PROJECT NO

101120731

PROJECT ACRONYM

MAGICIAN

PROJECT TITLE:

IMMERSIVE LEARNING FOR IMPERFECTION DETECTION AND REPAIR THROUGH HUMAN-ROBOT INTERACTION

CALL/TOPIC:

HORIZON-CL4-2022-DIGITAL-EMERGING-02-07

START DATE OF PROJECT:

01.10.2023

DURATION:

48 MONTHS

DELIVERABLE D4.2

Grinding Robot Solution

DUE DATE OF DELIVERABLE:

30.06.2025

ACTUAL SUBMISSION DATE:

15.07.2025

Work Package	**WP4 - Robotic platform and interfaces**
Associated Task	**T4.1, T4.2, T4.3, T4.4, T4.5, T4.6,**
Deliverable Lead Partner	**IIT**
Main author(s)	**Nikolaos Tsagarakis, Luca Muratore, Gionata Salvetti, Michele Pompilio, Domenico Prattichizzo, Luigi Palopoli, Geert Driessen, Antonis Argyros, Amamr Qammaz, Iason Oikonomidis, Veronica Campana, Songqun Gao, Jorge Gustavo Perez Fuentevilla**
Internal Reviewer(s)	**Daniele Fontanelli**
Version	**1.0**

DISSEMINATION LEVEL		
PU	Public	X
SEN	Sensitive - limited under GA conditions	

CHANGE CONTROL

DOCUMENT HISTORY

VERSION	DATE	CHANGE HISTORY	AUTHOR(S)	ORGANISATION
0.1	10.07.2025	First Draft	Nikolaos Tsagarakis, Luca Muratore, Gionata Salvetti, Michele Pompilio, Domenico Prattichizzo, Luigi Palopoli, Geert Driessen, Antonis Argyros, Amamr Qammaz, Iason Oikonomidis, Veronica Campana, Songqun Gao, Jorge Gustavo	UNITN

Co-funded by
the European Union

			Perez Fuentevilla	
0.2	12.07.2025	Internal Review	Daniele Fontanelli	UNITN
1.0	14.07.2025	Final Version	Luigi Palopoli	UNITN

> *Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.*
>
> *This deliverable is part of a project that has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no. 101120731.*

EXECUTIVE SUMMARY

This document accompanies the release of the first prototype of the project's robotic platform. The release was completed during two integration weeks: the first one in April 2025 took place at the in the IIT premises in Genova, the second during June 2025 in Trento. As described in the **DoA (Description of Action)**, the platform comprises two embodiments:

- The **Sensing Robot (SR)**: Used to analyse car bodies in search of defects.
- The **Cleaning Robot (CR)**: Used to automate the removal of defects.

While the title might imply otherwise, this document covers the technological solutions for both the SR and the CR (the grinding robot). These two robotic solutions share significant hardware and software components. They are both based on the **Doosan H2515 cobot manipulator** and are operated through a unified software infrastructure built upon **ROS2** and **Xbot2 middleware**. They also utilise shared solutions to ensure adequate safety levels and facilitate interaction between the robot and the car's chassis.

Other components are specific to each solution. These include the design of the end effector, which integrates the sensing devices for the SR and the grinding tool for the CR, as well as the task and motion planning components.

This deliverable is a natural follow-up to **D4.1**, which reported the initial design choices behind the different components. For this reason, we will adhere to the same section breakdown as D4.1.

For each section, we will:

1. Summarise the design choices described in D4.1.
2. Highlight any changes in the design (where applicable).
3. Report on the path to implementation for each component.

The main achievements and findings associated with the work progress performed in **Work Package 4 (WP4)** and reported in D4.2 include the following:

- **Tactile Perception Modules:** The tactile perception modules designed in previous project phases and described in [Deliverable of WP3 on tactile perception] have been developed, tested in isolation, and integrated into the

first version of the prototype. This integration includes the visual components described in [Deliverable of WP3 on visual perception]. The entire perception package is now integrated into a single hand effector and is currently being tested in a laboratory environment.

- **Defect Removal:** The defect removal module is based on a specialised hardware for applying a professional grinding tool to the end effector of the robot. The mechanics of the component have been designed, with a special care for a vibration suppression mechanism which is currently in the prototyping phase.
- **Interaction Control:** The first version of control methodologies for providing adaptive robot interactions has been fully implemented and tested, using the robot in various human-robot and environment-robot scenarios.
- **Software Infrastructure:** A comprehensive software infrastructure based on the ROS2 and Xbot-2 middleware has been developed, tested, and deployed on the prototypes.
- **Planning and Scheduling Algorithms:** Planning and scheduling algorithms and tools have been explored and developed to optimise the robot's defect detection and reworking tasks. These algorithms have been implemented, and their robustness and scalability tested.

The material presented in this report is naturally complemented by the content of Github repository of the project, which collects the code and the design document associated with each component of the system.

DEVIATIONS

No deviations to report.

TABLE OF CONTENTS

LIST OF TABLES

NO TABLE OF FIGURES ENTRIES FOUND.

LIST OF FIGURES

LIST OF ABBREVATIONS

ACRONYM	DESCRIPTION
D	Deliverable
EC	European Commission
WP	Work package
WT	Work task
CR	Cleaning robot
SR	Sensing robot

1 INTRODUCTION

This deliverable presents the progress made during the first 18 months of the MAGICIAN project on the development and integration of a robotic platform. The platform is realised in two distinct embodiments: one for **defect detection (Sensing Robot - SR)** and one for **reworking processes (Cleaning Robot - CR)**.

The interfaces of the robotic platform, both with the environment and the human user, were initially presented in D4.1 (a document frequently referenced in this write-up) and have now been fully developed, tested, and integrated within the platform.

Key challenges addressed during our development activities include:

- The design of intuitive interfaces for precise defect detection.
- The implementation of impedance and force control methods for enhanced safety and adaptability.

Additionally, this deliverable presents the first version of the planning and scheduling algorithms designed to optimise task sequencing for both defect detection and reworking, thereby ensuring efficient and safe operations in dynamic environments.

Overall, the artefacts presented in this document form the technological basis that will be used throughout the first experimentation phase at the consortium end-user's premises. This phase is crucial for gathering feedback to inform subsequent developments.

The platforms described in this deliverable have been validated and initially tested during the two integration weeks that took place at the IIT premises of Genova (April 2025) and at the UNITN premises in Trento (June 2025).

1.1 PURPOSE AND SCOPE

The purpose of this deliverable is to describe the initial integrated robotic platform developed within the project. We particularly focus on human-robot and environment-robot interfaces, end-effectors, and control methodologies. These components are essential for enabling the robot to operate in its two versions: **defect detection** and **defect removal**. Both robot embodiments share many crucial design choices – primarily, they are based on the same **Doosan H265 cobot manipulator** – and utilise a significant portion of the software infrastructure and codebase required for operation.

The **human-robot interfaces** developed include both a wearable and a handheld tactile device, the latter used to test the touch-based sensing algorithm. The wearable device has been designed with ergonomics in mind, allowing for testing large areas of a vehicle body while also facilitating high-precision testing operations on small surfaces.

The information collected with the handheld device has proven essential for developing the **tactile sensor**, which has been integrated within the end-effector of the **Sensing Robot (SR)**. The same device can be used to perform visual inspection, tactile inspection, or combinations thereof. Likewise, we have implemented an **end-effector for mounting the grinding tool**. This component is equipped with a **vibration suppression mechanism** that enables the collection of fine-grained force sensing while grinding is in process, thereby allowing for the application of force and impedance control algorithms.

In terms of **control methodologies**, we have developed and tested impedance and force control strategies. These strategies enable the robot to adapt to varying external forces, ensuring safety and precision in dynamic environments. Impedance control is further enhanced by introducing a **gravity-compensated teaching mode**, which facilitates intuitive human interaction for pose or trajectory teaching, with the robot remaining compliant with external forces while compensating for gravity.

To support the real-time execution of these components, we have developed a **low-latency and time-predictable interface** based on the **Xbot2 real-time middleware**. These real-time components are part of a larger architecture of interconnected components communicating through **ROS2**.

This deliverable also discusses the development of **task planning and scheduling algorithms** that enable the robotic platform to prioritise and execute defect detection and reworking tasks efficiently. The deliverable highlights **roadmap-based scheduling techniques**, **dynamic motion primitives**, and **trajectory adjustments**, laying the groundwork for a fully autonomous robotic system that enhances efficiency and safety in industrial operations.

This document describes a fully integrated deliverable, which is currently being tested and used in close simulation of the operational environment. The purpose of this testing is to collect usable feedback to guide the subsequent system refinement phases.

1.2 CONTRIBUTION TO PROJECT OBJECTIVES

This write-up details work that is central to the MAGICIAN project. It's therefore no surprise that the achievements outlined in this deliverable are essential for reaching the project's main objective. Let's look at the details:

Scientific and Technological Objectives

- O1: A robotic perception module integrating visual and tactile sensors for defect analysis and classification.
- O2: A robotic cleaning module with a specialised end-effector for defect reworking.
- O3: A software robotic platform integrating services for perception and

cleaning modules.

- O4: A closed-loop defect detection and avoidance system for robotic and welding processes.
- O5: Development of two TRL 7 integrated prototypes for defect analysis and reworking.

Social Sciences and Humanities (SSH) Objectives

- O6: A human-centred approach to human-robot collaboration, promoting usability, safety, and trustworthiness.

Demonstration Objectives

- O7: Demonstration of the prototypes in operational scenarios.
- O8: Expansion of MAGICIAN's scope and applicability via Financial Support to Third Parties (FSTP).

The development of human-robot interfaces and control methodologies outlined in this deliverable is pivotal to the MAGICIAN platform's capabilities. These advancements empower the platform to autonomously undertake defect detection, cleaning, and reworking operations. The implementation of impedance and force control systems significantly improves safety and adaptability during intricate industrial processes, whilst the planning and scheduling algorithms guarantee optimal task execution. These contributions are in fulfilment of the overarching project objectives of providing a highly automated, adaptable, and efficient system for defect handling and reworking, ultimately progressing the automation of industrial processes and showcasing the MAGICIAN platform's effectiveness in operational environments.

1.3 RELATION TO OTHER WORK PACKAGES

The work described here contains some of the key achievements of WP4, whose main purpose is to define and implement the robotic platform and its interfaces, developing the control, planning and scheduling algorithms for both the **Cleaning Robot (CR)** and the **Sensing Robot (SR)**. A synthetic list of the most important relations is offered next.

- **WP2 – Use case definition and platform design:** The design and the implementation of the robotic platform detailed in this report is a direct emanation of our understanding of the requirements of the automotive use case, which is the main focus of the project. The challenges imposed by the use-case were many and multi-faceted and required a huge development activity in the areas of end-effector design, reworking effectiveness, safety, integration with the perception module, and time and cost constraints.

- **WP3 – Data acquisition and skills learning:** WP3 has been developing a package of reliable and robust solutions for the perception component of the system, which are directly implemented in the robotic architecture presented in this deliverable. Specifically, planning and scheduling rely on defect analysis and predictions of human operator movements (T3.1). Motion control and active sensing directly take as input the outcome of the perception pipeline detailed in T3.2. The robot motion strategies exploit the knowledge on human behaviour and the related ability to predict the future positions of humans developed in T3.3. On the other hand, the relation between WP3 and WP4 is bidirectional. Indeed, much of the notion of on the potential of the robotic platform have a deep influence on the way perception strategies are defined and fine-tuned.
- **WP5 - Integration and performance analysis:** The robotic platform described in this deliverable will be the key component of the final platform (T5.1) and included in the demonstrator (T5.2), thus contributing to the project's KPIs.
- **WP6 – Cascade funding management:** The cascade funding schemes provide improvements and extensions of the platform as well as new use-case. Therefore, the knowledge on the hardware/software solution behind the platform are directly used by both the proposers of additional modules and by the new use-case proposers. On the other hand, the new component and the new use-cases considered in WP6 will eventually be integrated into the final version of the platform making it more reliable and flexible. Finally, as the robotic platform will be utilised, the WP findings will be essential in providing support and technical assistance (T6.4).

1.4 STRUCTURE OF THE DOCUMENT.

This document follows roughly the structure of D4.1. It is therefore organised in seven chapters, each one covering a different aspect of the robotic platform. For each chapter, we summarise the key design decision taken in the early phases of the project and detailed in D4.1 and report about changes, deviation and implementation status. The final section outlines future development in the integration of the platform such as can result from deferred activities, and possible design iterations resulting from the experimentation phase.

Chapter 2 focuses on the development of the human-robot interfaces, consisting of a tactile device that can be held or worn and used to acquire datasets on tactile inspection, which can be used for two different purposes: test the ability of sensing tactile sensing devices, which can be later integrated into the robot perception module, and collect observations on the strategies of human operators within an imitation learning framework.

Chapter 3 covers the design and implementation of the end effector used for the grinding and the sensing operations.

Chapter 4 covers the control methodologies, including impedance and motion control for safe and adaptive robot interactions. It also quickly describes the real--time software infrastructure used for the project and based on the XBot-2 control framework.

Chapter 5 describes the planning and scheduling algorithms developed to optimise the robot's defect detection and reworking tasks, while guaranteeing adequate levels of safety for the human operators working nearby the robot.

Chapter 6 outlines the future direction of WP4.

A list of references used throughout the document is reported in Chapter 7, which covers the scientific and technological foundations that support the research and development in WP4.

1.5 SYSTEM OVERVIEW

The MAGICIAN project's robotic system centres on a collaborative platform, specifically the **Doosan H2515 cobot**. This robot is engineered for tasks demanding both precision and safety, offering advanced capabilities for **defect detection and reworking**. The Doosan H2515 **features** six-axis control, an **extended reach of 1500 mm**, a **payload capacity of 25 kg**, and a **repeatability of 0.1 mm**, making it perfectly suited for complex industrial tasks. Crucially, its advanced safety features, including **six load cells for real-time force sensing**, ensure minimal contact forces and high precision during operation, significantly boosting both safety and performance in collaborative environments.

Additionally, the system incorporates **tactile sensors** into both wearable and handheld devices. These sensors enable the robot to acquire detailed surface information and autonomously replicate human-like movements during defect detection and cleaning operations.

1.5.1 REQUIREMENTS AND SPECIFICATIONS

The system requirements for the robotic platform emphasize safety, precision, adaptability, and integration. The Doosan H2515 robot was selected for its robust capabilities in handling industrial tasks, including defect detection and reworking. Key specifications include:

- **Payload capacity**: 25 kg, allowing the robot to handle a variety of tools and parts.
- **Reach**: 1500 mm, enabling the robot to cover large surface areas.
- **Repeatability**: 0.1 mm, ensuring high precision in tasks such as grinding and defect cleaning.
- **Safety features**: Six-axis force sensors ensuring a contact force of 0.2 N, guaranteeing high sensitivity in defect detection.
- **Communication protocols**: Ethernet (TCP/IP), ModBUS, and Profinet IO, providing flexible connectivity options.
- **Programming**: The cobot supports intuitive block-based programming and pre-configured routines, simplifying the setup for different tasks. Additionally, the robot's flange is equipped with connectors (6+6 I/O) for secure and efficient

tool integration.

- **Compliance**: The system adheres to international safety standards, including EN ISO 13849-1 and EN ISO 10218-1.

These specifications ensure the platform can be easily integrated with other hardware and systems developed in the project while maintaining compliance with industry standards.

2 HUMAN-ROBOT INTERFACES

2.1 SUMMARY OF HRI DESIGN

The **Human-Robot Interface (HRI)** is designed to enhance defect detection and reworking in car-body manufacturing using advanced robotics and sensing. A core aspect is building a comprehensive **tactile dataset** to improve defect classification and reworking accuracy. To this end, we've developed **two haptic interfaces: one wearable and one handheld**. These interfaces are engineered to realistically capture the interaction forces and accelerations experienced by workers during inspection, without impeding their ability to detect and classify defects. Despite using basic sensors, these interfaces have shown promising results in defect detection and identification, providing detailed force data.

Beyond the haptic interfaces, the **grinder tool** for defect reworking is equipped with force/torque sensors. This setup facilitates a **learning-by-demonstration** approach, allowing the cleaning robot to learn proper reworking procedures from human operators. Both the defect detection and grinding tools will incorporate **fiducial markers** for precise trajectory tracking. Post-deployment, a **haptic ring** will provide discrete feedback to operators regarding the outcomes of post-reworking inspections and verifications.

The **tactile sensing interface** aimed at being both user-friendly for human operators and readily adaptable for robotic integration. A key design principle is to ensure the system **seamlessly integrates** into existing workflows for defect detection, avoiding any disruption or added complexity.

At the same time, the interface is designed for **modularity**, a crucial requirement for enabling robots to "learn" from human expertise by replicating the tactile signals and feedback mechanisms used by operators for surface defect detection and refinement.

Our biggest challenging in the design of the system was to capture the **nuanced forces and vibrations** upon which operators rely, while maintaining sufficient flexibility for application across various environments and robotic systems.

2.1.1 TACTILE PERCEPTION MODULE

The **tactile sensors** from Deliverable 3.1 are integrated into two distinct devices: a **wearable tactile device** and a **handheld tactile device**, each designed for different user needs.

The **wearable device**, worn on the hand with palm-placed sensors, enables natural scanning movements akin to human tactile surface exploration. It features a probe with **interchangeable, textured tips** (Figure 1), enhancing defect detection by simulating varied sensations. These 8mm-radius tips are **3D printed from PLA** for durability, ensuring no surface damage and clear transmission of force and vibration signals.

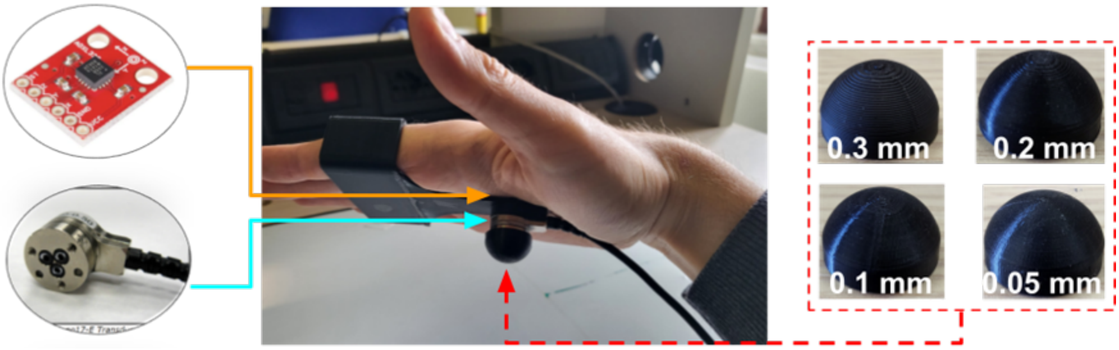


Figure 1: The wearable interface is equipped with a PLA tip, which can be 3D printed with varying texture resolutions (0.3, 0.2, 0.1, and 0.05 mm). These different resolutions are designed to enhance the magnification of tactile signals, allowing for more precise.

