



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 D4.3

First integrated robot solution

DUE DATE OF DELIVERABLE:

30.06.2026

ACTUAL SUBMISSION DATE:

30.06.2026

Work Package	WP4 - Robotic Platform & Interfaces
Associated Task	T4.1, T4.3, T4.4, T4.6 T2.3, T5.1
Deliverable Lead Partner	PIP
Main author(s)	Geert Driessen, Luigi Palopoli, Songqun Gao, Nikolaos Tsagarakis, Luca Muratore, Gionata Salvetti, Patrick Floor, Tahsin Sumer, Antonis Argyros, Ammar Qammaz, Elena Basei, Michael Peschl
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	31.05.2026	First Draft	Geert Driessen, Luigi Palopoli, Songqun Gao, Nikolaos Tsagarakis, Luca Muratore, Gionata Salvetti, Patrick Floor, Tahsin Sumer, Antonis Argyros, Ammar Qammaz, Elena Basei, Michael Peschl	PIP
0.2	26.06.2026	Internal Review	Daniele	UNITN

D4.3 FIRST INTEGRATED ROBOT SOLUTION



			Fontanelli	
0.3	30.06.2026	Quality Check	Annalisa Baldo	UNITN
1.0	30.06.2026	Final Version	Daniele Fontanelli, Geert Driessen	PIP, 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 deliverable presents the first integrated robot solution developed within the MAGICIAN project. The robotic platform is realised in two embodiments: the Sensing Robot, used to analyse car bodies in search of defects, and the Cleaning Robot, used to automate defect removal. Both solutions are based on the Doosan H2515 collaborative robot and share a common software infrastructure based on ROS2 and XBot2.

For the Sensing Robot, the deliverable reports the integration of visual and tactile sensing components, enabling visual inspection, tactile inspection, and combined inspection workflows. For the Cleaning Robot, it describes the grinding end effector, the vibration damping interface, and the control methods used to regulate the interaction between the tool and the workpiece.

The document also presents the integration of the robotic platform within the demonstrator at the ALT premises, including the work cell infrastructure, communication and safety interfaces, and the human machine interface. In addition, it reports the progress made on task and motion planning, defect detection and reworking workflows, human aware planning, interaction with welding information, and the configuration optimisation tool. The components described in this deliverable form the first integrated release of the MAGICIAN robotic solution, which is part of the deliverable per se, and provide the basis for the next experimentation, validation, and refinement activities.

DEVIATIONS

At the time of writing, no deviation affecting the scope of this deliverable has been identified. The integration activities have been carried out progressively, and some components and validation activities require further refinement before the final release. These remaining activities concern the experimental validation of the vibration-damping interface, the refinement of the integrated workflows at the ALT demonstrator, and the further development of the interface with welding information. They are described in the relevant sections of this document and will be addressed during the next integration and validation activities.

TABLE OF CONTENTS

1	INTRODUCTION	9
1.1	PURPOSE AND SCOPE.....	9
1.2	CONTRIBUTION TO PROJECT OBJECTIVES.....	10
1.3	RELATION TO OTHER WORK PACKAGES AND DELIVERABLES.....	11
1.4	STRUCTURE OF THE DOCUMENT.....	11
2	INTEGRATED ROBOT SOLUTION OVERVIEW	12
2.1	SYSTEM ARCHITECTURE	12
2.2	INTEGRATION STATUS.....	13
3	HARDWARE AND WORK-CELL INTEGRATION.....	14
3.1	ROBOTIC PLATFORM.....	14
3.2	SENSING END-EFFECTOR.....	14
3.2.1	Visual Sensing Module	14
3.2.2	Tactile Sensing Module	15
3.3	CLEANING END-EFFECTOR	16
3.3.1	Grinding Tool.....	16
3.3.2	Vibration-Damping Interface	18
4	SOFTWARE, CONTROL, AND DEPLOYMENT	21
4.1	CONFIGURATION OPTIMIZATION TOOL.....	21
5	HUMAN MACHINE INTERFACE.....	24
5.1	INTERFACE WITH WORK-CELL INFRASTRUCTURE AND HMI	24
5.2	REPLICABILITY AND DEPLOYMENT CONSIDERATIONS.....	25
5.3	PLANNED IMPROVEMENTS FOR THE NEXT RELEASE	25
5.4	GRAPHICAL USER INTERFACE	26
6	PLANNING AND INTEGRATED FUNCTIONALITIES.....	28
6.1	TASK AND MOTION PLANNING	28
6.1.1	Defect Detection Planning.....	28
6.1.2	Defect Reworking Planning	33
6.1.3	Human-Aware Motion Planning	37
6.2	INTEGRATED WORKFLOWS	39

6.2.1	Visual Defect Detection	39
6.2.2	Tactile Defect Detection.....	41
6.2.3	Combined Visual–Tactile Inspection.....	41
6.2.4	Defect Removal	45
6.3	INTERACTION WITH WELDING INFORMATION	45
7	SYSTEM OPERATION AND NEXT STEPS.....	46
8	BIBLIOGRAPHY	46

