Vehicle Battery: A Case Study." *Automation* 5 (2): 50–67. https://doi.org/10.3390/automation5020005.

Quigley, Morgan, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. 2009. "ROS: an open-source Robot Operating System." In *ICRA Workshop on Open Source Software*, .

Rana, Krishan, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. 2023. "SayPlan: Grounding Large Language Models using 3D Scene Graphs for Scalable Robot Task Planning." In *Proceedings of The 7th Conference on Robot Learning*, edited by Jie Tan, Marc Toussaint, and Kourosh Darvish, Vol. 229 of *Proceedings of Machine Learning Research*, 06–09 Nov, 23–72. PMLR.

Rasheed, Adil, Omer San, and Trond Kvamsdal. 2020. "Digital Twin: Values, Challenges and Enablers From a Modeling Perspective." *IEEE Access* 8: 21980–22012. https://doi.org/10.1109/ACCESS.2020.2970143.

Ritchie, Hannah. 2024. "Tracking global data on electric vehicles." https://ourworldindata.org/electric-car-sales.

Sandrini, Samuele, Marco Faroni, and Nicola Pedrocchi. 2025. "Learning and planning for optimal synergistic human–robot coordination in manufacturing contexts." *Robotics and Computer-Integrated Manufacturing* 95: 103006. https://doi.org/10.1016/j.rcim.2025.103006.

Scholz, Constantin, Hoang-Long Cao, Emil Imrith, Nima Roshandel, Hamed Firouzipouyaei, Aleksander Burkiewicz, Milan Amighi, et al. 2024. "Sensor-enabled safety systems for human-robot collaboration: A review." *IEEE Sensors Journal* .

Simonič, Mihael, Rok Pahič, Timotej Gašpar, Saeed Abdolshah, Sami Haddadin, Manuel G. Catalano, Florentin Wörgötter, and Aleš Ude. 2021. "Modular ROS-based software architecture for reconfigurable, Industry 4.0 compatible robotic workcells." In *2021 20th International Conference on Advanced Robotics (ICAR)*, 44–51.

Singh, Ishika, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. 2023. "ProgPrompt: Generating Situated Robot Task Plans using Large Language Models." In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 11523–11530.

Stereolabs. 2024. "Stereolabs ZED2 camera." https://www.stereolabs.com/products/zed-2. Accessed: July 2024.

Stereolabs. 2024. "Zed Ros 2 Wrapper." Accessed: July 2024, https://github.com/stereolabs/zed-ros2-wrapper.

STIIMA-CNR. 2024. "ROS 2 Comau Driver." Accessed: July 2024, https://github.com/CNR-STIIMA-IRAS/comau_c5gopen_ros2.

Tan, Wei Jie, Christina Mei Mei Chin, Akhil Garg, and Liang Gao. 2021. "A hybrid disassembly framework for disassembly of electric vehicle batteries." *International Journal of Energy Research* 45 (5): 8073–8082. https://doi.org/10.1002/er.6364.

TM ROS Driver. 2024. "TM ROS Driver." Accessed: July 2024, https://github.com/TechmanRobotInc/tmr_ros2.

Tonola, Cesare, Marco Faroni, Manuel Beschi, and Nicola Pedrocchi. 2023. "Anytime Informed Multi-Path Replanning Strategy for Complex Environments." *IEEE Access* 11: 4105–4116. https://doi.org/10.1109/ACCESS.2023.3235652.

Verma, Janardan Kumar, and Virender Ranga. 2021. "Multi-robot coordination analysis, taxonomy, challenges and future scope." *Journal of intelligent & robotic systems* 102: 1–36.

Villagrossi, Enrico, Michele Delledonne, Marco Faroni, Manuel Beschi, and Nicola Pedrocchi. 2023. "Hiding task-oriented programming complexity: an industrial case study." *International Journal of Computer Integrated Manufacturing* 36 (11): 1629–1648. https://doi.org/10.1080/0951192X.2023.2203676.

Villagrossi, Enrico, and Tito Dinon. 2023. "Robotics for electric vehicles battery packs disassembly towards sustainable remanufacturing." *Journal of Remanufacturing* 13: 355–379. https://doi.org/10.1007/s13243-023-00134-z.

Viso, Mikel Bueno, Jingjing Huang, Seemal Asif, Fahad Khan, and Phil Webb. 2024. "Towards Robot Software Abstraction: ROS 2-Based Framework for Object Handling within a Robot Cell." In *2024 IEEE 22nd International Conference on Industrial Informatics (INDIN)*, 1–8. IEEE.

Wang, Huanqing, Kaixiang Zhang, Keyi Zhu, Ziyou Song, and Zhaojian Li. 2023. "ABatRe-Sim: A Comprehensive Framework for Automated Battery Recycling Simulation." In *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1–8. IEEE.

Wegener, Kathrin, Stefan Andrew, Annika Raatz, Klaus Dröder, and Christoph Herrmann. 2014. "Disassembly of Electric Vehicle Batteries Using the Example of the Audi Q5 Hybrid

System." *Procedia CIRP* 23: 155–160. 5th CATS 2014 - CIRP Conference on Assembly Technologies and Systems, https://doi.org/10.1016/j.procir.2014.10.098.

Wegener, Kathrin, Wei Hua Chen, Franz Dietrich, Klaus Dröder, and Sami Kara. 2015. "Robot Assisted Disassembly for the Recycling of Electric Vehicle Batteries." *Procedia CIRP* 29: 716–721. The 22nd CIRP Conference on Life Cycle Engineering, https://doi.org/10.1016/j.procir.2015.02.051.

William Woodall. 2019. "ROS on DDS." Available at: `https://design.ros2.org/articles/ros_on_dds.html`.

Yin, Hao, Jinhua Xiao, and Guoxian Wang. 2022. "Human-Robot Collaboration Re-Manufacturing for Uncertain Disassembly in Retired Battery Recycling." In *2022 5th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, 595–598.

Yuan, Gang, Xiaojun Liu, Chaoyong Zhang, Duc Truong Pham, and Zhiwu Li. 2023. "A new heuristic algorithm based on multi-criteria resilience assessment of human–robot collaboration disassembly for supporting spent lithium-ion battery recycling." *Engineering Applications of Artificial Intelligence* 126: 106878. https://doi.org/10.1016/j.engappai.2023.106878.

Yumbla, Francisco, Marcelo Fajardo-Pruna, Anthonny Piguave, Diego Ronquillo, Ricardo Ortiz, Jongseong Choi, Gabriel Diaz, Xabiel G Pañeda, and Hyungpil Moon. 2025. "An Open-Source Multi-Robot Framework System for collaborative environments based on ROS2." *IEEE Access* .