The second device is a **handheld tool**, allowing the operator slightly modified scanning movements compared to freehand exploration, while still offering freedom of movement (Figure 2).

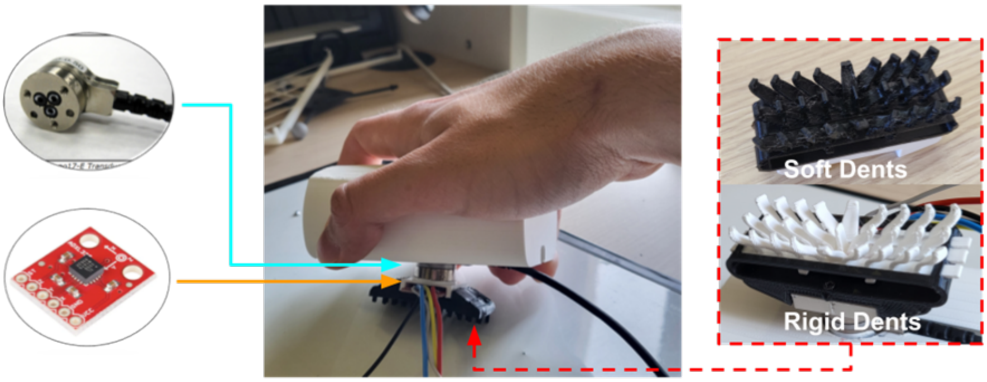


Figure 2: The handheld interface equipped with a scallop component. The scallop is 3D printed using a combination of ABS and TPU materials to ensure effective defect detection while maintaining optimal contact between the surface and the scallop.

This handheld approach provides greater flexibility in probe design. For example, a **scallop-shaped probe** was selected for this design. This shape simultaneously increases the surface area that can be scanned in a single acquisition and miniaturises the contact point with the surface. Such a design is particularly advantageous for detecting smaller, more subtle defects, thereby enhancing the overall effectiveness of the tactile scanning process.

2.1.2 INTEGRATION IN THE HUMAN-ROBOT INTERFACE

The **tactile perception module** has been designed with a focus on **modularity**, ensuring seamless integration across the various interfaces required by the MAGICIAN project. This adaptability makes the system usable across diverse defect detection applications in both human-operated and robotic systems.

At its core, the module is engineered to be compatible with a wide range of **end-effectors** and can be mounted on multiple interfaces, from **wearable devices** for human operators to **robotic platforms** for automated inspection (Figure 3). Furthermore, the tactile module is designed to meet the specific requirements of each scenario, such as the precision needed for human-driven inspections or the robustness necessary for robotic operations. This versatility makes it crucial for achieving the project's goal of transferring human expertise in defect detection to robotic systems, by enabling robots to replicate human tactile feedback.

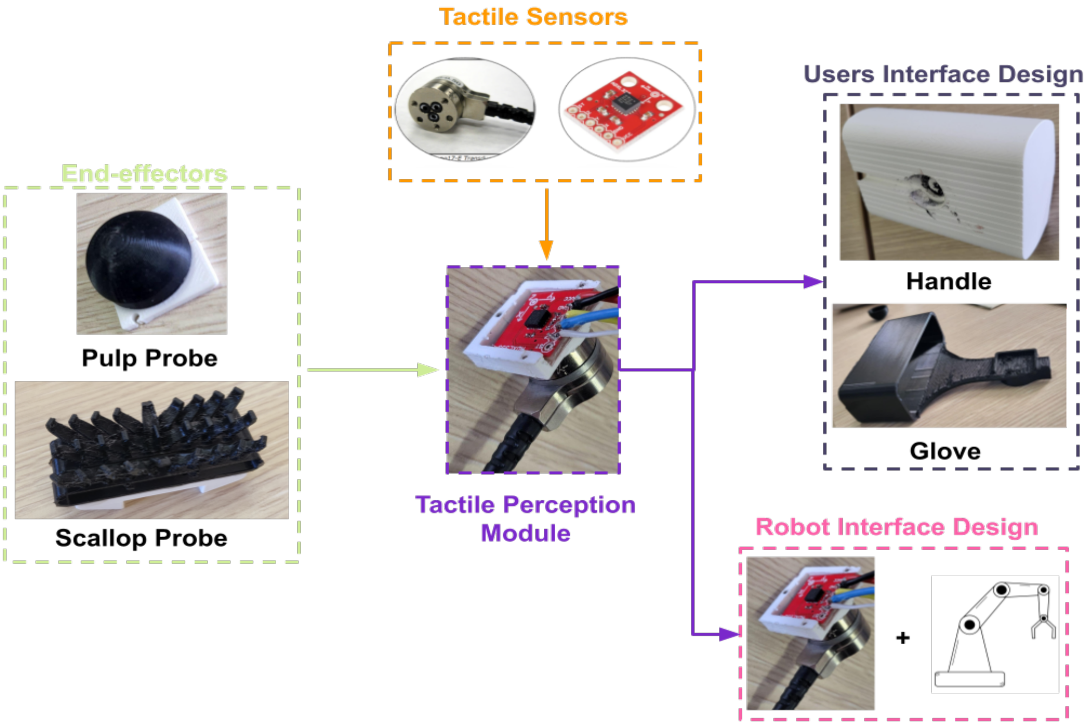


Figure 3: The tactile system is composed of various modules, each serving a specific function within the overall setup. The tactile perception module is designed to be versatile, allowing it to be equipped with different end effectors tailored for surface scanning.

2.2 SYSTEM IMPLEMENTATION

The development of the Tactile Perception Module has progressed with a focus on validating the interface design and assessing its integration capabilities across multiple use cases. During this phase, particular attention was given to the handheld interface equipped with the scallop-shaped end-effector. This specific configuration was selected as the primary tactile interface for experimentation due to its structural robustness, ease of handling, and geometric suitability for covering large surface areas while ensuring effective point-wise contact during scanning. The modular architecture defined in D4.1 was exploited to enable rapid reconfiguration and reuse of sensing components across interfaces. This modular design facilitated its immediate use in the dataset acquisition activities described in D3.2, where the handheld configuration was employed without modification to the sensing core. Furthermore, the interface was evaluated for integration in two operational modes (Figure 4): (i) in combination with the visual perception system, enabling multimodal defect inspection, and (ii) as a stand-alone module directly mounted on the Sensing Robot. The mechanical and electrical interfaces developed in WP4 allowed straightforward adaptation for both configurations. Overall, the current implementation of the Tactile Perception Module confirms the effectiveness of the interface abstraction and modularisation strategy defined in the early design phase. The scallop-equipped interface served as a reference platform for integration testing and validation, supporting the broader objective of enabling interoperability between human interfaces and design of the end effectors.

2.3 SUMMARY OF THE SR END-EFFECTOR DESIGN

The **SR end-effector** has been designed to integrate both **visual sensing** (described in D3.2) and **tactile sensing**. This approach prepares for the integration of both modalities during system operation.

For certain scenarios, defect analysis may rely solely on a visual scan. In others, tactile exploration will be the only method used. Finally, some situations may require both analyses to be performed sequentially (see D2.1).

2.3.1 THE VISUAL SENSING COMPONENTS

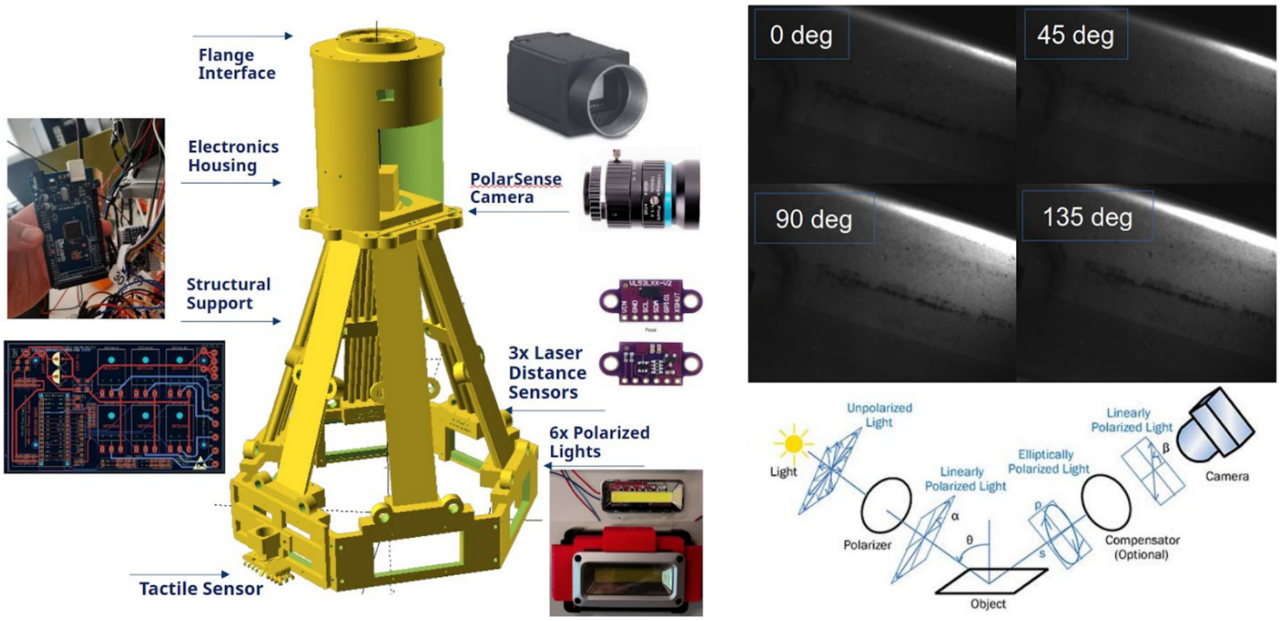


Figure 4: Left: Overview of the Vision End effector and its main components. Right: View of the four polarization channels deinterlaced and underneath them a graph of light polarization propagation principles.

Visual sensing is conducted using the "PolarShadow" vision end effector. The current sensor has a height of 48.5 cm and a weight of 1.85 kg (Figure 4). Its main component is a SONY XCG-CP510 Global Shutter Gigabit Ethernet. Unlike conventional RGB sensors that use Bayer tiling to capture colour information, this camera employs a polarization-sensitive Bayer pattern to simultaneously capture four polarization angles—0°, 45°, 90°, and 135°—at each pixel group (Figure 4). The camera is connected to the MAGICIAN host computer using the ARAVIS SDK, an open-source gigabit camera API that allows raw real-time access to the images received from the sensor. After extensive experiments, the camera lens was updated from a 12mm lens to a 16mm "zoom-lens" with a minimum aperture of F1.4. This setup yields a 14 mm × 10 mm viewing area when the sensor is hovering 5 cm above the observed surface.

The polarised lighting for the physical camera is provided via 6x LED COB lights that feature a polarizer in front of them, which allows only a specific polarization of light to propagate through them. The lights require a 3V 1000 mA power source to be illuminated. An Arduino based board acts as an active lighting switch controlling which of the lights is active at any time. The Arduino board is also connected to 3x VL43LXX-V2 ToF laser range sensors that constantly measure the distance from the observed target. After initial trials with an automated "round-robin" light scheduler, two more lighting modes were implemented. In the first mode, a 3D plane is automatically extracted using the range sensors and where the closest light to the surface is automatically selected to maximize the relative polarization angle between the light and the camera. In the second mode, the MAGICIAN Host computer can select the light ordering based on a strategy derived using machine learning to make the optimal lighting decisions.

A MAGICIAN grabber software suite was developed to drive the sensor, successfully unifying and synchronizing the various data streams from the sensor and performing software light synchronization to the camera shutter. The 2448 x 2048 image frames received from the camera are directly stored in a shared memory buffer allocated using the Linux SHM mechanism. The data from the sensor can either be stored on disk using lossless file formats (pnm, csv) to facilitate creating a training dataset, or be dynamically accessed using the SHM interface by a python interface to directly use the data from the sensor to perform defect detection. The classification neural network we are actively developing using the ResNet18 backbone saturates the 23 Hz image stream received from the camera sensor, which is defined by the bandwidth available by the Gigabit Ethernet bus.

2.3.2 THE TACTILE SENSING COMPONENTS

The end-effectors shown in Figure 3 are designed to **complement the camera system**, especially when visual inspection struggles with defect classification, offers uncertain results, or faces difficulties accessing certain car part areas.

For example, probe tips with varied textures are particularly effective at **detecting hidden defects**, such as those on car part edges. Their precise, localised scanning capability enhances classification in these challenging scenarios. Conversely, the **scallop-shaped end-effectors** are beneficial when the camera system identifies potential defects on larger surfaces with high uncertainty. Here, the scallop design helps **maximise the scanned area**, thereby improving operational efficiency and ensuring more accurate defect detection.

The integration with the visual sensing component had to comply with the following requirements:

- **Limit potential occlusion** introduced by the tactile sensors, which could hamper the operation of the visual sensors.
- **Enable the tactile sensors to operate using different motion patterns**, as may be required by the scenario.
- **Limit potential perturbation** introduced on the force measurement by unnecessary vibration of the structure.

2.4 FIRST IMPLEMENTATION RUN OF THE SR END-EFFECTOR

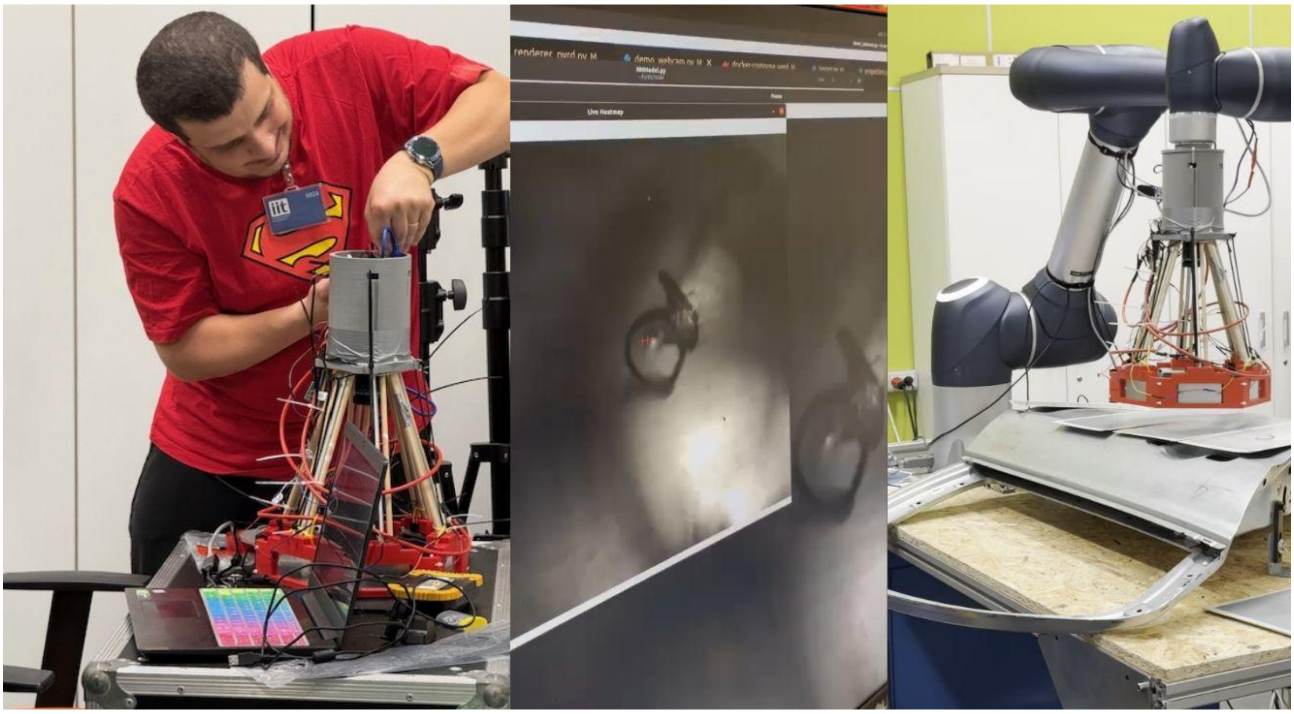


Figure 5: The first prototype of the MAGICIAN SR End-effector mounted on the Doosan robot and performing live defect detection during the 1st integration meeting of the project hosted in IIT.

The first prototype implementation of the MAGICIAN SR End-effector was based on a 3D printed design. Being able to easily 3D print and replace parts made experimentation with different camera configurations more viable and significantly sped-up the development process of the prototype. In order to make development, maintenance and overview of the sensor easier, its cabling was also kept externally using jumper wires that could be easily intercepted to debug the electronics circuit and dynamically rearranged. At the same time, the choice of aluminium supports and frugal use of metal only for the heatsinks resulted in a very low overall weight for the sensor of just 1.85 kg which is an order of magnitude less than what the Doosan robot can easily handle. This first prototype was used to record datasets from shipments of metal parts to FORTH, develop the initial versions of the neural networks, and then also successfully mounted and operated on the robot (Figure 5) during the 1st integration meeting of the project hosted in IIT.

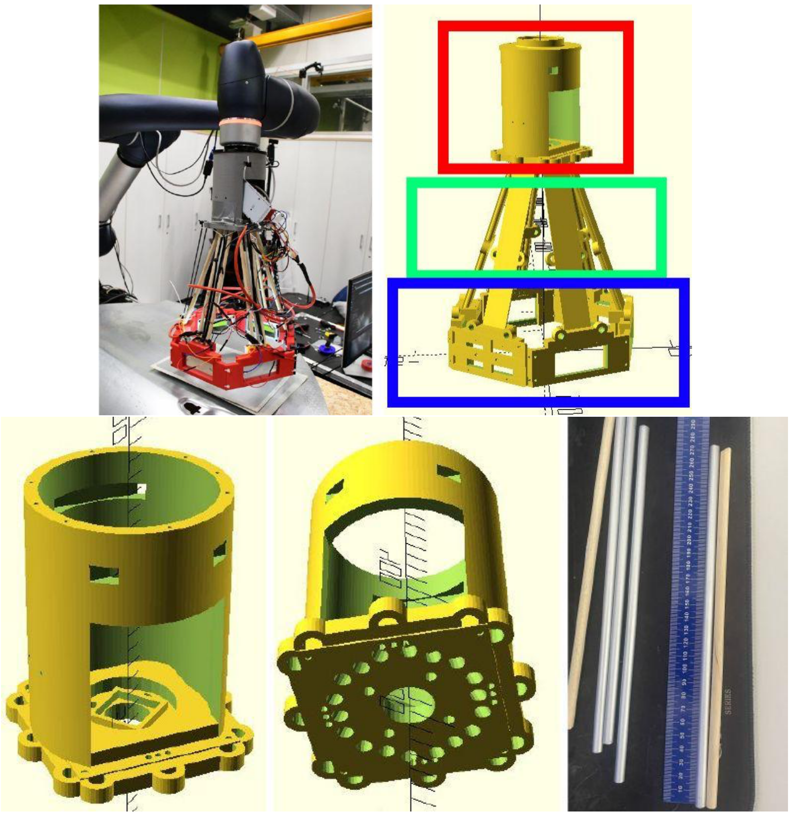


Figure 6: The light holders offer mounting points for the laser range sensors (area marked with red in centre) and can also support the tactile sensor which is attached on them using a “back-pack”.