LIST OF FIGURES

Figure 1: Robotic set-up with car body.....	13
Figure 2: Overview of the Tactile Perception Module. (a) Exploded view and (b) assembled prototype of the tactile module. The system integrates a six-axis force/torque sensor (A), a tri-axial accelerometer (B), and a compliant TPU-based comb-like contact interface.	16
Figure 3: Modified grinding tool with relay switch, input connector and mounting interface	17
Figure 4: The grinding tool integrated on the Cleaning Robot composed by the ATI FT Mini-45 and the DeWALT DCW210N-XJ 18V XR 125mm grinder.	17
Figure 5: The 3D printed prototypes of the two vibration isolators studied.	18
Figure 6: The optimization process employed for tuning the mechanical design of the vibration isolators.	19
Figure 7: The results from the study influence of the beam angle relative to the horizontal base plate of the Stewart–Gough platform.	20
Figure 8: The realized new 3–3 Stewart–Gough vibration isolator module.	20
Figure 9: The integrated MAGICIAN grinding tool composed of the Stewart-Based Folded-Beam Passive Vibration Isolator, the FT sensor, and the DeWalt DCW210 grinder.....	21
Figure 10: Configuration Optimization Tool GUI.....	24
Figure 11: Human-machine interface in the MAGICIAN work cell.....	25
Figure 12: MaGULcian main viewer GUI.	27
Figure 13: MaGULcian Cleanup Manager.....	28
Figure 14: Diagram of the proposed active sensing framework.	30
Figure 15: 2D ergodic trajectory generated for the inspection of the first surface in the sequence, considering a target PDF for ergodic control that approximates a uniform distribution.	31
Figure 16: The corresponding trajectory mapped back onto the 3D coordinates of the mesh frame.	32
Figure 17: Path Optimizer Framework.	35
Figure 18: Representation of the detected human being (in green) and of the robot (in red), with the graph nodes depicted as 3D red points in space.....	38
Figure 19: Ergodic trajectory projected onto the mesh frame (purple spheres), together with the defects detected by the visual sensor during trajectory execution (yellow spheres).....	40
Figure 20: The updated probability density function (PDF) and the corresponding ergodic trajectory over the 2D rectangular search space.	40
Figure 21: Ergodic trajectory executed on a door, and next PDF computed with force-position data.	41

LIST OF ABBREVIATIONS

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 in the MAGICIAN project on the development and integration of the first integrated robot solution. The platform is realised in two distinct embodiments: the **Sensing Robot (SR)**, used to analyse car bodies in search of defects, and the **Cleaning Robot (CR)**, used to automate the removal of defects.

This document follows on D4.2, which reported the first prototype of the robotic platform and the initial integration of its main hardware and software components. D4.3 focuses on the subsequent integration step, reporting the first release of the integrated robot hardware and control components. The work includes the refinement of the sensing and cleaning end-effectors, the integration of the tactile and visual sensing devices, the development of the grinding end-effector and vibration isolation device, and the implementation of the control, planning, and configuration tools required for system operation.

The main challenges addressed during this phase include:

- the integration of the SR and CR components in a common robotic architecture
- the development of compact and functional interfaces for visual and tactile defect detection
- the implementation of impedance and force control methods for safe and adaptive interaction with the car-body surface
- the collection and use of human grinding demonstrations for learning-based reworking strategies
- the development of planning, scheduling, and human-aware motion planning algorithms
- the implementation of configuration tools to support deployment and future adaptation of the platform

The activities reported in this deliverable have been supported by the integration work carried out within the consortium, in particular the preparation of the demonstrator at ALT and the successive integration weeks. These activities have allowed the partners to test the robotic platform in increasingly realistic conditions and to identify the remaining refinements needed before the final validation phase.

Overall, the artefacts presented in this document form the first integrated robot solution of the MAGICIAN project and provide the technological basis for the next experimentation and validation activities at the consortium end-user premises.

1.1 PURPOSE AND SCOPE

The purpose of this deliverable is to describe the first integrated robot solution developed within WP4. The focus is on the hardware and control components that enable the MAGICIAN platform to operate in its two versions: defect detection and defect

removal. Both robot embodiments share the same robotic basis, the Doosan H2515 cobot, and use a common software infrastructure based on XBot2 and ROS2.

For the Sensing Robot, the deliverable reports the integration of visual and tactile perception components into the robotic platform. The same platform can be used to perform visual inspection, tactile inspection, or combinations thereof, preparing the ground for the multi-modal sensing strategies developed in parallel in WP3.

For the Cleaning Robot, the deliverable describes the grinding end-effector, the vibration isolation device, and the control methods used to regulate the interaction between the tool and the workpiece. The platform has also been used to collect human grinding data, including pose and wrench information, which will support the development of learning-based reworking policies.

The deliverable also discusses the planning and scheduling algorithms developed to prioritise and execute defect detection and reworking tasks. These include human-aware planning methods, active sensing strategies, and extensions toward more complex surface geometries.

Finally, the document reports the progress on the configuration optimisation tool, which is intended to support the deployment, parameter tuning, and adaptation of the platform to future use cases and third-party components.