After thorough testing during the 1st integration week. the initial design of the sensor was found to successfully detect defects when mounted on the Doosan robot. Furthermore, although the sensor had been trained using metal samples shipped to FORTH it was able to generalize to metal examples from IIT that it had never seen before. Class A defects were detected consistently with more than 3/6 available lights while smaller Class B/C defects were more difficult but still detectable especially when testing lower hovering heights than 5 cm. The key improvements for the next version of the sensor that we identified were (see Figure 6 for reference):

- The 3D printed shell should be upgraded to metal to improve its robustness.
- The sensor could be further miniaturized since for example the electronics box was overly large and mostly empty.
- The cabling should become permanent and not be exposed outside of the shell of the device.
- Connectivity between the different modules should happen using standardized “sandwiched” connectors.
- Sharp edges of the design should be minimized to provide passive safety to humans close to the MAGICIAN robot.
- Flashing lights could cause nausea or headaches to people working close to the

sensor, so an opaque cover to the sensor could prevent the lights from being visible from bystanders.

- The sensor design should match the visual appearance of the rest of the Doosan robot and not appear intimidating to the workers.

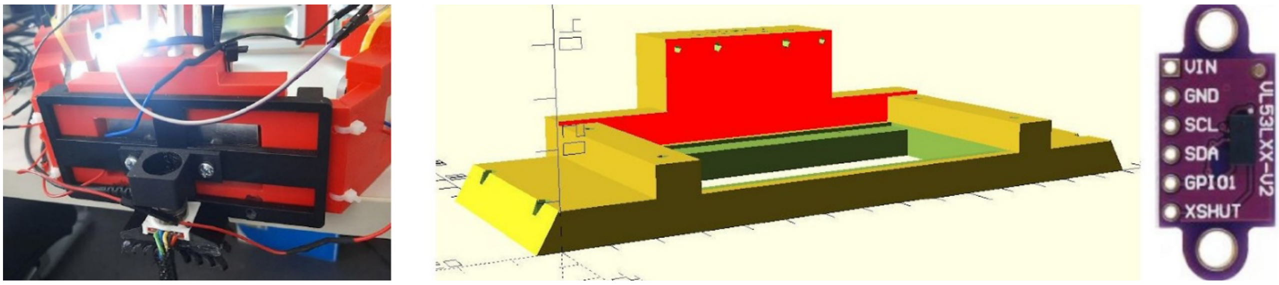


Figure 7: The light holders offer mounting points for the laser range sensors (area marked with red in centre) and can also support the tactile sensor which is attached on them using a "back-pack" mechanism.

The modular design of the Tactile Perception Module has enabled its seamless integration with the visual perception system, resulting in a unified multimodal sensing solution suitable for deployment on the Sensing Robot. To enable the physical coupling of the two systems, a custom-designed mechanical clip was developed (Figure 7). This clip allows the tactile module, equipped with the scallop probe end-effector, to be securely mounted onto the vision system developed by FORTH, preserving alignment and structural integrity during surface exploration tasks (Figure 8). From a software standpoint, the data acquisition framework was extended to support synchronized recording of both tactile and visual data streams. The architecture allows for flexible configuration, enabling the collection of visual-only, tactile-only, or joint visual-tactile datasets. This flexibility is crucial for adapting the sensing strategy to different inspection scenarios. For example, in areas where visual perception may be hindered by surface properties (e.g., reflectivity, shadows, or occlusions), the tactile sensor can provide reliable complementary information. Conversely, in large homogeneous regions where tactile scanning is less efficient, vision can serve as the primary modality. This integration not only facilitates co-localized data acquisition but also unlocks the potential for multimodal learning and inference, laying the foundation for more robust and accurate defect detection. The combined system allows the Sensing Robot to adapt its perception strategy dynamically, leveraging the strengths of each sensing modality based on contextual requirements. This approach aligns with the project's goal of enabling human-like inspection capabilities through the fusion of complementary sensory information.

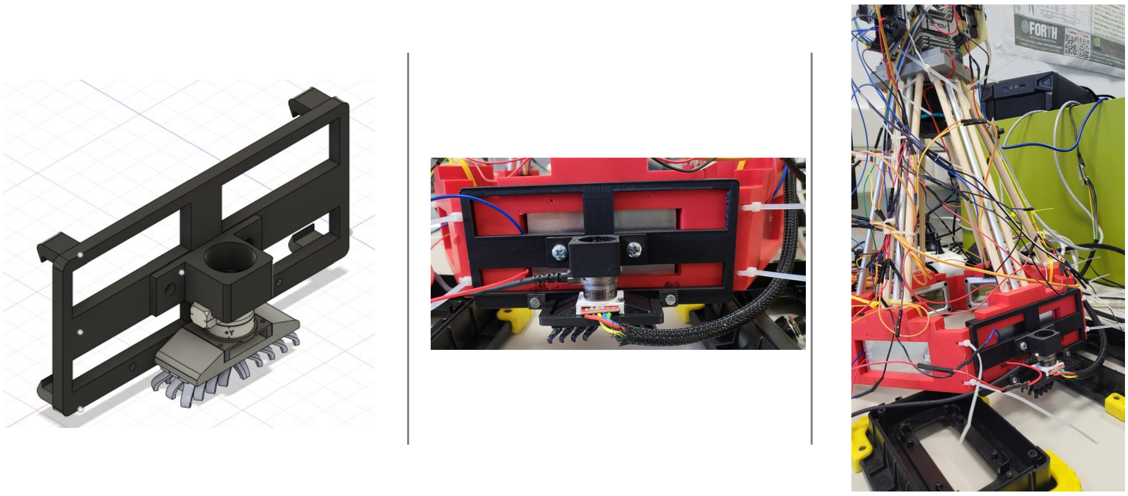


Figure 8: The tactile system has been seamlessly integrated with the camera system, enabling the robot to acquire and process visual and tactile information simultaneously.

2.5 SUMMARY OF THE CR END-EFFECTOR DESIGN

A crucial component of our platform is the interface between the wrist of the manipulator and the grinding tool. The design of the component has in its goals: 1. a substantial reduction of the vibrations, 2. the correct distribution of the contact forces. Both are needed to prevent damages and to improve the accuracy of the vibrations. To meet these challenging goals, we have adopted a co-design approach based on the simultaneous optimisation of hardware and control parameters. The approach required to set-up a physically accurate simulation environment.

During grinding, the robot's end-effector experiences two primary forces: **friction forces**, generated upon contact with the surface, and **vibration forces**, stemming from the grinder's internal dynamics. Friction forces arise when the grinder touches the surface, opposing its motion, and include both normal (perpendicular) and tangential (opposing motion along the contact plane) components. Simultaneously, the grinder itself induces vibrations, varying in amplitude and frequency based on the tool type and operational characteristics. For our model, we simplified these grinder-generated forces, representing them as **sinusoidal forces along the three Cartesian axes (x, y, z)**.

In particular, to simulate grinder vibrations on the robot's end-effector, we assumed **sinusoidal forces** along the x, y, and z Cartesian axes, with a frequency range of **10 to 200 Hz**. This range is based on studies of typical sanding tool vibrations, such as Radwin et al. (1990), which show significant accelerations up to 150 Hz for tools like "palm grip orbital sanders." We extended this to 200 Hz to account for potential variations and ensure comprehensive modelling.

For a common "Palm Grip Orbital" sander at 150 Hz, the frequency-weighted RMS arm accelerations are: X: 25.4 m/s², Y: 30.3 m/s², Z: 45.6 m/s². These accelerations were used to calculate forces on the end-effector using the relationship F=m·a, where 'm' is the combined mass of the end-effector and grinder.

2.5.1 NORMAL FORCE AND TANGENTIAL FORCE (FRICTION)

The normal force F_n is modelled by combining an elastic and a viscous component:

$$F_n = K(\Delta z) + Cv$$

where:

- K is the elastic constant (stiffness) of the surface.
- Δz is the deformation of the surface along the z axis due to contact.
- C is the viscous damping coefficient.
- v is the relative velocity along the z axis.

The tangential friction force F_r is a force acting along the plane tangent to the contact surface and is modelled as:

$$F_r = -k_f\, F_n \frac{v_t}{\|v_t\|}$$

where:

- k_f is a constant describing the relationship between the normal force and the tangential friction force.
- F_n is the normal force calculated as described above.
- $v_t = (v_x, v_y)$ is the tangential relative velocity vector between the grinder and the surface.
- $\|v_t\|$ is the norm of the tangential velocity.

The negative sign indicates that the friction force is always opposite to the direction of the grinder's tangential motion.

2.5.2 SIMULATION SETUP AND ACCURACY OF INTEGRATION METHODS

Simulations were conducted using **Python** and the **Adam library**, chosen for its efficient computation of robot dynamics via automatic differentiation. While derivatives weren't essential here, they could aid future hardware interface co-design. All simulations used the **Doosan H2515 robot model**.

A **PD controller** was implemented with a 1 ms timestep, ensuring high-frequency control updates. The simulation itself ran with a finer 1/16 ms timestep to maintain accuracy and minimise numerical errors.

For end-effector-surface contact, we modelled a **surface stiffness** (Ks) of 105 N/m, typical for a car chassis, and set the **damping coefficient** (Kd) to Ks. The end-effector's reference trajectory was a **sine wave with varying amplitudes**, designed to test the robot's smooth motion tracking.

The accuracy of each test was measured as the difference between the trajectory calculated with a certain time step and the ideal trajectory, obtained with the lowest time step (1/64 ms). This difference is expressed in terms of the infinity norm of the error between the two trajectories.

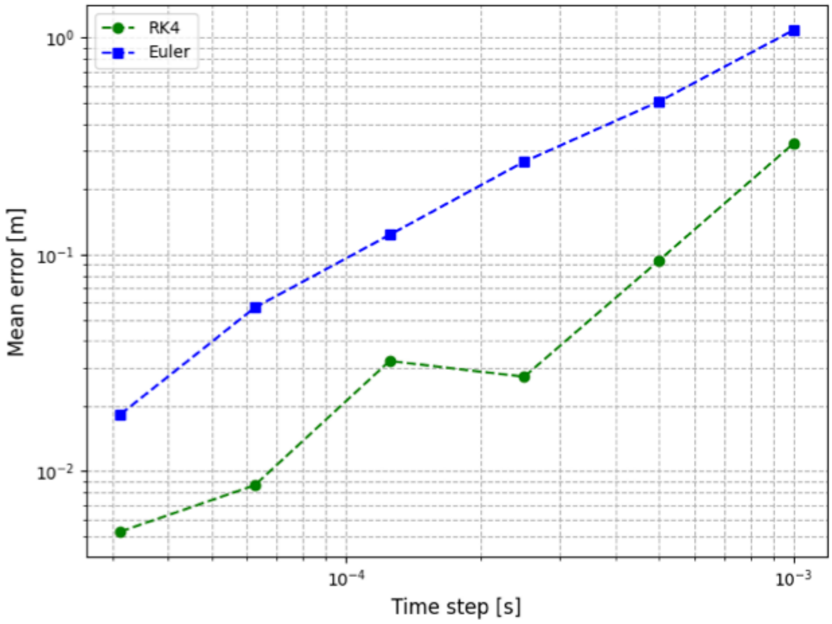


Figure 9: Integration error as a function of integration time step for the Euler and RK4 numerical integration methods.

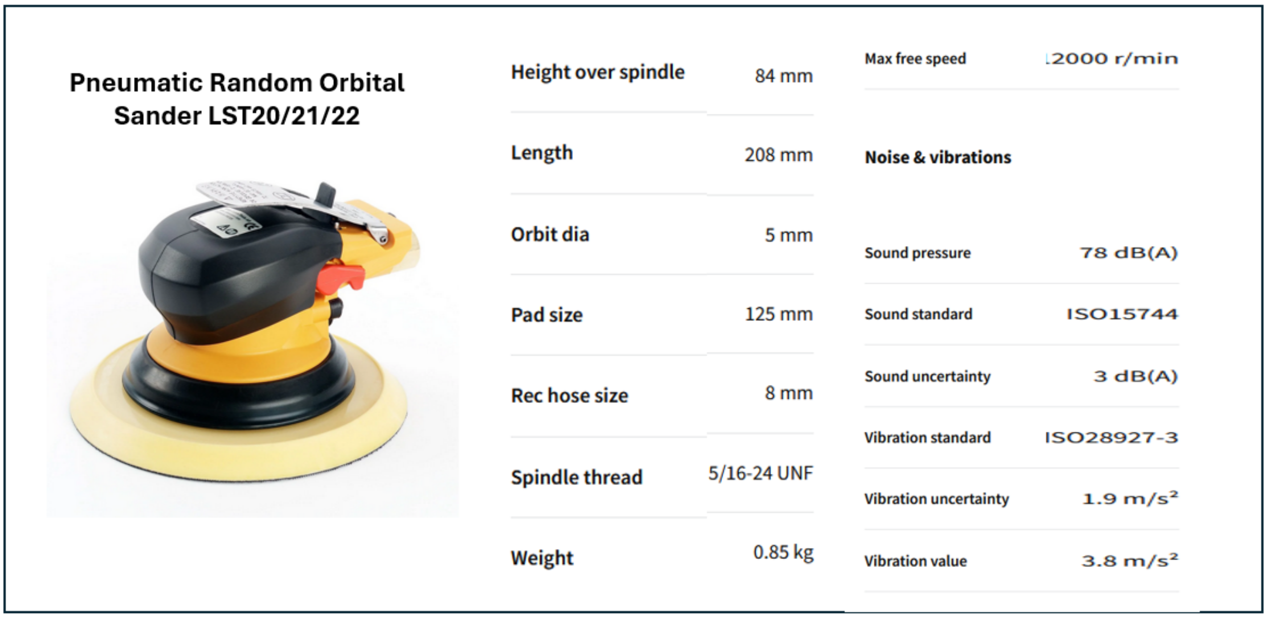
Our results, depicted Figure 9, show that when using the typical integration time step of 1 ms, the resulting integration error is about 1 for Euler and 0.3 for RK4. These errors are both too large for considering the simulation results reliable. The plot shows that to get to a simulation error below 0.01, which we could consider low enough for our purposes of co-design, we should use an integration step of 1/16 ms with RK4, and lower than 1/32 ms with Euler. Better results could be obtained using more sophisticated techniques.

2.6 FIRST IMPLEMENTATION RUN OF THE CR END-EFFECTOR

2.6.1 HARDWARE SOFTWARE DESIGN

2.6.1.1 GRINDING MODULE SELECTION

Starting from the grinding tool model suggested by the end-user (Figure 10), alternative models were evaluated. The main motivation was to identify a full electrical grinder that would eliminate the need for a pneumatic compressor needed for the originally suggested tool.



Pneumatic Random Orbital Sander LST20/21/22

Height over spindle	84 mm
Length	208 mm
Orbit dia	5 mm
Pad size	125 mm
Rec hose size	8 mm
Spindle thread	5/16-24 UNF
Weight	0.85 kg

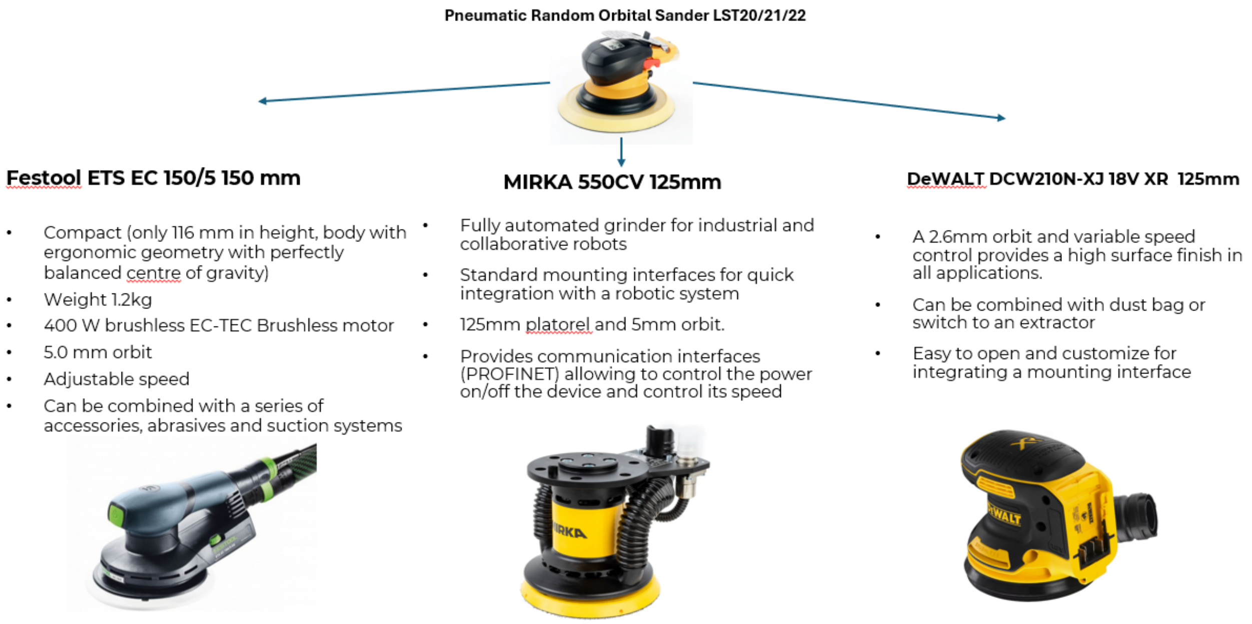
Max free speed	12000 r/min
Noise & vibrations	
Sound pressure	78 dB(A)
Sound standard	ISO15744
Sound uncertainty	3 dB(A)
Vibration standard	ISO28927-3
Vibration uncertainty	1.9 m/s²
Vibration value	3.8 m/s²

Figure 10: The pneumatic orbital grinding tools originally proposed by the end-user.

Within this activity the following three models were evaluated (Figure 11):

- Festool ETS EC 150/5 150 mm
- MIRKA 550CV 125mm
- DeWALT DCW210N-XJ 18V XR 125mm

The above models represent fully electrical grinder/sander tools with specifications and form factors close to the pneumatic grinding tools proposed by the end-user. The specifications of these alternative electrical grinding tools are presented below.