1.2 CONTRIBUTION TO PROJECT OBJECTIVES

This report details work that is central to the MAGICIAN project. The achievements outlined in this deliverable contribute to the following project 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.
- O6: A human-centred approach to human-robot collaboration, promoting usability, safety, and trustworthiness.
- O7: Demonstration of the prototypes in operational scenarios.
- O8: Expansion of MAGICIAN's scope and applicability via Financial Support to Third Parties.

The development of the robotic hardware, end-effectors, control methodologies, and planning tools described in this deliverable is essential for enabling the MAGICIAN platform to perform defect detection and reworking operations in realistic industrial

conditions. These contributions support the project objective of providing an integrated, adaptable, and safe robotic solution for defect handling in manufacturing environments.

1.3 RELATION TO OTHER WORK PACKAGES AND DELIVERABLES

The work described here contains 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 SR and the CR. The main relations with other work packages are the following.

- WP2 – Use case definition and platform design: The design and implementation of the robotic platform are driven by the requirements of the automotive use case. These requirements influence the end-effector design, safety constraints, usability, reworking effectiveness, and integration with the production-like demonstrator.
- WP3 – Data acquisition and skills learning: WP3 develops the visual, tactile, and human perception modules integrated in the robotic architecture presented in this deliverable. Planning, scheduling, and motion control rely on the outcome of the perception pipeline and on the prediction of human motion. At the same time, the constraints of the robotic platform influence the design and operation of the perception modules.
- WP5 – Integration and performance analysis: The robotic platform described in this deliverable is a key component of the MAGICIAN demonstrator. The integration activities in WP5 provide the context in which the SR and CR components are tested and progressively validated.
- WP6 – Cascade funding management: The platform and its configuration tools will support the integration of additional components and new use cases emerging from the open calls. The WP4 results will therefore be used to provide technical support and to facilitate future extensions of the robotic system.
- WP7 – Dissemination, communication, and exploitation: The integrated robot solution is one of the main technical outcomes of the project and provides material for dissemination, demonstration, and exploitation activities.

1.4 STRUCTURE OF THE DOCUMENT

This document is organised in seven chapters, each covering a different aspect of the integrated robot solution developed in WP4.

Chapter 2 provides an overview of the integrated robot solution, including the system architecture, the functionalities of the Sensing Robot and Cleaning Robot, and the current integration status.

Chapter 3 describes the hardware and work-cell integration. It presents the robotic platform, the sensing and cleaning end-effectors, and the electrical, communication, and safety interfaces required for operation.

Chapter 4 focuses on the software, control, and deployment components. It describes the ROS2-based infrastructure, the XBot2 real-time control layer, the interfaces between modules, the control components, and the configuration optimisation tool.

Chapter 5 presents the human-machine interfaces. It describes the interface with the work-cell infrastructure and industrial HMI, replicability and deployment considerations, planned improvements for the next release, and the graphical user interface developed for the integrated workflows.

Chapter 6 presents the planning and integrated functionalities of the system. It describes task and motion planning for defect detection and reworking, human-aware motion planning, the main integrated workflows for visual, tactile, and combined inspection, defect removal, and the interaction with welding information.

Chapter 7 reports the system operation and validation activities to be executed on the integrated robotic system.

2 INTEGRATED ROBOT SOLUTION OVERVIEW

2.1 SYSTEM ARCHITECTURE

The hardware integration framework enables the coordination of mechanical, electrical, and control components within a unified work-cell architecture. Established at the ALT premises in Istanbul, the demonstrator layout features two collaborative robots (cobots) mounted on dedicated linear sliders, enabling extended workspace coverage and precise accessibility to all vehicle body surfaces. The work-cell is organized into two sequential operational stations: Station 1 (Sensing): Accommodates the sensing robot equipped with visual and tactile inspection tools. Station 2 (Cleaning): Hosts the cleaning robot configured with a grinding end-effector for surface correction tasks. The vehicle body traverses the work-cell via conveyors driven by asynchronous motors equipped with inductive sensors for position detection, ensuring deterministic material flow and precise part positioning. Critical control functions are centralized within an industrial electrical cabinet housing a Siemens S7-1500 PLC, a safety-rated controller, multiple motor drives, and power distribution units. This integrated infrastructure, combined with PROFINET-based communication, establishes a deterministic, safety-compliant automation system capable of coordinating complex multi-robot operations while maintaining real-time responsiveness through an industrial HMI panel. The slider system provides extended linear motion with a usable stroke sufficient for full workspace coverage, positioning the robots at an elevated height above ground level to ensure effective access to upper and lateral vehicle surfaces while maintaining safe clearance for human operators.

All electrical and signal cabling is routed through dedicated trays and conduits to minimize electromagnetic interference, with power and communication lines segregated according to industrial standards. Ultimately, this comprehensive hardware

integration demonstrates the feasibility of deploying advanced robotic technologies within realistic industrial settings, supporting the MAGICIAN project's objective to deliver scalable, replicable solutions for quality control and surface treatment in automotive manufacturing.