Pneumatic Random Orbital Sander LST20/21/22

Festool ETS EC 150/5 150 mm

- Compact (only 116 mm in height, body with ergonomic geometry with perfectly balanced centre of gravity)
- Weight 1.2kg
- 400 W brushless EC-TEC Brushless motor
- 5.0 mm orbit
- Adjustable speed
- Can be combined with a series of accessories, abrasives and suction systems

MIRKA 550CV 125mm

- Fully automated grinder for industrial and collaborative robots
- Standard mounting interfaces for quick integration with a robotic system
- 125mm platorel and 5mm orbit.
- Provides communication interfaces (PROFINET) allowing to control the power on/off the device and control its speed

DeWALT DCW210N-XJ 18V XR 125mm

- A 2.6mm orbit and variable speed control provides a high surface finish in all applications.
- Can be combined with dust bag or switch to an extractor
- Easy to open and customize for integrating a mounting interface

Figure 11: The electrical grinding tools studied as alternative to the pneumatic grinding tool (on the top) originally suggested.

The **Festool ETS EC 150/5 150 mm** is a compact sander with only 116 mm in height body, providing ergonomic geometry with perfectly balanced center of gravity. The device has a weight of 1.2kg and powered by a 400 W brushless EC-TEC Brushless motor, offering also 5.0mm orbit and adjustable speed. It can be combined in series with accessories, of abrasives and suction systems.

The MIRKA 550CV 125mm is a fully automated grinder for industrial and collaborative robot. It provides 5mm orbit and a standard mounting interface for quick integration with a robotic system. This grinder also offers a communication interface (PROFINET) allowing the user to control the power ON/OFF of the device and control its speed.

Finally, the **DeWALT DCW210N-XJ 18V XR 125mm** is a 2.6mm orbit grinding tool, which provides variable speed control and high surface finish in all applications. This grinder can be combined with dust bag or switch to an extractor. Finally, is easy to open and customize for integrating a mounting interface.

Considering the specifications and features of the three grinding tools studied it can be concluded that the MIRKA 550CV 125mm is the only automated grinder in the market that enables its quick integration and control on a robotic system. Its high price (approximately 9K€) compromised its selection in MAGICIAN. FESTOOL and DeWALT grinders mainly differ in the following specifications: Orbital diameter (5.0mm vs 2.6mm) and Max speed (10000rpm vs 12000rpm) that are not considered critical specification differences for the MAGICIAN application.

Both devices will therefore be evaluated with the DeWALT DCW210N-XJ 18V XR 125mm finally selected given that it provides more convenient access / opening of its body that facilitates its customization and adaptation for integrating it in the MAGICIAN grinding robot.

2.6.1.2 GRINDING MODULE CUSTOMIZATION

To enable the integration of the grinding tool on the robotic system the body of the selected grinder was modified to add the following two features.

A relay switch was integrated inside the body of the grinding tool to permit the ON/OFF control of the device through a dedicated electrical interface connected to the control unit of the robotic system.

This permits the ON/OFF state to be set through the Doosan internal digital interface permitting the coordinated control of the device ON/OFF state with that of the robot motion/actions. The connection of the tool to the interface of the Doosan robot is enabled through a connector mounted on the body of the grinding tool. Finally, the top side of the tool body was modified to integrate the mounting interface, which permits to fix the grinding tool on the wrist of the robotic arm (see Figure 12).

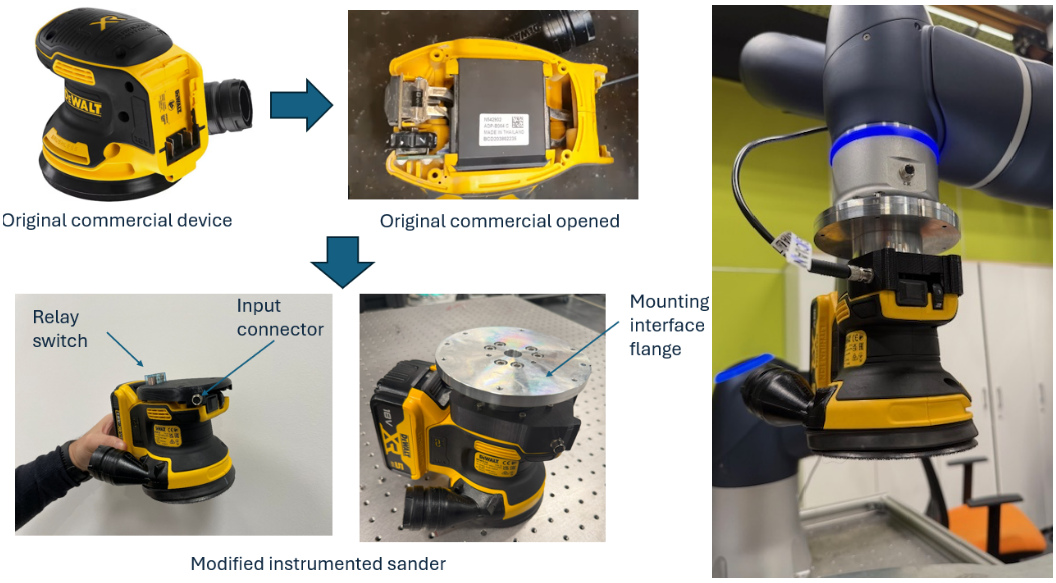

Original commercial device
Original commercial opened
Relay switch
Input connector
Mounting interface flange
Modified instrumented sander

Figure 12: The customized grinding tool integrating the mounting interface, the relay ON/OFF switch and its communication input connector permitting to control the device from the control unit of the robotic system.

2.6.2 DAMPING MODULE DESIGN

The operation of the grinding tool into the robotic system can produce disturbances to the robotic system due to the intrinsic vibrations generated by the orbital grinding tool. A foundational component of the grinding robotic system was therefore to the interface damping module, which can minimise the effect of these vibrations and improve interaction precision through integrated force sensing and damping mechanisms.

2.6.2.1 GRINDING TOOLS VIBRATIONS STUDY

To achieve this, an important phase of the development of such a module was to understand the nature of the vibrations generated by the grinding tools. For this purpose, we conducted an experimental testing phase using a force/torque (FT) sensor, specifically the FT Mini-45 model, attached to the grinding device. The FT sensor was used to measure the interaction forces during the use of a grinder tool, providing data critical to understanding and mitigating the effects of tool-induced vibrations (Figure 13).

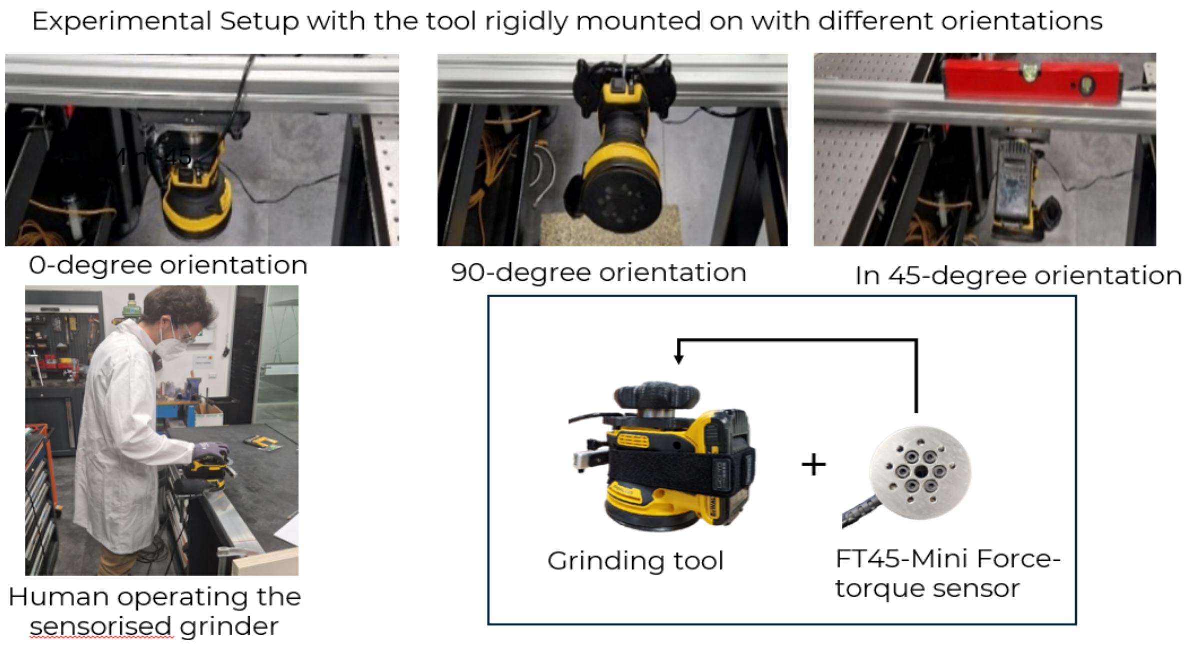

Experimental Setup with the tool rigidly mounted on with different orientations
0-degree orientation
90-degree orientation
In 45-degree orientation
Human operating the sensorised grinder
+
Grinding tool
FT45-Mini Force-torque sensor

Figure 13: The sensorised grinding tool used for the collection of vibration measurements.

The experiments investigated various tool orientations, notably 0 degrees, 45 degrees, and 90 degrees relative to a fixed reference frame. These configurations were selected to analyse how the contact angle influences the magnitude and direction of transmitted forces. Vibration analysis revealed a significant frequency range between 150 and 190 Hz in the 0-degree configuration, highlighting a critical band that must be accounted for in both the control loop and mechanical design (Figure 14).

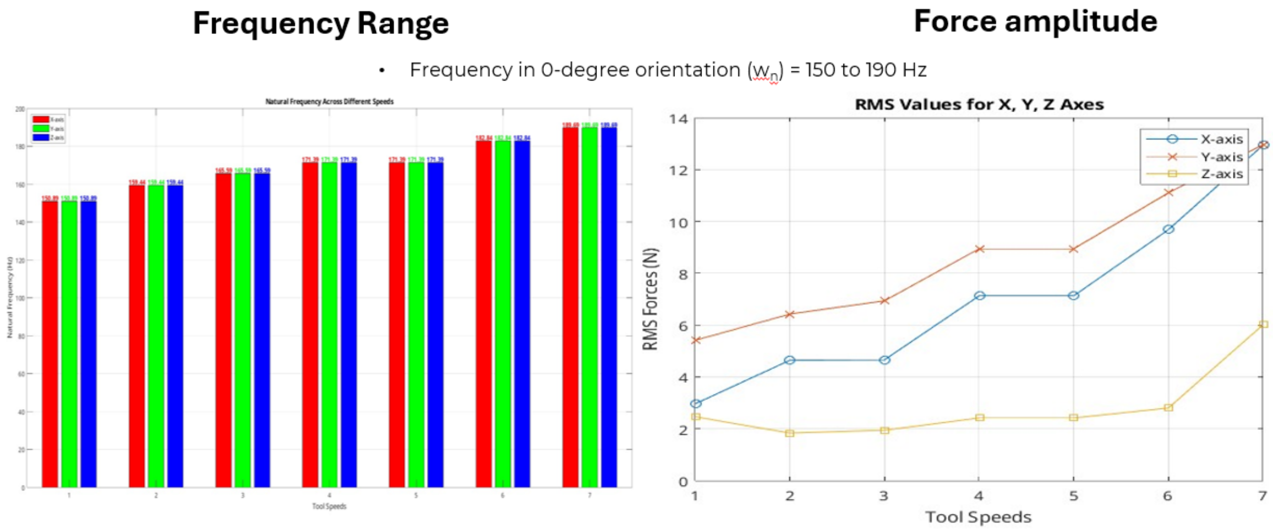
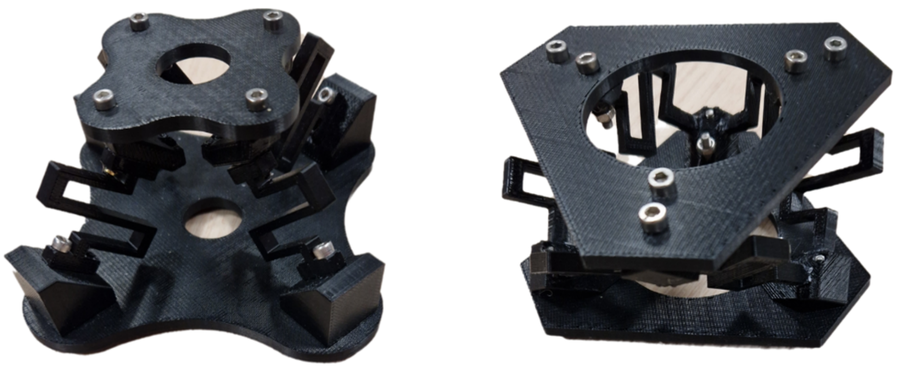


Figure 14: The frequency range and amplitude of the grinding tool induced vibrations exhibiting a frequency in the range of 150-190Hz for the different tool speed levels.

2.6.2.2 MECHANICAL ISOLATION STRUCTURE

Using the data collected from the study of the grinding tool vibrations, we developed innovative mechanical isolators that will form an integral part of the robotic end-effector. The first isolator design, referred to as the Distributed Integrated Folded Beam Isolator, was engineered to dampen vibrations primarily in the three translational axes. This design is characterised by its structural simplicity and potential for rapid iterative refinement (Figure 15). While effective in reducing translational vibration, the isolator exhibits limitations in filtering rotational disturbances, particularly the moments generated during contact with angled surfaces.



Distributed Integrated Folded Beam Isolator Platform

The novel 3–3 Stewart–Gough isolator platform

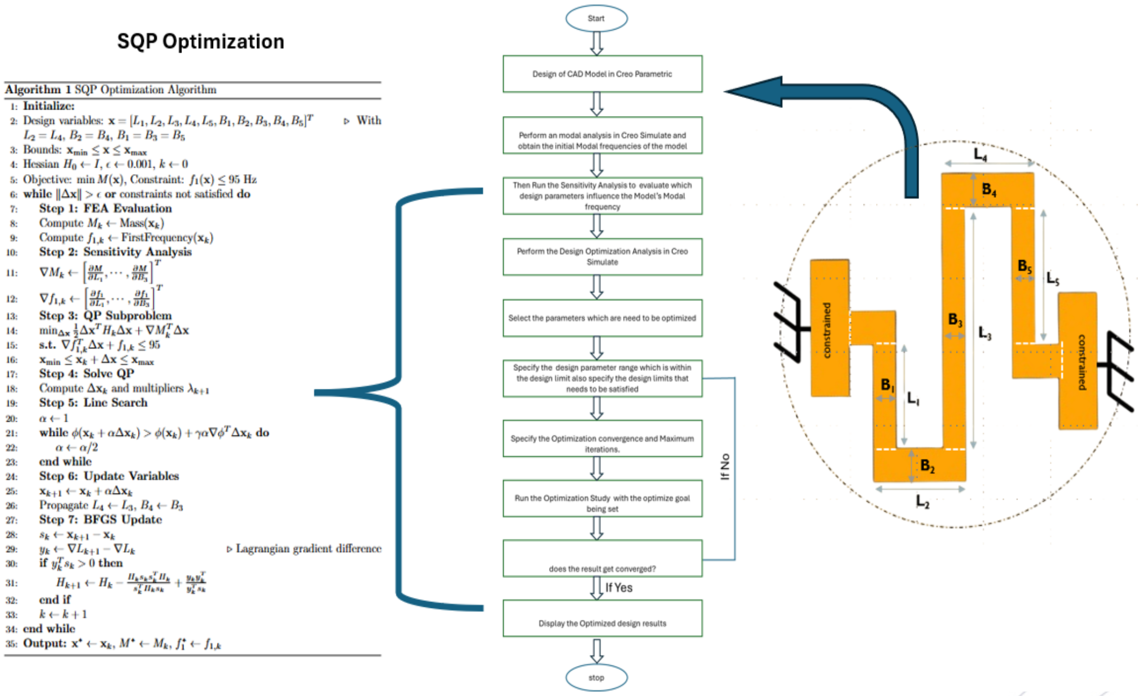
Figure 15: The 3D printed prototypes of the two vibration isolators studied.

Considering the need for a more comprehensive solution, we also introduced a novel design that combines the principles of a Stewart–Gough platform with folded beam isolation technology. The resulting hybrid platform is capable of damping vibrations in all six degrees of freedom, thereby offering significantly improved performance in environments where multi-directional forces and torques are present. The folded beam elements are integrated into the structure of a rigid Stewart–Gough platform, enabling it to retain stiffness while simultaneously providing flexibility for damping.

The mechanical behaviour of these structures was examined through analytical and simulation-based approaches. Stress levels in the folded beams were predicted using kinematic models combined with the Euler–Bernoulli bending theory. These calculations permitted to evaluate how the beams deform under different load conditions and to assess the overall effectiveness of the vibration isolator module.

2.6.2.3 ISOLATOR OPTIMISATION

To ensure that the isolator met stringent performance and weight requirements, we carried out a numerical optimization process based on Sequential Quadratic Programming (SQP). This approach was used to optimize the geometry and material distribution of the CAD model representing the isolator (Figure 16). The optimization aimed to achieve a balance between stiffness, damping effectiveness, and structural weight, all of which are essential factors in determining the suitability of an end-effector for dynamic tasks.



SQP Optimization

```
Algorithm 1 SQP Optimization Algorithm
Initialize:
Design variables: $\mathbf{x} = [L_1, L_2, L_3, L_4, L_5, B_1, B_2, B_3, B_4, B_5]^T$    ▷ With $L_2 = L_4$, $B_2 = B_4$, $B_1 = B_3 = B_5$
Bounds: $\mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}$
Hessian $H_0 \leftarrow I$, $\epsilon \leftarrow 0.001$, $k \leftarrow 0$
Objective: $\min M(\mathbf{x})$, Constraint: $f_1(\mathbf{x}) \leq 95$ Hz
while $\|\Delta\mathbf{x}\| > \epsilon$ or constraints not satisfied do
    Step 1: FEA Evaluation
    Compute $M_k \leftarrow \text{Mass}(\mathbf{x}_k)$
    Compute $f_{1,k} \leftarrow \text{FirstFrequency}(\mathbf{x}_k)$
    Step 2: Sensitivity Analysis
    $\nabla M_k \leftarrow \left[\frac{\partial M}{\partial L_1}, \cdots, \frac{\partial M}{\partial B_5}\right]^T$
    $\nabla f_{1,k} \leftarrow \left[\frac{\partial f_1}{\partial L_1}, \cdots, \frac{\partial f_1}{\partial B_5}\right]^T$
    Step 3: QP Subproblem
    $\min_{\Delta\mathbf{x}} \frac{1}{2}\Delta\mathbf{x}^T H_k \Delta\mathbf{x} + \nabla M_k^T \Delta\mathbf{x}$
    s.t. $\nabla f_{1,k}^T \Delta\mathbf{x} + f_{1,k} \leq 95$
    $\mathbf{x}_{\min} \leq \mathbf{x}_k + \Delta\mathbf{x} \leq \mathbf{x}_{\max}$
    Step 4: Solve QP
    Compute $\Delta\mathbf{x}_k$ and multipliers $\lambda_{k+1}$
    Step 5: Line Search
    $\alpha \leftarrow 1$
    while $\phi(\mathbf{x}_k + \alpha\Delta\mathbf{x}_k) > \phi(\mathbf{x}_k) + \gamma\alpha\nabla\phi^T\Delta\mathbf{x}_k$ do
        $\alpha \leftarrow \alpha/2$
    end while
    Step 6: Update Variables
    $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha\Delta\mathbf{x}_k$
    Propagate $L_4 \leftarrow L_3$, $B_4 \leftarrow B_3$
    Step 7: BFGS Update
    $s_k \leftarrow \mathbf{x}_{k+1} - \mathbf{x}_k$
    $y_k \leftarrow \nabla L_{k+1} - \nabla L_k$    ▷ Lagrangian gradient difference
    if $y_k^T s_k > 0$ then
        $H_{k+1} \leftarrow H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$
    end if
    $k \leftarrow k + 1$
end while
Output: $\mathbf{x}^* \leftarrow \mathbf{x}_k$, $M^* \leftarrow M_k$, $f_1^* \leftarrow f_{1,k}$
```

Start
Design of CAD Model in Creo Parametric
Perform an modal analysis in Creo Simulate and obtain the initial Modal frequencies of the model
Then Run the Sensitivity Analysis to evaluate which design parameters influence the Model's Modal frequency
Perform the Design Optimization Analysis in Creo Simulate
Select the parameters which are need to be optimized
Specify the design parameter range which is within the design limit also specify the design limits that needs to be satisfied
Specify the Optimization convergence and Maximum iterations.
Run the Optimization Study with the optimize goal being set
does the result get converged?
If No
If Yes
Display the Optimized design results
stop
L4
B4
B5
L5
B3
L3
constrained
B1
L1
B2
L2
constrained

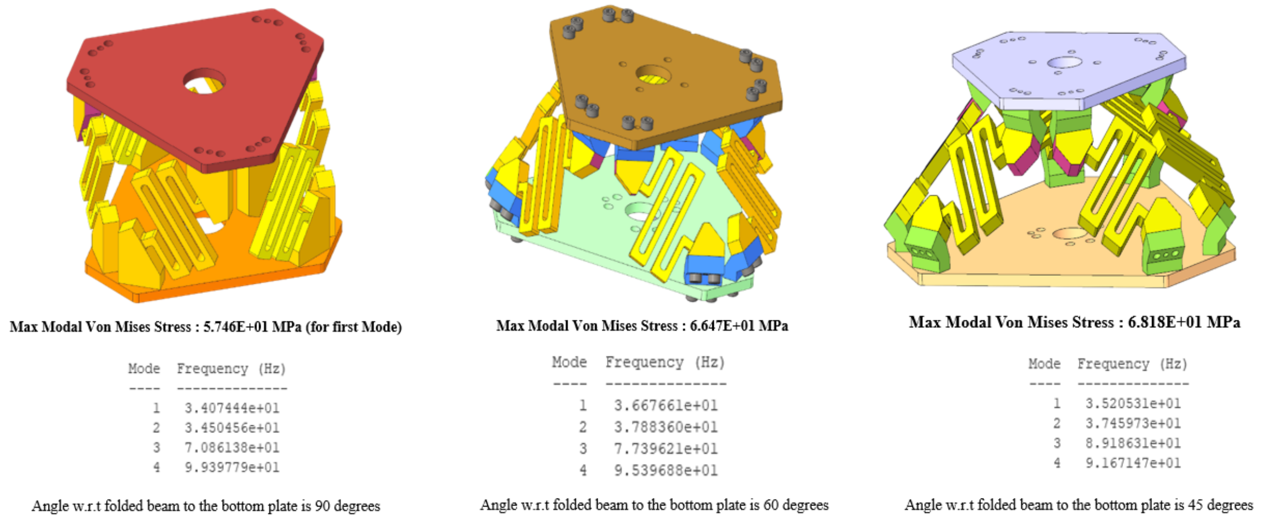
Figure 16: The optimization process employed for tuning the mechanical design of the vibration isolators.

Beyond static optimization, a detailed modal analysis was conducted to understand the dynamic behaviour of the isolators under different configurations.

This analysis included both the conventional folded beam isolator and the newly proposed Stewart–Gough hybrid platform. Modal analysis is a critical step in characterising the vibrational modes of a mechanical structure, as it reveals how the system will naturally respond to external excitations at various frequencies. These findings guided the design to avoid resonant conditions that could compromise performance.

Furthermore, the influence of the beam angle relative to the horizontal base plate of the Stewart–Gough platform. Three configurations were analysed, where the angle between the folded beam and the bottom plate was set to 45 degrees, 60 degrees, and 90 degrees, respectively. This parametric study was aimed at understanding how changes in geometry affect both the modal (vibrational) and static (structural) performance of the isolator.

The results of this study revealed that these geometric variations significantly influenced stress distribution and natural frequencies (Figure 17). For instance, a 90-degree configuration yielded a modal Von Mises stress of approximately 57.5 MPa, while the 60-degree and 45-degree configurations resulted in higher stresses of 66.5 MPa and 68.1 MPa, respectively. These findings illustrate that increasing the angle reduces compliance but increases internal stresses, thereby affecting the system's ability to absorb vibration without undergoing material fatigue or failure.



Max Modal Von Mises Stress : 5.746E+01 MPa (for first Mode)

```
Mode  Frequency (Hz)
----  --------------
   1   3.407444e+01
   2   3.450456e+01
   3   7.086138e+01
   4   9.939779e+01
```

Angle w.r.t folded beam to the bottom plate is 90 degrees

Max Modal Von Mises Stress : 6.647E+01 MPa

```
Mode  Frequency (Hz)
----  --------------
   1   3.667661e+01
   2   3.788360e+01
   3   7.739621e+01
   4   9.539688e+01
```

Angle w.r.t folded beam to the bottom plate is 60 degrees

Max Modal Von Mises Stress : 6.818E+01 MPa

```
Mode  Frequency (Hz)
----  --------------
   1   3.520531e+01
   2   3.745973e+01
   3   8.918631e+01
   4   9.167147e+01
```

Angle w.r.t folded beam to the bottom plate is 45 degrees

Figure 17: The results from the study influence of the beam angle relative to the horizontal base plate of the Stewart–Gough platform.

2.6.2.4 ISOLATOR DESIGN CONCLUSION FOR INTEGRATION

The final phase of the study focused on interpreting the modal analysis results to guide design decisions for integration into the robotic system. The 60-degree beam configuration emerged as a favourable compromise, providing sufficient damping capacity while maintaining manageable stress levels.

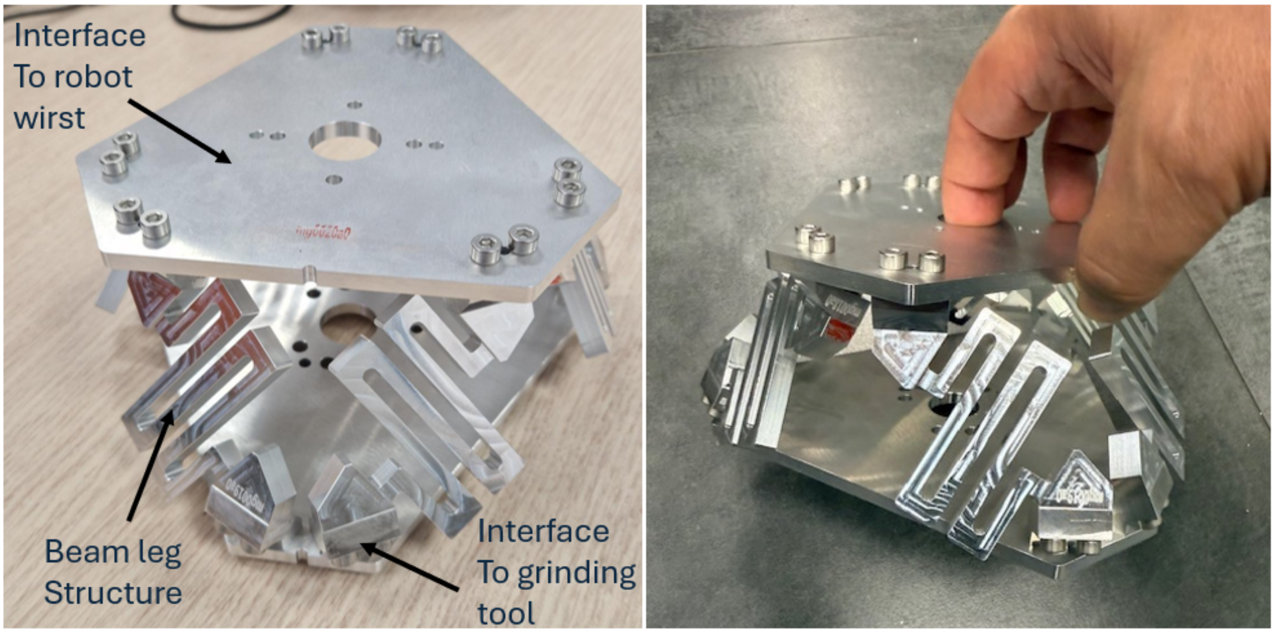


Figure 18: The realized new 3–3 Stewart–Gough vibration isolator module.

The analysis also confirmed that the novel Stewart–Gough platform, enhanced by the folded beam architecture, could serve as an effective solution for robotic applications requiring both precision and robustness in high-vibration tasks such as grinding. This synergy between mechanical isolation design and active control architecture that leverages on force/torque sensing combined with real-time feedback from the impedance control framework will allow the robot to dynamically adapt to disturbances will reduce the effect of the grinding vibrations on the robot system hardware and its control. A prototype of the new 6DOF vibration isolator is shown below.

The realized prototype of the vibration isolation module (Figure 18) will undergo a validation study considering the same setup used for the study of the grinding tool operations to evaluate it performance in attenuating the generated vibrations. As the project progresses, the design principles and insights from the vibration isolator developments and its use/validation will guide any subsequent tuning iterations of the isolator design.

3 MOTION AND INTERACTION CONTROL

3.1 SUMMARY OF THE SOFTWARE INFRASTRUCTURE DESIGN

Modern robot control systems are distributed. In the case of the MAGICIAN project (and of many others), the architecture is comprised of a low-latency, hardware-proximate component and a remote, often cloud-based, portion which is used to run the modules for which a strict real--time behaviour is not required. Our design is based on **XBot2**, a **real-time middleware** controlling the hardware-proximate layer. XBot2, distinct from fixed firmware, is a customisable, plugin-based framework.

Designed for the MAGICIAN project, XBot2 provides seamless integration points for both real-time and non-real-time components, aiming for **mixed hardware topology support**, a **component-based design**, and **high modularity** for reusability. It ensures real-time components access the same user-friendly APIs as their non-real-time counterparts, offering a real-time toolbox with debugging tools.

Deterministic real-time behaviour in Linux is achieved via **PREEMPT_RT** (pre-emptive kernel patching) or **dual-kernel methods** like **Xenomai**. The advantage of single-kernel approaches is that they use standard UNIX/POSIX primitives, whilst dual-kernel mandate the use of a specialised API to execute real--time activities.

XBot2 introduces several significant advancements over its predecessor XbotCore:

- A **fully dynamic hardware abstraction layer (HAL)** that supports on-the-fly device auto-discovery and generates high-level APIs, simplifying integration with user-defined behaviours.
- An **enhanced, flexible multi-threaded plugin system** for managing periodic operations.
- **Operating system service abstractions** that form architectural building blocks, greatly improving portability across various **Real-Time Operating Systems (RTOSs)**. Unlike XBotCore, which was limited to Xenomai, XBot2 is now agnostic to single or dual-kernel approaches.
- **User-land facilities** enabling fully decoupled internal component communication through lock-free synchronous and asynchronous paradigms. This allows components (at both HAL and behaviour levels) to be seamlessly relocated across different execution threads.
- A more robust and flexible **dual-process, client-server architecture** for communicating with robot hardware or simulators, a notable improvement over the previous single-process design. This simplifies support for multiple simulators (e.g., Gazebo, PyBullet, MuJoCo) and robotic systems, while enhancing fault tolerance and process separation.
- **Full integration with the ROS2 environment**, facilitating communication with non-real-time software components.

3.1.1 XBOT2 DESIGN HIGHLIGHTS

Figure 19 illustrates the MAGICIAN control framework. At the lowest level, there's a dedicated layer for controlling both the **Doosan H2515 cobot** and the **MAGICIAN End-Effector**. We'll use the **Real-Time DRFL API** (Doosan Robotics Framework Library) for the cobot and a custom implementation of the **SOEM library** (Simple Open EtherCAT Master Library) for the end-effector.

By leveraging the **XBot2 software architecture**, we can transparently control the entire MAGICIAN robot system in both simulation and on real hardware. The main aspects of the middleware are detailed next.

- The **Hardware Abstraction Layer (HAL)** in XBot2 serves two primary purposes. Firstly, it provides controlled access to external devices, meeting specific performance needs. Where possible, it also **auto-discovers connected devices** to minimise user configuration effort. In XBot2, "hardware" is broadly defined, encompassing direct fieldbus access (e.g., EtherCAT), network interfaces to robot control boxes (common with commercial research robots), or any other indirect connection to hardware or simulators.
  Secondly, the HAL presents users with a programming interface that **abstracts away unnecessary details**, promoting software component reuse even if compatible hardware changes. This means users don't need to alter their code when switching from simulation to experiments, changing device vendors, or even entirely new robot vendors. The HAL system must also meet the general requirements, allowing both the HAL itself and its clients to be **transparently relocatable to any execution thread**. To achieve this, XBot2's HAL employs a **client-server approach**, with two distinct components:

  - **DeviceDriver**: A unique, per-device component that connects and communicates directly with the hardware, acting as the server side of a device's HAL.
  - **DeviceClient**: The component responsible for providing the user-facing API. Typically, multiple instances of the client side exist within the system.

  While XBot2 offers developers maximum flexibility in connecting and synchronising driver and client components, it also provides ready-to-use base classes that rely on a set of primitives. This ensures the satisfaction of generic requisites in a standardised manner. The implemented class hierarchy and its relevant UML diagram are shown in Figure 20. From a user's perspective, device drivers and clients are automatically loaded by XBot2's HalManager during framework initialisation, based on a configuration file. Available device clients can then be inspected and used from a control module via the RobotInterface object, as detailed in the next section.

Finally, device drivers and clients are loaded into containers. These containers group together all devices of the same type (e.g., all joints), allowing for operations that involve all such devices simultaneously, depending on the hardware interface characteristics.

- **Behaviours:** XBot2's ultimate aim is to facilitate the **simple and dynamic use of devices** by control or monitoring modules to achieve desired behaviours. This functionality is implemented within a component called a **ControlPlugin**, which can be loaded into the system dynamically at runtime. The XBot2 ControlPlugin comprises three main elements:
    - A **finite state machine** that outlines the module's lifecycle.
    - A set of **facilities for named resource resolution**.
    - A **RobotInterface object** that provides access to the HAL's client side.

The **ControlPlugin lifecycle** defines the component's states and allowed transitions, managed via callbacks (Figure 21). For example, periodic tasks use a callback assigned to the 'Run' state, executed with each clock tick. The **TaskManager** oversees plugin execution, broadcasts module status and statistics (like CPU time), and provides a service for other system components to trigger state changes (e.g., to start a plugin).

XBot2's RobotInterface, an evolution of XBotCore's design, is the primary access point to the HAL's client side. Part of the broader **XBotInterface package**, it includes an XBotInterface base class, ModelInterface, and RobotInterface. The XBotInterface leverages URDF and SRDF to present robot state data (e.g., motor positions, velocities, torques) in a more manageable, chain-by-chain format.

The **ModelInterface** inherits this chain-wise structure, adding methods to retrieve kinematic and dynamic quantities from the robot state, with a concrete implementation based on the RBDL library.

The **RobotInterface** connects the inherited robot state to a physical robot via virtual sense/mode methods. Two concrete implementations are provided: **RobotInterfaceRos**, a ROS-based client library connecting to XBot2's ROS API, and **RobotInterfaceRt**, which uses the HAL client side. This abstraction allows control code from an XBot2 **ControlPlugin** to be moved to a remote ROS node with minimal effort, though real-time guarantees are then lost.

Each **ControlPlugin** loaded generates a **RobotInterface** instance, which dynamically loads all defined HAL device client sides. Users can then inspect the HAL system using the getDevices method to retrieve device clients conforming to a specific virtual interface. This enables writing generic control modules that use the most appropriate interface for a task, with optional, more specific code executing only if the target hardware supports it.

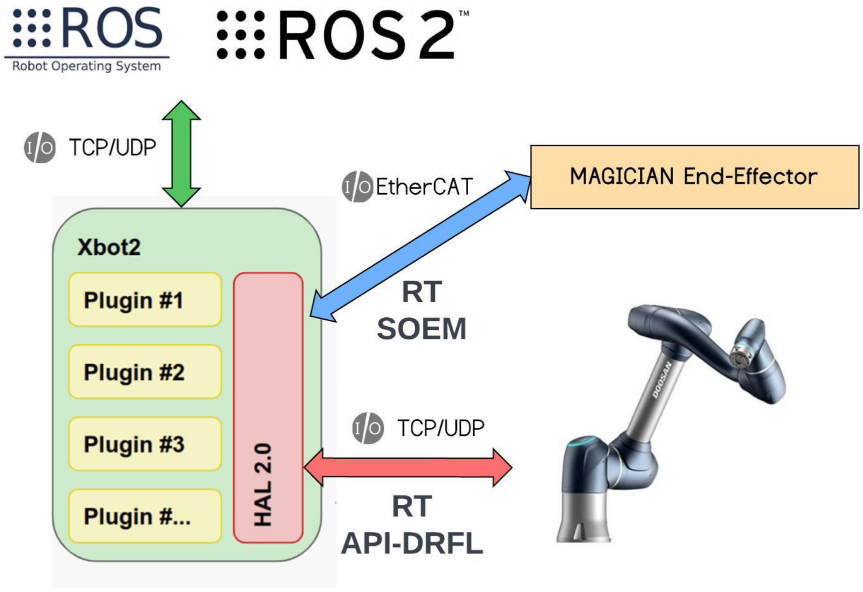


Figure 19: Overview of the MAGICIAN framework.