2.2 INTEGRATION STATUS

The integration of the demonstrator system has progressed through multiple design and planning phases, with the physical infrastructure on-site now fully prepared. The work-cell layout, slider systems, conveyor infrastructure, and electrical cabinet have been established, calibrated, and validated. Additionally, the Siemens PLC control infrastructure, HMI panel, safety interlocks, and power distribution systems have been successfully tested to ensure compliance with industrial safety standards. Integration sessions and workshops facilitated the validation of visual and tactile sensor data acquisition pipelines, as well as the task orchestration system used for high-level control. A snapshot of the two integrated robot solutions can be seen in Figure 1: Robotic set-up with car body..



Figure 1: Robotic set-up with car body.

3 HARDWARE AND WORK-CELL INTEGRATION

3.1 ROBOTIC PLATFORM

The robotic platform selected for the demonstrator system is the Doosan H2515 collaborative robot. Belonging to the H-Series, this high-payload cobot is designed for industrial applications requiring high precision and safe human-robot interaction. In the demonstrator work-cell, two H2515 units are deployed in a complementary configuration. The inspection unit performs vision-assisted and tactile inspection tasks. The processing unit executes surface-grinding operations requiring continuous force application and accurate trajectory tracking under compliance control. Both robots are integrated with the XBot2 middleware, enabling deterministic real-time control, Cartesian motion execution, and impedance and force control.

3.2 SENSING END-EFFECTOR

The sensing robot is equipped with an integrated end-effector that combines visual and tactile sensing capabilities for the inspection of car-body surfaces. The end-effector has been designed to support different operational strategies depending on the characteristics of the surface and the confidence of the inspection results. Visual sensing can be used for non-contact scanning of larger areas, while tactile sensing provides complementary information through direct interaction with the surface. The two modalities can also be used sequentially, with tactile exploration refining the analysis of regions that cannot be classified reliably using vision alone.

The integrated design allows the same robotic platform to perform visual inspection, tactile inspection, or combined visual-tactile inspection without changing the underlying robot architecture. The following subsections describe the visual and tactile sensing modules integrated into the SR end-effector.

3.2.1 VISUAL SENSING MODULE

The visual sensing module is based on a SONY XCG-CP510 polarisation camera equipped with a 16 mm lens. The camera simultaneously captures images at four linear polarisation angles, enabling the detection of small variations in light reflection caused by surface irregularities. Six programmable polarised LED light sources provide illumination from different directions, while three Time-of-Flight range sensors support distance estimation and positioning of the sensor with respect to the inspected surface. The module is used for close-range, non-contact inspection and provides the visual data required for defect detection and classification. The details of the visual sensor design can be found in D4.2.

3.2.2 TACTILE SENSING MODULE

The tactile system is based on a compact multimodal sensing module designed to reproduce the informative aspects of human tactile inspection. IIT integrated two complementary sensing modalities: a six-axis force/torque sensor, ATI Nano17, and a tri-axial MEMS accelerometer, Analog Devices ADXL335. The force/torque sensor captures quasi-static interaction forces and moments associated with the global contact state and surface geometry, while the accelerometer records high-frequency vibrations generated by local discontinuities, surface irregularities, and transient contact events. The sensing hardware is integrated into a custom tactile end-effector, the Tactile Perception Module (TPM), whose exploded and assembled configurations are reported in Figure 2. The TPM includes a compliant contact interface fabricated from thermoplastic polyurethane (TPU) with a Shore A 95 hardness. This material provides a semi-rigid interaction with the inspected metallic surface, ensuring non-destructive contact while preserving sufficient mechanical stiffness to transmit force and acceleration transients to the sensors.

The contact interface has a rectangular footprint of approximately 21 x 50 x 8 mm and includes 44 S-shaped pins, each 8 mm high and with an approximate contact area of 4.5 mm². The S-shaped geometry was selected to improve the mechanical transmission of surface-induced vibrations. Unlike a rigid vertical probe, the compliant curved structure behaves as a spring-like contact interface: when the probe slides over a surface defect, the deformation and elastic recovery of the pins generate dynamic signatures that can be captured by the accelerometer. This design allows the tactile module to amplify subtle surface variations that may be difficult to detect through force sensing alone. The integration of force and vibration sensing is therefore central to the MAGICIAN tactile perception strategy. Force measurements provide information about stable contact, normal load, and macroscopic interaction conditions, while acceleration measurements provide sensitivity to fast local events and micro-scale changes in surface profile. The resulting multimodal signal forms the basis for defect detection and classification.

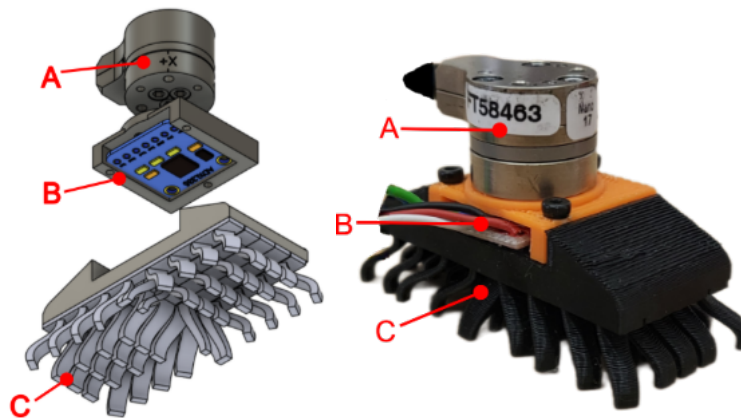


Figure 2: Overview of the Tactile Perception Module. (a) Exploded view and (b) assembled prototype of the tactile module. The system integrates a six-axis force/torque sensor (A), a tri-axial accelerometer (B), and a compliant TPU-based comb-like contact interface.