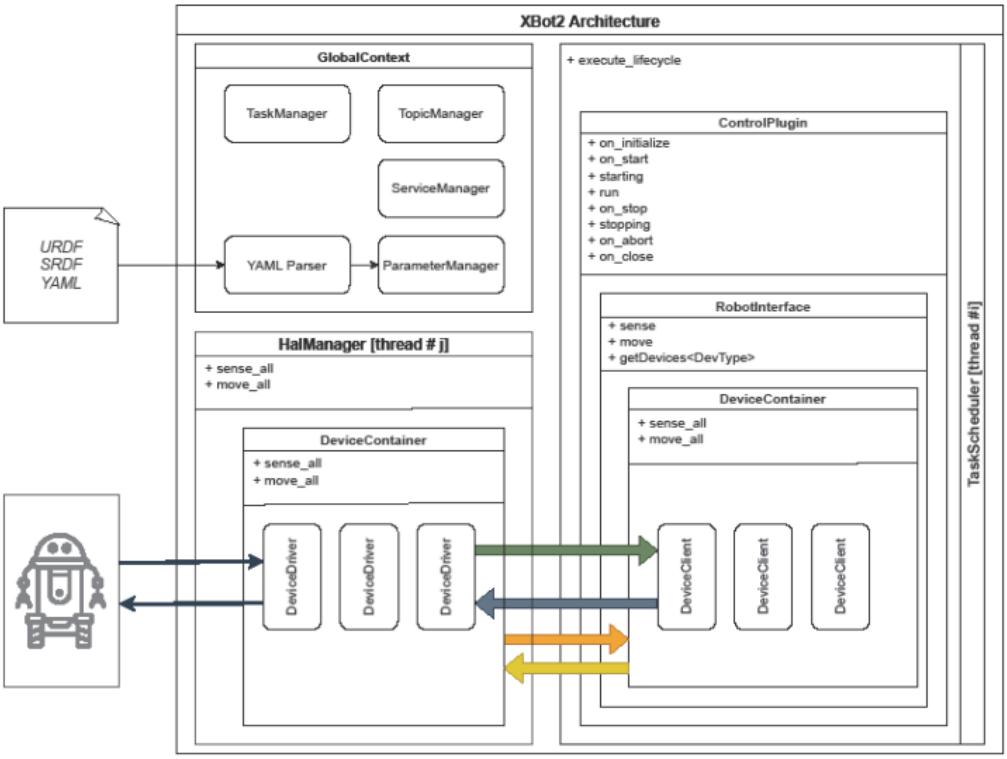


Figure 20: Xbot2 implemented class hierarchy and related UML diagram.

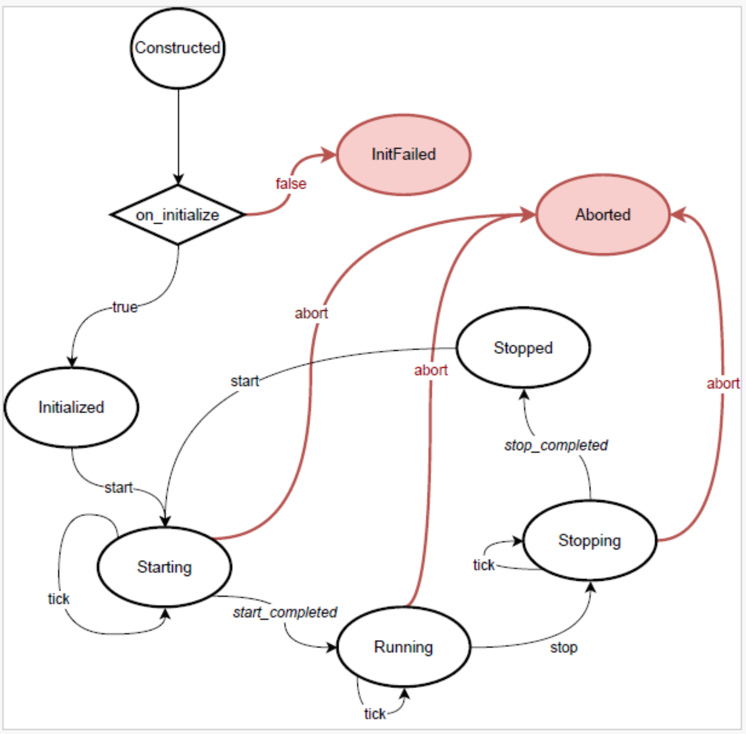


Figure 21: States and permitted transitions of the Control Plugin lifecycle.

3.2 DESIGN OF MOTION/IMPEDANCE CONTROL

One crucial module enabled by the Xbot-2 middleware is the **motion generation system based on impedance control**, a scheme widely recognised in literature since Hogan's work in 1985 [Hogan1985-PartI]. As modern robots, like those in this project, must operate in dynamic and changing environments, accounting for **manipulator interaction forces** is fundamental. These forces are not only unavoidable when handling objects or interacting with the environment, but they are also vital for ensuring **safety during unexpected collisions**. This is critical for protecting both the robot and, more importantly, humans working near or collaborating with it. These considerations are central to our project's mission of developing human-centred technologies.

Impedance control can be implemented at different levels: **task/operational space** and **joint space**. The joint space implementation is based on the following formula:

$$\tau = K(q_d - q) + D(\dot{q}_d - \dot{q}) + M(q)(\ddot{q}_d) + C(q, \dot{q})\dot{q} + g(q) + J^T(q)f$$

Here, Ks represents the stiffness gain matrix, K_d the damping gain matrix, M the mass matrix, C the Coriolis torques, g the gravity torques, J^T the transposed Jacobian matrix, and *f* any other external forces, such as those from interaction with objects or people. By adjusting these impedance parameters, varying levels of **compliance** can be achieved as required. This control module is implemented using the XBot middleware and can be deployed in real-time control plugins.

A key application of our impedance control module is the **gravity-compensated**

teaching/interaction mode. By appropriately tuning the gains in the impedance formula, the robot can become fully compliant to external forces (e.g., those applied by a human), and yet still compensate for gravity. This means it remains stationary when no external forces are present. This mode allows for **safe and effortless teaching of robot poses or trajectories** by intuitively moving the manipulator to desired positions, as detailed in [Muratore2023].

3.2.1 GRINDING SYSTEM MOTION/IMPEDANCE CONTROL

Robotic systems, designed to operate in unstructured or dynamic environments, as is the case in MAGICIAN, must be capable of managing physical interactions in a safe and adaptive manner. These interactions include deliberate contacts, such as those during object manipulation or surface grinding, as well as unexpected contacts, such as collisions with unknown objects or humans (Figure 22).

As part of the control architecture developed for the grinding robot within the MAGICIAN project, a flexible motion/impedance control module has been implemented for the grinding robot using the XBot2 middleware. This middleware-based approach allows for modular and real-time control of the grinding robot, enabling it to adapt its behavior across various tasks and operating environments.

A key feature of this architecture is a motion generation module based on impedance control, a paradigm that has been extensively validated in robotics, starting with the foundational work by Hogan [Hogan1985-PartII]. Impedance control provides a robust framework for regulating the dynamic interaction between a robot and its environment by shaping the mechanical impedance, i.e., the relationship between motion and force of the manipulator.

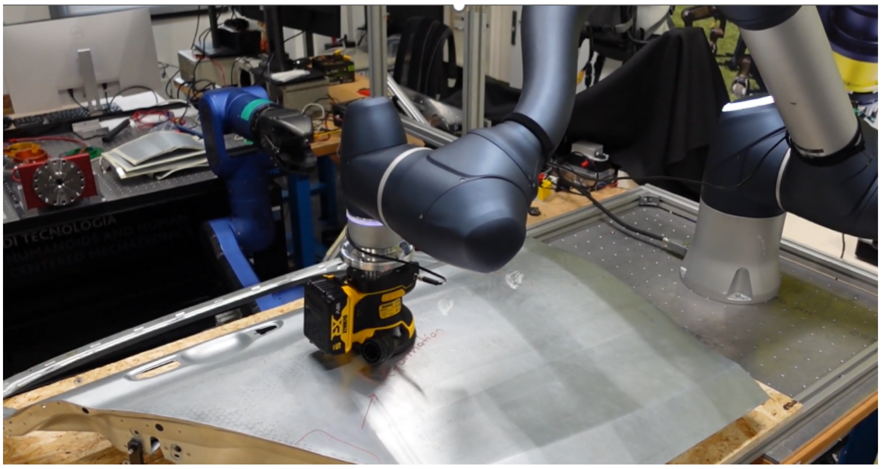


Figure 22: Snapshot from first grinding experiments showing the griding tool in contact with a car body part while its motion and contact are regulated by the impedance control module.

This approach directly supports the project's goal of developing human-centered robotic technologies, ensuring that the robot behaves in a safer manner during close robot collaboration. Impedance control also enables the robot to respond to external disturbances in a compliant way, reducing the risk of injury or damage in the event of accidental contact. The Impedance control can be implemented at different hierarchical levels, depending on the application requirements:

- Operational/task space impedance control allows the system to regulate interaction dynamics in Cartesian space, aligning directly with task-related forces and motions.
- Joint space impedance control, on the other hand, regulates the dynamics at the actuator level. This is formulated as follows:

$$\tau = K(q_d - q) + D\,(\dot{q_d} - \dot{q}) + M(q)(\ddot{q}_d) + C(q, \dot{q})\dot{q} + g(q) + J^T(q)f$$

where K is a matrix of stiffness gains, D is a matrix of damping gains, $M(q)$ is the mass matrix, $C(q, \dot{q})\dot{q}$ are Coriolis torques, $g(q)$ are gravity torques, $J^T(q)$ is the transposed Jacobian matrix, and f any kind of other external forces due, for example, to interaction with objects or people. By modulating the impedance parameters, it is possible to obtain different levels of compliance, according to the needs. As mentioned earlier, this control module is implemented using the XBot2 middleware and can be leveraged in real-time control plugins when needed.

By tuning the impedance parameters, stiffness K and damping D, the compliance and damping behavior of the robot can be finely modulated. This allows the system to achieve behaviors ranging from rigid position tracking to highly compliant interaction, depending on task demands. The impedance control module is fully integrated into the XBot2 framework and is available for use in real-time control plugins, ensuring low-latency, synchronous response, and easy integration with other control layers and safety systems.

4 PLANNING AND SCHEDULING

4.1 SUMMARY OF SCHEDULING ALGORITHM DESIGN

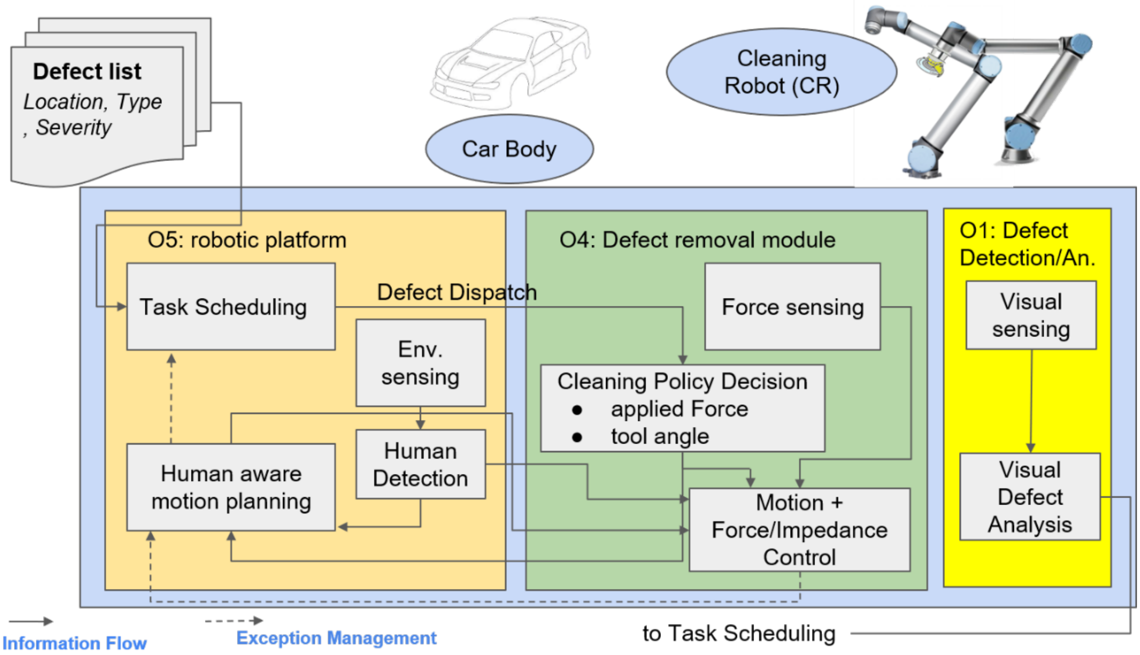


Figure 23: The conceptual architecture of the cleaning robot.

Figure 23 illustrates the **Cleaning Robot's** conceptual architecture. Its **task scheduling component (TS)** receives severity, type, and location-classified defect lists. The TS sequences defect treatment based on the required cleaning policy, defect severity, and time constraints. A **human-aware motion planner** then determines optimal trajectories between tasks, adjusting plans to minimise collision risk by predicting human motion detected by an environment monitoring system (techniques from D4.1).

Conversely, Figure 24 shows the **Sensing Robot's** architecture. Here, the task scheduling component generates a trajectory to visit and identify defects. It leverages statistical data from past executions to prioritise areas where defects are likely, often stemming from systematic problems. However, the scheduler must balance revisiting known defect areas with exploring new ones to account for changes over time.

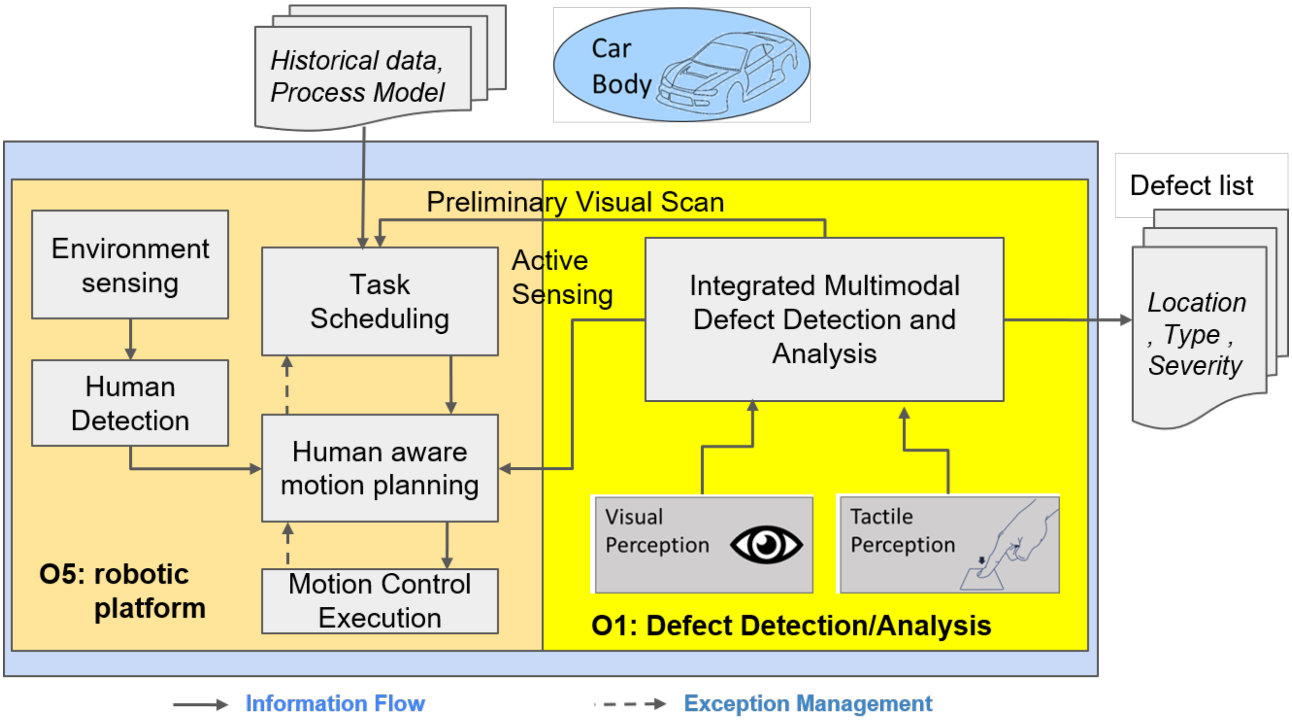


Figure 24: The logical architecture of the sensing robot.

The workings of the human-aware motion planner are essentially the same as for the CR.

4.2 DESIGN OF PLANNING AND SCHEDULING FOR THE CR:

As shown in Figure 23, the **Task Scheduling component** receives input detailing potential defect locations on the car body. Each location is associated with several key parameters:

- **Robot Workspace Position:** A standard position for the robot to begin cleaning, such as the end-effector being 10cm from the surface and orthogonal to it.
- **Reward (Ri):** Linked to the defect's potential removal, this combines the defect's severity and the importance of its position.
- **Probability (Pi):** The likelihood of the defect being present, reflecting the **Surface Repair (SR) robot's** confidence in detection.
- **Processing Time (Ti):** The estimated time needed for defect removal, based on its severity and location within the robot's workspace.

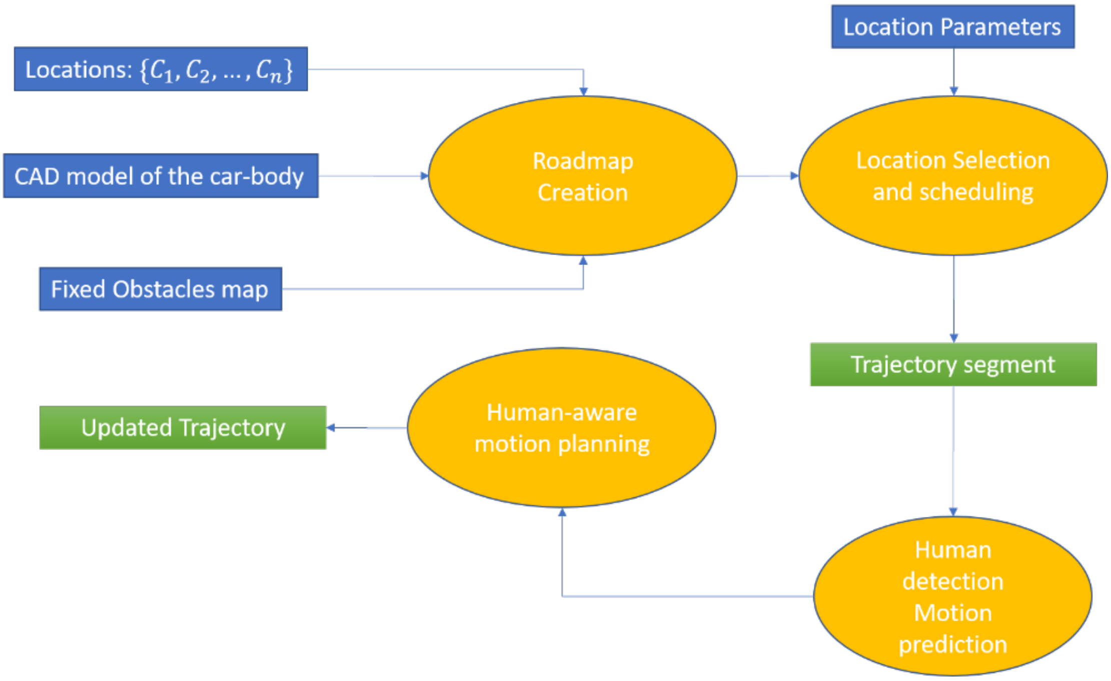

Locations: $\{C_1, C_2, \dots, C_n\}$
CAD model of the car-body
Fixed Obstacles map
Roadmap Creation
Location Parameters
Location Selection and scheduling
Trajectory segment
Human-aware motion planning
Updated Trajectory
Human detection Motion prediction

Figure 25: Planning pipeline for MAGICIAN "classic approach".

Crucially, there's a **maximum time (Tmax)** for completing the exploration, determined by the "Takt time." At present, we've established a standard decomposition between motion and task planning, as illustrated in Figure 25.

4.2.1 ROADMAP CREATION

The first step is finding a roadmap, i.e., a graph-based representation joining the different locations. This step can be performed using standard planning algorithm to connect the points pairwise [Lavalle2006]. The construction of a roadmap amounts to the creation of a graph in which:

- The nodes correspond to the locations $C = \{C_0, \dots, C_{n-1}\}$
- The arcs correspond to the pairwise motion between two nodes, with each arc having a travel cost τ_{ij}.

Once the robot arm has reached the operation area associated with each location, the robot starts executing dynamic motion primitives [Saveriano2023] to remove the defect. Such primitives are the result of a dynamic equation of the form:

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f(x),$$

$$\tau \dot{y} = z,$$

$$\tau \dot{x} = -\alpha_x x$$

where y is the desired position (e.g., of the end-effector), z is associated with velocity $f(x)$ is a forcing term and x is an auxiliary variable, which vanishes with exponential rate and is used to modulate in time the forcing action. This aspect is analysed in depth in D3.1.

4.2.2 LOCATION SELECTION AND SCHEDULING

The location selection and scheduling problem is a typical problem of graph optimisation. One of the best-known problems of this kind is the so-called **Traveling Salesman Problem (TSP)**, in which an agent is required to visit a set of nodes.

In our case, the problem takes a different form because of the maximum time budget constraint. This modified problem is known as the **Orienteering Problem (OP)** and is defined as follows. Consider a set of nodes connected by a graph, each one associated with a reward that is earned if the node is visited. Suppose an agent starts its travel at a first node (1) and finishes at a final destination node (n). The problem amounts to finding a selection and a sequence of nodes so that the total travel time remains below T_m (1) and the total reward is maximised (2).

The problem can be modelled as an Integer Linear Programming (ILP) problem or solved using heuristic solutions. When the number of nodes is reasonably small, like in our case, the solution time is very low even for a full ILP formulation [Archetti2007]. A complete mathematical formulation of the orienteering problem is in D4.1.

The input parameters required in our case are Time Constraint (T_{Max})

- Travel Times Matrix (T_{xy})
- Estimated Cleaning Times (t_i)
- Profit for Cleaning Defect (ϕ_i)

While the time constraint is a constant, the other inputs must be derived by three different modules, as visualized in Figure 26.

To create the travel times matrix (T_{xy}), as visualized in Figure 26 as the "Distance Matrix Calculator", we need to follow these detailed steps: 1. **Collect Defect Locations**, 2. **Calculate Euclidean Distance.** The details are in D4.1

To determine the profit ϕ_i for cleaning each defect, follow these steps:

1. **Identify Defect Types and Severities**: From the sensing robot's output, obtain the type and severity of each defect.

2. **Define Priority Formula:** Develop a formula to calculate the priority of cleaning each defect based on its type and severity. For example, a simple weighted sum between type and severity.

Lastly, to estimate the cleaning times (t_i) for each defect for the "Cleaning Times Calculator" in Figure 26, use the following procedure:

1. **Obtain Cleaning Time Data**: Based on historical data or expert input, determine the average time required to fix defects of given type severity.

2. **Calculate Individual Cleaning Times**: Using the defect type and severity data, lookup or calculate the estimated cleaning time for each defect (e.g., using a lookup table).

3. **Format the Cleaning Times**: Compile the estimated cleaning times into a list

$[t_1, t_2, \dots, t_n]$, ensuring that all times are in a consistent unit (e.g., seconds or minutes).

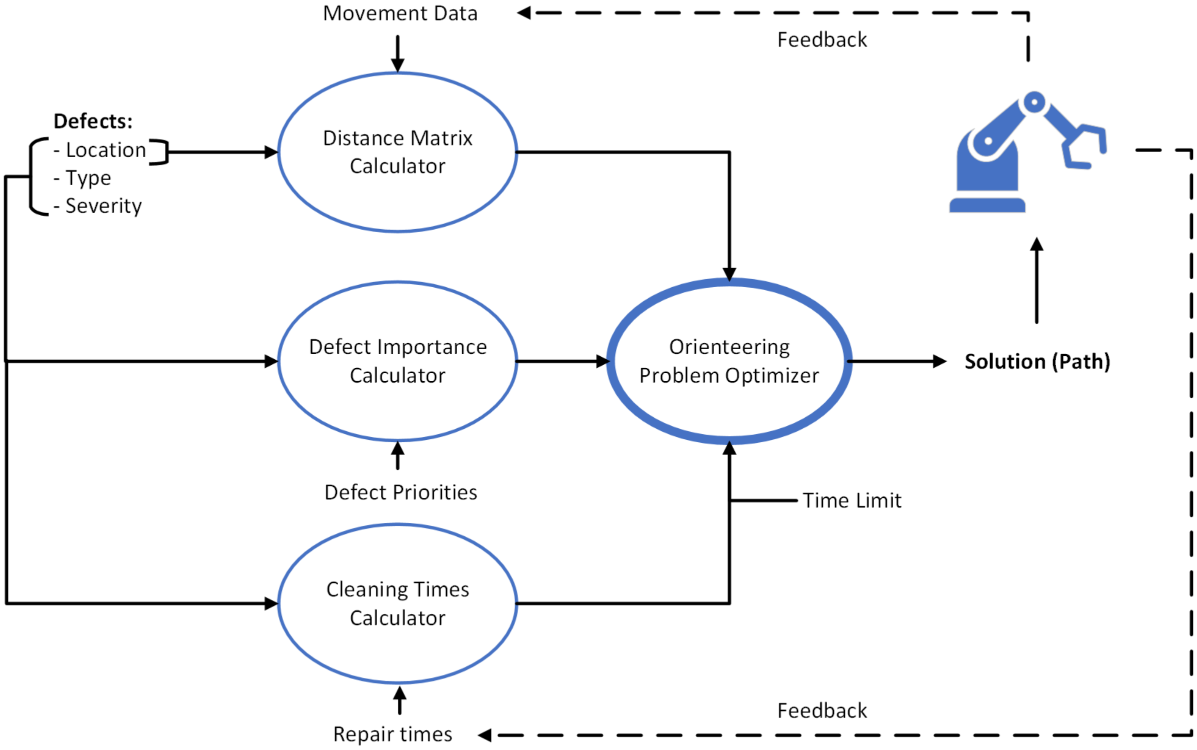


Figure 26: Path Optimizer Framework.

4.3 CR TASK PLANNER IMPLEMENTATION

In the development of the OP solver, several design choices have been made, each of which plays a crucial role in the functionality of the module. Some of these choices serve as placeholders and may be adapted to suit alternative use cases. In such instances, these placeholders will be explicitly identified.

Profit Function

The OP module necessitates a profit function to effectively optimize the path. Without this, it is impossible to quantify the quality of a path. Several profit functions have been evaluated, including:

1. **Exponential Profit Ratio Reward Function**: This function combines a profit-collecting coefficient with a profit ratio function to simulate profit accumulation. It is represented as $f_i(s_i) = 1 - e^{-\beta_i s_i}$, where $f_i(s_i)$ denotes the profit earned for defect i given that s_i time is spent at that node. The parameter β allows for the manipulation of the decline in marginal profit. Notably, as the time spent at a node increases, the total profit increases while marginal profit diminishes (Yu et. al., 2019).

2. **Quadratic Profit Function**: This function initially allows for significant reward accumulation when addressing a defect, but profits peak and then decline. It is formulated as $f_i(s_i, d_i) = -\frac{200d_i}{h^2}(s_i - h)^2 + 200d_i$, where $f_i(s_i, d_i)$ denotes the profit earned for defect i given that s_i time is spent at that node with severity level d_i. The parameter h allows for manipulation of the time required to reach the maximum profit before the marginal profit becomes negative.
3. **Uniform Profit Function**: This function yields profit linearly until the maximum service time is reached. It is expressed as $f_i(s_i) = \frac{s_i}{s_{max_i}}$, where $f_i(s_i)$ denotes the profit earned for defect i given that s_i time is spent at that node. The value of s_i cannot exceed the value of s_{max_i} since then the maximum profit to be earned at that node is reached.

Using these functions, the fraction of the maximum profit that can be earned at node i can be calculated. The OP module employs the uniform profit function, as it most accurately reflects the profit dynamics observed in practical applications. This choice is grounded in the assumption that the grinder operates at a consistent pace, steadily addressing the defect. The defect in this case is defined as a positive dent that is slowly sanded down until no dent can be seen. Additionally, the uniform profit function offers the advantages of simplicity and easy computation, making it a pragmatic choice for profit determination. In contrast, the exponential profit function does not provide computational benefits and complicates the profit function process unnecessarily. Similarly, the quadratic profit function is considered unrealistic because we can stop the polishing process at any time, making the negative portion of the function redundant.

Problem description for the path optimization

The OP module is designed to optimize profit by finding a path among defects within a given time constraint. This scenario results in a more complex version of the Traveling Salesman Problem, known as the Orienteering Problem, where the optimal solution may not require visiting all nodes due to time constraints. Additionally, the module accommodates variable profits.

In the current use case, the goal is to find a path to polish defects identified on a car, acknowledging that there is a possibility that only partial fixes are possible due to time limitations. Therefore, the module is restricted to an Orienteering Problem with Variable Profits (OPVP) for this scope. Other use cases might involve time windows, stochasticity, or multiple robotic arms, increasing complexity and the requirement of adjustments in the algorithms. However, these are not required within the current scope for simplicity.

The task of finding paths to polish defects can be categorized into known problems in the literature:

- **Traveling Salesman Problem (TSP)**: Requires visiting each defect while minimizing total time.

- **Orienteering Problem (OP)**: The OP is a more constrained version of TSP, where not all nodes need to be visited, but a time restriction is imposed.

For both problems, various add-ons can be applied to tailor the solution to specific use cases, although they introduce additional complexity. Below is a list of possible relevant add-ons, though it does not cover every possible option:

- **Set**: Clusters nodes, affecting profit if not all nodes in the same cluster are visited. This add-on is unnecessary for the current use case due to the absence of dependencies between nodes.
- **Team**: Allows for multiple routes to visit nodes. Not needed for the current use case, as only one robotic arm polishes the defects.
- **Time Windows**: Specifies time ranges for visiting nodes to gain profit. This add-on is not required, as each defect can be fixed at any time as long as it is within the time limit.
- **Variable Profits**: Enables partial profit if full service time is not met. Variable profits are applicable since this could be the case in the current use case.

For other use cases, the problem may differ from the one described in the OP module (OPVP). For simplicity, the OP module is currently limited to this specific use case. It will be made more flexible at a later stage to accommodate a broader range of problems.

Deterministic service times and travel times

In the OP module, it is assumed that service times and travel times are deterministic rather than stochastic. This assumption is based on the observation that the robotic arm consistently takes the same amount of time when traveling the same path, and the grinder tool requires a consistent amount of time to fix similar defects. Although these assumptions should be validated in practice, they are not anticipated to have significant variability, and thus are unlikely to cause major discrepancies between expected and actual results. Should variability prove to be an issue, service and travel times could be modelled as stochastic. However, this would increase the complexity of the OP and requires a complete revision of the OP module. Therefore, for simplicity, service and travel times are currently considered deterministic.

Algorithms to solve the OP

Using simulated data, we evaluated the performance of several algorithms to identify the best options for use in the OP module. Based on their performance in the current use case, as illustrated in Figure 27, the OP solver should use a **Branch-And-Bound Algorithm** for scenarios with up to **7 defects**. In the case of more than 7 defects the **Profit Ratio Search** heuristic proves to be the most promising.

In this use case, the threshold is set at 7 defects to utilize an exact solver. For cases involving more than 7 defects, the secondary algorithm, profit ratio search, is recommended as a heuristic. For alternative configurations or new datasets, these choices should be re-evaluated to ensure optimal performance.

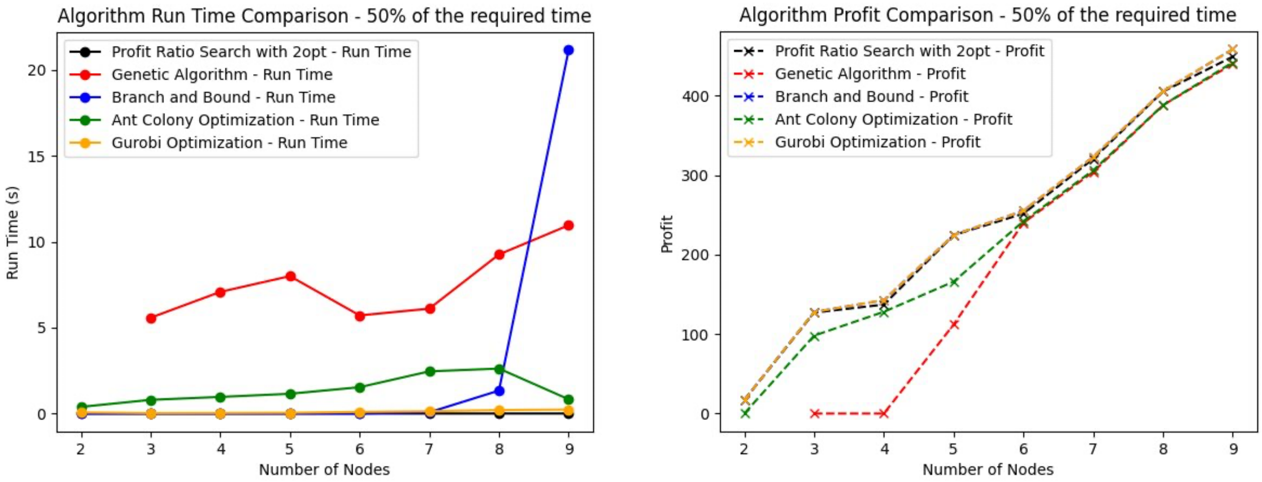


Figure 27: Performance measures of several algorithms applied to the Orienteering Problem with Variable Profits.

In determining the optimal algorithmic strategy, we considered two primary metrics: run time and profit. If the run time increases drastically, it can lead to excessive computational delays, thereby hindering the production line, as detection and polishing occur sequentially. Therefore, it is crucial to balance run time and profit, aiming for a low run time and near-optimal profit from path solver. The branch-and-bound algorithm, similar to Gurobi, is an exact method capable of finding the optimal solution. However, Gurobi is not considered due to high license costs. Branch-and-bound tends to require excessive computation time beyond 8 nodes (7 defects plus the starting point). Among the heuristics evaluated, the profit ratio search offers the best performance for the current use case, delivering a fast run time and a solution that is relatively close to optimal.

4.4 DESIGN OF PLANNING AND SCHEDULING FOR THE SR

The scheduling activities for the SR have to be orchestrated seeking a good balance between looking for defects based on historical data (exploitation) and explore new areas where defects can be found due to changes in the production process (exploration).

As regards exploitation, we have defined the following design guidelines: 1. visual inspection should focus on the areas where defects are most likely to be found (e.g., based on historical data), 2. tactile inspection has to be applied in areas that are obstructed or not easy to reach for the camera (e.g., the area underneath the doors or on their side); 3. some defects can require the combination of the two sensing modalities. The decision over which sensing mode should be preferred van be taken on the following basis.

4.4.1 INTEGRATING DIFFERENT SENSORS IN SR OPERATION SCHEDULING

Let D represent a defect at a certain location, with V and T denoting the use of visual and tactile sensors, respectively. Suppose we already have a good estimate of the probability L that a defect appears at this location. We also know:

- The accuracy and false positive rate for the visual sensor: $P(V|D) = M$, $P\left(V\middle|\overline{D}\right) = m$
- The accuracy and false positive rate for the tactile sensor: $P(T|D) = N$, $P\left(T\middle|\overline{D}\right) = n$

Assuming the two sensors operate independently, we can calculate the probability of a defect being present given:

- Only the visual sensor ($P(V|D)$),
- Only the tactile sensor ($P(D|T)$),
- Both sensors combined ($P(D|T \wedge V)$),

and can be computed using the respective Bayesian formulas. In general (see D4.1), using multiple sensors is advantageous if:

$$P(D|\overline{T} \wedge \overline{V}) \leq P(D|\overline{\mathrm{V}}) \leq P(D) \leq P(D|V) \leq P(D|T \wedge V)$$

This condition holds when each sensor is better at detecting real defects than it is at giving false positives. Essentially, each sensor's positive reading should contribute to a clearer understanding of whether a defect is present. The calculations help quantify this improvement.

Clearly, both the processing time and the probabilities associated with different locations are influenced by this decision. At the start of each work cycle, we can calculate the likelihood of detecting a defect at each location using only the visual sensor. If this probability falls below a set threshold, we may choose to use both the visual and tactile sensors together. This decision directly affects the parameters used to configure the task scheduling algorithm.

4.4.2 DESIGN OF THE TASK PLANNING SOLUTION

Our initial approach favoured a separated design for the exploration and exploitation phases. Specifically, we intended to apply a variation of the orienteering algorithm for the exploitation phase, while employing a Lloyd-based method for exploration (see D4.1).

Subsequent research has indicated that a solution based on **ergodic control** can serve as a unified conceptual framework, enabling similar treatment of both exploration and exploitation. The use of ergodic control was originally considered only for active sensing — i.e., to guide system motion once the sensing end-effector had reached a candidate area likely to contain defects (Figure 28).

Our current direction is to adopt ergodic control as a more general solution to manage the full vehicle scan.