3.3 CLEANING END-EFFECTOR

3.3.1 GRINDING TOOL

To enable integration of the grinding tool into the robotic system, the body of the selected **DeWALT DCW210N-XJ 18V XR 125mm** grinder was modified to include two additional features.

First, a relay switch was integrated into the grinding tool's body to enable ON/OFF control via a dedicated electrical interface connected to the robotic system control unit. This enables the tool state to be controlled via the Doosan internal digital interface, allowing coordinated activation and deactivation of the device with the robot's motions and actions. The connection between the tool and the Doosan robot interface is provided through a connector mounted on the body of the grinding tool.

Second, the upper side of the tool body was modified to integrate a mounting interface, allowing the grinding tool to be fixed to the wrist of the robotic arm, as shown in Figure 3.

D4.3 FIRST INTEGRATED ROBOT SOLUTION

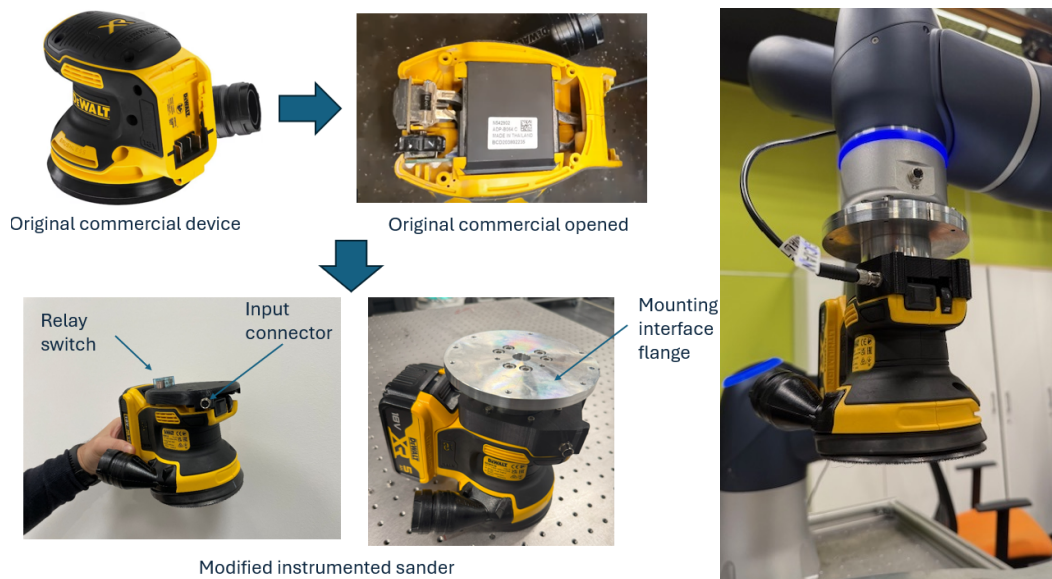


Figure 3: Modified grinding tool with relay switch, input connector and mounting interface

The grinding tool was also integrated with an ATI FT Mini-45 model to enable closed-loop force control at the end effector during the grinding operation. All MAGICIAN Cleaning Robot prototypes were equipped with this device, as shown in Figure 4.



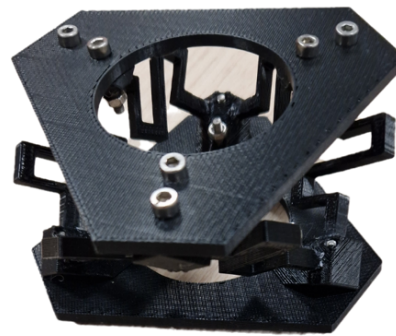
Figure 4: The grinding tool integrated on the Cleaning Robot composed by the ATI FT Mini-45 and the DeWALT DCW210N-XJ 18V XR 125mm grinder.

3.3.2 VIBRATION-DAMPING INTERFACE

Using the data collected from the study of the grinding tool vibrations (reported in D4.2), 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 characterized by its structural simplicity and potential for rapid iterative refinement (Figure 5). 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 5: 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.

To ensure the isolator met stringent performance and weight requirements, we conducted a numerical optimization using Sequential Quadratic Programming (SQP). This approach was used to optimize the geometry and material distribution of the CAD model representing the isolator (Figure 6). 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

```

1: Initialize:
2: Design variables:  $\mathbf{x} = [L_1, L_2, L_3, L_4, L_5, B_1, B_2, B_3, B_4, B_5]^T$   $\triangleright$  With  $L_2 = L_4, B_2 = B_4, B_1 = B_3 = B_5$ 
3: Bounds:  $\mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}$ 
4: Hessian  $H_0 \leftarrow I, \epsilon \leftarrow 0.001, k \leftarrow 0$ 
5: Objective:  $\min M(\mathbf{x})$ , Constraint:  $f_1(\mathbf{x}) \leq 95$  Hz
6: while  $\|\Delta \mathbf{x}\| > \epsilon$  or constraints not satisfied do
7:   Step 1: FEA Evaluation
8:   Compute  $M_k \leftarrow \text{Mass}(\mathbf{x}_k)$ 
9:   Compute  $f_{1,k} \leftarrow \text{FirstFrequency}(\mathbf{x}_k)$ 
10:  Step 2: Sensitivity Analysis
11:   $\nabla M_k \leftarrow \left[ \frac{\partial M}{\partial L_1}, \dots, \frac{\partial M}{\partial B_5} \right]^T$ 
12:   $\nabla f_{1,k} \leftarrow \left[ \frac{\partial f_1}{\partial L_1}, \dots, \frac{\partial f_1}{\partial B_5} \right]^T$ 
13:  Step 3: QP Subproblem
14:   $\min_{\Delta \mathbf{x}} \frac{1}{2} \Delta \mathbf{x}^T H_k \Delta \mathbf{x} + \nabla M_k^T \Delta \mathbf{x}$ 
15:  s.t.  $\nabla f_{1,k}^T \Delta \mathbf{x} + f_{1,k} \leq 95$ 
16:   $\mathbf{x}_{\min} \leq \mathbf{x}_k + \Delta \mathbf{x} \leq \mathbf{x}_{\max}$ 
17:  Step 4: Solve QP
18:  Compute  $\Delta \mathbf{x}_k$  and multipliers  $\lambda_{k+1}$ 
19:  Step 5: Line Search
20:   $\alpha \leftarrow 1$ 
21:  while  $\phi(\mathbf{x}_k + \alpha \Delta \mathbf{x}_k) > \phi(\mathbf{x}_k) + \gamma \alpha \nabla \phi^T \Delta \mathbf{x}_k$  do
22:     $\alpha \leftarrow \alpha/2$ 
23:  end while
24:  Step 6: Update Variables
25:   $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha \Delta \mathbf{x}_k$ 
26:  Propagate  $L_4 \leftarrow L_3, B_4 \leftarrow B_3$ 
27:  Step 7: BFGS Update
28:   $\mathbf{s}_k \leftarrow \mathbf{x}_{k+1} - \mathbf{x}_k$ 
29:   $\mathbf{y}_k \leftarrow \nabla L_{k+1} - \nabla L_k$ 
30:  if  $\mathbf{y}_k^T \mathbf{s}_k > 0$  then
31:     $H_{k+1} \leftarrow H_k - \frac{H_k \mathbf{s}_k \mathbf{s}_k^T H_k}{\mathbf{s}_k^T H_k \mathbf{s}_k} + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}$ 
32:  end if
33:   $k \leftarrow k + 1$ 
34: end while
35: Output:  $\mathbf{x}^* \leftarrow \mathbf{x}_k, M^* \leftarrow M_k, f_1^* \leftarrow f_{1,k}$ 

```

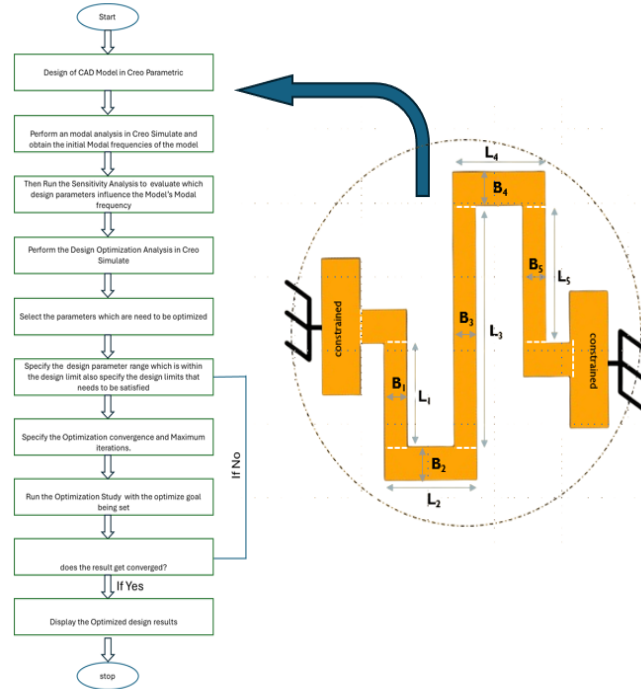


Figure 6: The optimization process employed for tuning the mechanical design of the vibration isolators.

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 characterizing the vibrational modes of a mechanical structure, as it reveals how the system naturally responds to external excitations at different 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 aimed to understand how changes in geometry affect 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 7). For instance, a 90-degree configuration yielded a modal Von Mises stress of approximately 57.5 MPa, whereas the 60-degree and 45-degree configurations yielded 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.