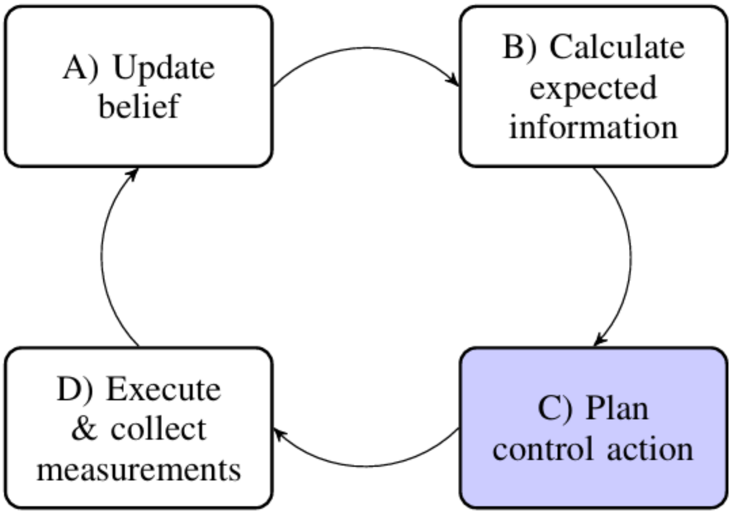

A) Update belief
B) Calculate expected information
C) Plan control action
D) Execute & collect measurements

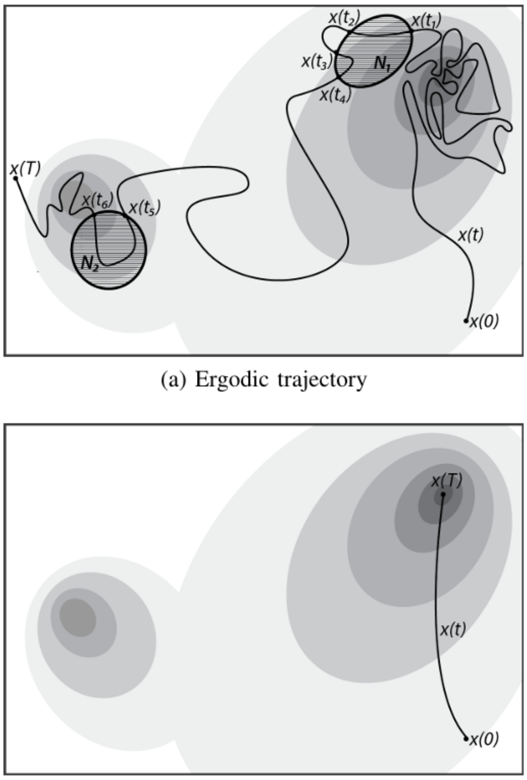
Figure 28: Typical phases of information Based sensing algorithm (courtesy [Miller2015]).

4.4.2.1 ERGODIC CONTROL

It is instructive to examine the example in Figure 29. In the lower panel, the trajectory targets points that maximise information gain, whereas in the upper panel — generated via *ergodic control* — the objective is to cover the workspace in proportion to a prior distribution. In our application, this prior corresponds to the historical probability of locating defects.

Let the search domain be X. The spatial statistics of the trajectory $x(t)$ are described by a distribution $C(x)$ associating with each point the probability of its being affected by a defect.

By using the technique shown in [Mathew2011], it is possible to set up an optimal control looking of requiring that the generated trajectory be as close as possible to the ergodic behaviour.



(a) Ergodic trajectory

(b) Information maximizing trajectory

Figure 29: Example of ergodic control (a) as opposed to information maximisation (b) [Miller2015].

Thee cost function is given by:

$$J(\boldsymbol{x}(t)) = \underbrace{\gamma\mathcal{E}[x(t)]}_{\text{ergodic cost}} + \underbrace{\int_0^T \frac{1}{2}u(\tau)^T R u(\tau).}_{\text{control effort}}$$

Classic active sensing solutions, divide the exploration and the exploitation phase switching between two cost functions. The application of ergodic control allows us to treat the two phases in the same way. The only differentiation is in the way we collect measurements. During the exploration phase, we choose our sampling policy to prioritise coverage, while in the exploitation phase we increase our sampling rate in the proximity of the hotspots. We report in Figure 30 an example from [Miller2015].

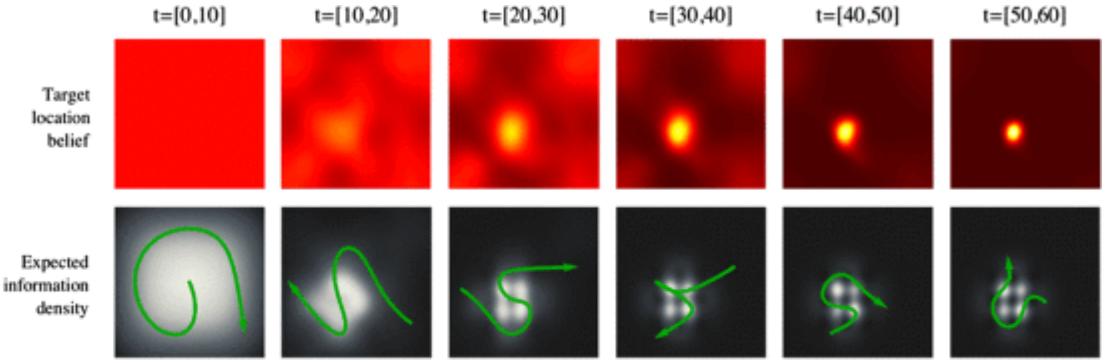


Figure 30: Ergodic location belief and expected information density reported in [Miller2015].

The example illustrates the localisation of a 2D target. The top row shows the evolution of the belief state, while the bottom row depicts the trajectory generated via ergodic control. This trajectory enables iterative information updates, progressively refining the probability density function $EID(x)$ representing the target's likely location.

Implementing ergodic control is non-trivial and requires dedicated research, which is currently underway. Consequently, our initial approach will rely on more established methods, namely orienteering and Lloyd-based navigation, as previously described.

4.5 CURRENT IMPLEMENTATION OF THE SR TASKS SCHEDULER

Even if research is currently under way to adapt to our case the ergodic framework outlined above, our current implementation is simplified and reflects the original idea.

Robot operation is presently divided into two distinct phases: *exploitation* and *exploration*. The duration allocated to each phase is currently a fixed parameter.

In the exploitation phase, we employ a variation of the orienteering methods outlined above. Drawing on previous operational history, we construct a roadmap that connects all relevant locations, distinguishing between *fly-over* segments—where the robot moves rapidly between areas—and *inspection* segments, during which the robot surveys the area while maintaining a fixed distance from the vehicle's body. This framework closely resembles the one previously described for defect removal.

With respect to the exploration phase, the robot performs a vertical sweep of the entire area in search of new locations for future inspection. During this process, regions already examined during the exploitation phase are bypassed. Typically, the time available for exploration is insufficient to achieve full coverage within the allocated interval. As a result, when a new vehicle is introduced, the exploration resumes from the point at which it was halted during the scan of the previous vehicle.

4.6 DESIGN OF THE MOTION PLANNING MODULE

Since the robot operates in a promiscuous work environment, its trajectories have to account from the possible presence of humans.

Our approach is proactive: we make prediction on the human motion for the next 1.5/2 s and modify the trajectory in real--time if required by circumstances.

We are currently considering two different approaches:

- Roadmap manipulation
- Replanning.

At the time of this writing, we are still in the stage of evaluating different approaches to find the most suitable to our application. In the write-up below, we report the main techniques that we are considering.

4.6.1 ROADMAP MANIPULATION

We begin with a roadmap that links the key locations and, where helpful, add extra ‘transition points’ solely to ease movement between them. The presence of a human—and their predicted path—renders certain arcs temporarily impassable owing to collision risk. To address this, we treat the orienteering problem with time-varying constraints and time windows: the travel time of an arc depends on when it is traversed, and each node is only accessible within a specific interval.

Although the formulation can be cast as an integer linear programme, efficient heuristics already exist. The computational bottleneck is deciding, on the fly, whether an arc might be blocked; we handle this *lazily* during exploration and are adapting the method of [Mainprice2013] for this purpose.

An alternative strategy (after [Huppi2022]) augments a probabilistic road-map with temporal data: each node is assigned an availability window considering moving obstacles, after which a time-aware variant of A* is used at query time.

4.6.2 REPLANNING

When human motion is detected *after* a path between two points has been planned, we can redesign that path directly with state-of-the-art, model-based methods [Oleinikov2024] or neural approaches to ensure collision-free travel.

Alternatively, we first generate a probabilistic road-map (PRM) using only static data. During graph exploration we perform *lazy* collision checks; if a conflict is found, we re-route locally by connecting the nodes immediately before and after the blockage via an RRT search.

A further option is to employ reactive variants of RRT and RRT*. These methods adjust the trajectory on-the-fly with minimal computation, but—lacking human-motion prediction—tend to be more conservative. The robot must slow down in anticipation of sudden obstacles, whereas predictive planning allows faster, more confident motion along trajectories that are likely to remain clear.

4.7 IMPLEMENTATION OF THE MOTION PLANNING MODULE

Our current implementation in the integrated prototype is essentially based on manipulating the roadmap using information and predictions from the human perception module. The current implementation is based on a reactive approach: we change the travel time of each arc based on the presence of humans and compute the new path using a modified version of A*. Additionally, due to the significant computation time required to find the optimal path, we parallelized the A* algorithm, achieving an average planning time of 0.25 seconds. We also implemented a multithreaded solution for path planning, utilizing a defined number of threads to continuously search for the optimal path based on updated information. This approach enables the system to disregard plans generated from unusual and inaccurate motion predictions, which typically take longer to compute.

5 MOTION GENERATION AND ACTIVE SENSING

5.1 MOTION GENERATION MODULE

The MAGICIAN control architecture will adopt the *CartesI/O* Cartesian control framework developed at IIT [Laurenzi2019]. This system enables non-expert users to perform complex motion tasks through a straightforward, auto-generated ROS-based interface.

Unlike conventional motion frameworks such as ROS *MoveIt!*, *CartesI/O* is designed for the real-time execution of Cartesian trajectories defined online, rather than pre-planned paths. It also addresses the challenge of generating these motions within a strict real-time control loop.

As illustrated in Figure 31, *CartesI/O*:

- Offers a consistent interface for interacting with Cartesian controllers.
- Automatically produces a complete ROS API for issuing control references.
- Supports use of this API even when the motion solver operates within a real-time thread.

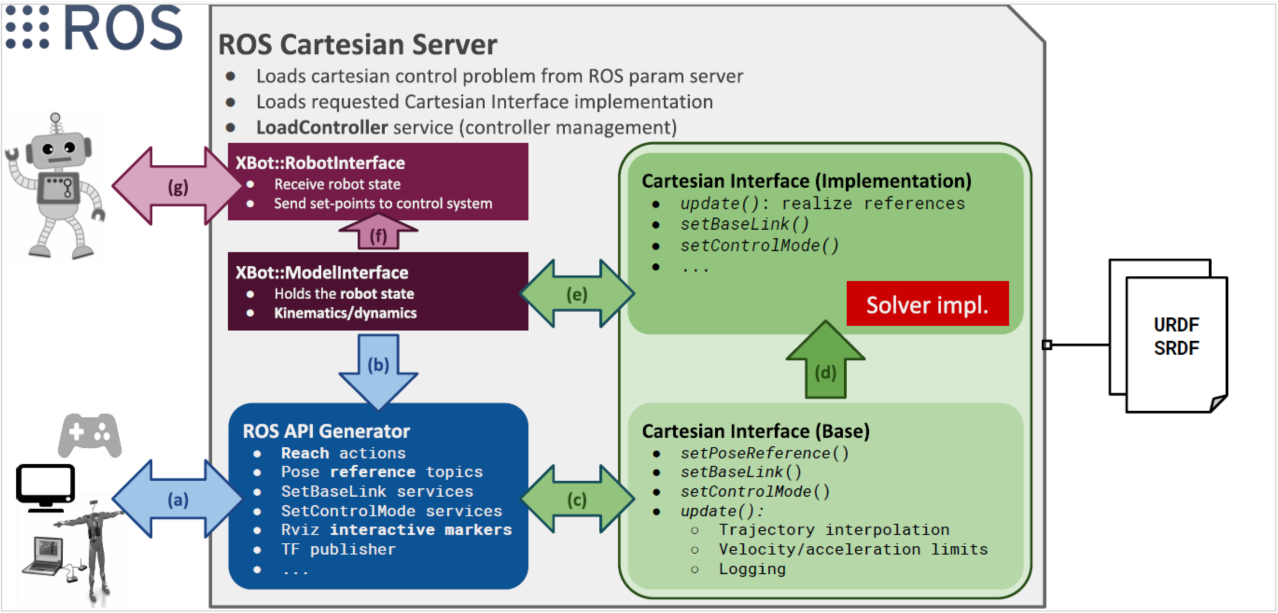


Figure 31: The main components of the software architecture with its motion modules.

An example of what is achievable with CartesI/O motion generation and the XBot middleware is a *Cartesian Impedance modulation* principle. Acting in the task space, the control law formula is now the following:

$$\tau = J^T \left(K \left(x_d - x \right) + D \left(\dot{x_d} - \dot{x} \right) + f_{ff} \right)$$

where K is the Cartesian stiffness gain, D is the Cartesian damping gain, x, x_d are the actual and the desired Cartesian tool control point pose, respectively, and f_{ff} an optional *feed-forward* force.

Operating in Cartesian space allows directional modulation of the manipulator's compliance, enabling more flexible and effective physical interaction. For instance, in a grinding or polishing task, reducing stiffness along the surface normal maintains contact while preserving high precision in other directions. Figure 32 shows this in simulation, with a mobile collaborative robot tracking a curved surface. The XBot-based plugin is ready for deployment in real-world scenarios.

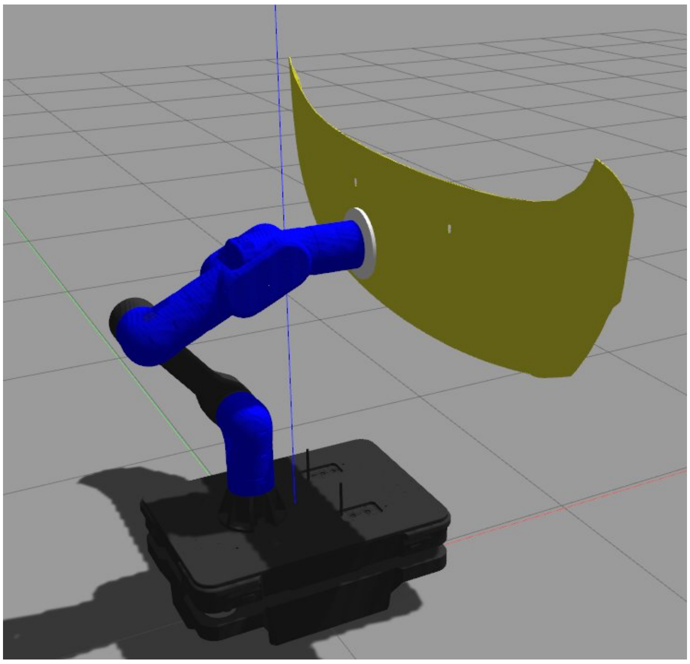


Figure 32: A mobile robot performing a surface following task in the Gazebo simulation.

The software framework used enables seamless integration of key modules for motion optimisation and safety. One such feature in the *CartesI/O* motion module is self-collision avoidance, implemented as a constraint within the task stack. It uses simplified collision meshes to represent the robot and continuously monitors for potential collisions between links, adjusting trajectories in real time to ensure safe operation.

6 CONCLUSIONS AND FUTURE ACTIVITIES

The algorithmic methodologies and technological solutions presented in this report represent the foundation tools considered and are currently under development for the realisation of the robotic system of MAGICIAN.

The development of these tools is expected to continue in the next period as their implementation and testing on the MAGICIAN robotic platform will progress. Further extensions and upgrades in the functionalities of the robotic solution will be performed based on the outcome of their experimentation and validation in the field during the execution of the defect detection and reworking use case tasks.

- The tuning of the control tools will be carried out considering the performance measured and the requirements imposed by the use case tasks. For example, additional control methodologies and/or tuning may be necessary to address issues imposed by the interaction and end-effector intrinsic vibrations.
- The physical interaction parameters of the robotic platform such as impedance settings and contact force regulation for satisfying the necessary task performance will also form and interesting topic of the follow up activities.
- Similarly, adaptation and upgrades on the task planning and scheduling tools will be guided by the results obtained and observed execution efficiency.
- Developments on the interfaces of the different tools will be necessary to facilitate their integration within the overall software and control framework and the communication among the different algorithmic and technological components.

A different ongoing activity is on the developments of components that have not reached a sufficient level of maturity to be included in the prototype. Such components will be fully integrated and tested during the final phase of the project and include:

- A solution for proactive human-aware motion planning
- The integration of motion planning and task planning
- The optimisation of the system parameters based on the quality level detected by the SR
- Models and solution for feedforward utilisation of information from the welding station to optimise the search of the defects.

7 REFERENCES

[Archetti2007] Archetti, Claudia, Alain Hertz, and Maria Grazia Speranza. "Metaheuristics for the team orienteering problem." Journal of heuristics 13.1 (2007): 49-76.

[Hogan1985-PartII] Hogan, Neville. "Impedance control: An approach to manipulation: Part II—Implementation." Journal of dynamic systems, measurement, and control 107.1 (1985): 8-16.

[Lavalle2006] LaValle, S. M. (2006). Planning algorithms: Cambridge university press, 2006.

[Mathew2011] G. Mathew and I. Mezic, "Metrics for ergodicity and design of ergodic dynamics for multi-agent system," Phys. D-Nonlinear Phenomena, vol. 240, nos. 4/5, pp. 432–442, Feb. 2011

[Miller2015] Miller, Lauren M., et al. "Ergodic exploration of distributed information." IEEE Transactions on Robotics 32.1 (2015): 36-52.

[Laurenzi2019] A. Laurenzi, E. M. Hoffman, L. Muratore and N. G. Tsagarakis, "CartesI/O: A ROS Based Real-Time Capable Cartesian Control Framework," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 591-596, doi: 10.1109/ICRA.2019.8794464.

[Hogan1985-PartI] N. Hogan, "Impedance control: An approach to manipulation: Part I-theory". Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, 107(1). https://doi.org/10.1115/1.3140702

[Muratore2023] L. Muratore, A. Laurenzi, A. De Luca, L. Bertoni, D. Torielli, L. Baccelliere, E. Del Bianco, N. G. Tsagarakis, "A Unified Multimodal Interface for the RELAX High-Payload Collaborative Robot" Sensors 2023, 23, 7735. https://doi.org/10.3390/s23187735

[Saveriano2023] Saveriano, M., Abu-Dakka, F. J., Kramberger, A., & Peternel, L. (2023). Dynamic movement primitives in robotics: A tutorial survey. The International Journal of Robotics Research, 42(13), 1133-1184.

[Mainprice2013] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in 2013 IEEE/RSJ International

Conference on Intelligent Robots and Systems, 2013, pp. 299–306.

[Huppi2022] M. Hüppi, L. Bartolomei, R. Mascaro, and M. Chli, "T-prm: Temporal probabilistic roadmap for path planning in dynamic environments," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022, pp. 10 320–10 327.

[Oleinikov2024] A. Oleinikov, S. Soltan, Z. Balgabekova, A. Bemporad, and M. Rubagotti, "Scenario-based model predictive control with probabilistic human predictions for human–robot coexistence," Control Engineering Practice, vol. 142, p. 105769, 2024.