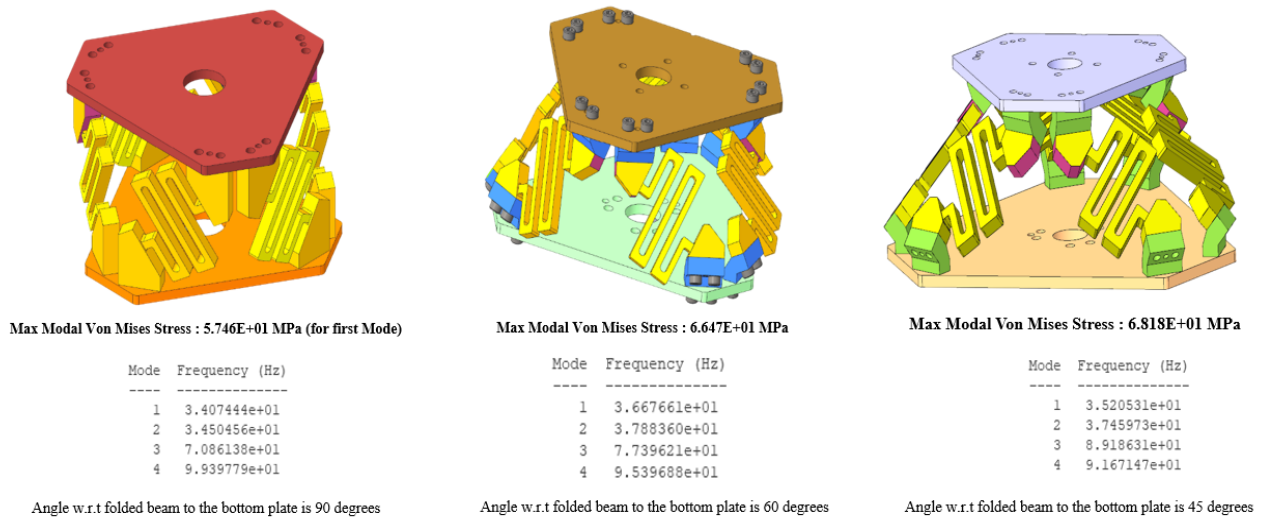


Figure 7: The results from the study influence of the beam angle relative to the horizontal base plate of the Stewart-Gough platform.

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.

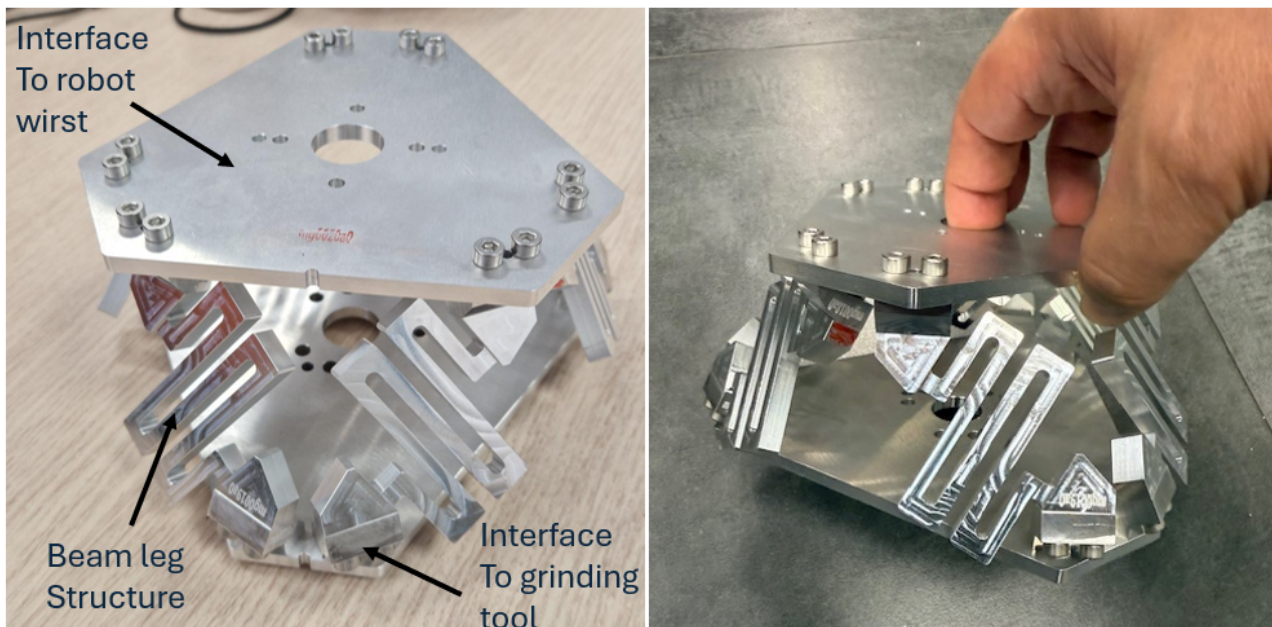


Figure 8: The realized new 3-3 Stewart-Gough vibration isolator module.

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 environments, such as

grinding. This synergy between mechanical isolation design and an active control architecture that leverages force/torque sensing, combined with real-time feedback from the impedance control framework, will allow the robot to dynamically adapt to disturbances and reduce the effects of grinding vibrations on the robot's hardware and control. A prototype of the new 6DOF vibration isolator is shown below.

The prototype of the vibration isolation module (Figure 8) will be validated using the same setup used for the grinding tool operations study to evaluate its performance in attenuating the generated vibrations.

The MAGICIAN integrated grinding tool with the vibration-damping interface is shown in Figure 9.



Figure 9: The integrated MAGICIAN grinding tool composed of the Stewart-Based Folded-Beam Passive Vibration Isolator, the FT sensor, and the DeWalt DCW210 grinder.

4 SOFTWARE, CONTROL, AND DEPLOYMENT

The detailed description of the software architecture and of the control components, including the control components, the motion control designed solution, and the impedances and force control can be found in D4.2. In this chapter we focus on the newly designed and integrated components, namely the one pertaining the configuration and optimization tool.

4.1 CONFIGURATION OPTIMIZATION TOOL

The Configuration Optimization Tool (COT) is a decision-support and orchestration platform developed as part of the MAGICIAN project to help engineers adapt the robot system to new or evolving inspection use cases. The core idea is that when an operator

or systems engineer needs to deploy MAGICIAN for a different task whether that is a minor variation of an existing job or a completely new inspection scenario, they do not need to manually trace through the robot software to figure out what needs changing. Instead, COT provides a guided, structured process that converts a use case description into concrete, actionable recommendations for each of the MAGICIAN software modules.

The starting point for a new use case in COT is a questionnaire. This questionnaire is divided into thematic sections: robot configuration, sensors and perception, material and surface properties, defect detection, and time and priority constraints. Across these sections, questions collectively capture the key characteristics of a deployment scenario, such as the number of robot arms, the camera type in use, whether additional sensors like tactile or force sensing are needed, the types of surfaces and defects involved, and whether there are time constraints on operations. Each question is designed not as a generic survey but as a direct reflection of the parameters and design choices that are known to affect one or more of the MAGICIAN modules. This questionnaire design is the result of a knowledge-gathering process with module owners who provided input on which operational characteristics drive parameter changes in their module, what those parameters are, and what values or ranges are acceptable. This part is also still under development, since not all modules are finalized yet.

This collaboration with module owners is a central part of the COT design philosophy. For each module, we worked through which characteristics of a use case would require the module to be reconfigured, retrained, or at minimum reviewed by a qualified engineer. The Orienteering Solver is sensitive to the number of arms, the presence of time windows, and the profit structure of the inspection task, because each of these shifts the underlying combinatorial optimisation problem to a different variant from the standard Orienteering Problem to the Team Orienteering Problem, the Time-Windowed variant, or a Variable Profit formulation. The Tactile Classifier and Vision Classifier are both machine learning models, meaning they need retraining whenever materials, defect types, or object sizes change, because the underlying data distributions shift with these characteristics.

Once a user fills in the questionnaire and saves the use case, COT runs its rule-based impact engine. This engine evaluates each answer against the known rules derived from module owner input and produces a structured impact report per module. Each impact entry describes the specific characteristic that triggered it, what aspect of the module is affected, what change is needed, and what type of action is required, either a reconfiguration, a retraining, or a review by a human expert. Alongside these impacts, the engine can raise flags for cases where automated guidance is not sufficient and human oversight from a robotics engineer is explicitly required.

The results are shown in a Module Dashboard, where each of the five MAGICIAN modules is represented as a tile. Each tile clearly communicates the required action type and any active flags, so engineers can immediately see where attention is needed. Clicking on a module tile opens a detailed view that shows all individual impact entries, the reasoning

behind each one, and any flags that need to be acknowledged. Engineers can acknowledge flags, review the recommendations, and when they are satisfied that the module configuration is correct, mark the module as validated. The Deploy to ROS button at the bottom of the dashboard only becomes active once all affected modules have been validated, ensuring that no deployment happens without explicit human sign-off on every impacted component.

A second key feature of the COT is the offline parameter computation step. For each module, COT can run a dedicated compute script that reads the questionnaire answers and derives the specific ROS parameter values needed for that module without requiring a live ROS environment. These computed values are stored in a result file per module and serve as the source of truth for what will eventually be sent to ROS. This offline computation approach means that all configuration decisions can be reviewed, audited, and if necessary corrected before any command is issued to the running robot system. Module owners need to provide these scripts and are part of the close collaboration between the COT and the module owners.

The final step sending configuration updates to ROS is handled through a dedicated ROS interface layer in COT. Each module has its own ROS interface that translates the computed parameter values into a precise list of ROS 2 commands, typically `ros2 param set` calls targeting specific nodes, or ROS 2 service calls for more complex update operations. COT is intentionally built outside the ROS framework, meaning it does not run inside ROS and does not depend on a live ROS environment for its decision-making logic. This separation keeps the ROS runtime stable and predictable while giving COT full flexibility as an external orchestration tool.

For the machine learning modules, the Tactile Classifier and the Vision Classifier the workflow in COT goes beyond parameter reconfiguration. When material types, defect categories, or object sizes change, these models need to be retrained on new data. COT will be designed to eventually host this training pipeline directly, so that an engineer can trigger model training from within the tool, monitor progress, and then pass the resulting model checkpoint through to the ROS deployment step. At that point, the relevant ROS parameter pointing to the active model file would be updated to reference the new checkpoint, completing the full loop from use case definition to deployed model. The training integration is currently in its placeholder phase, with the interfaces and data flow already defined and waiting for the real training pipelines to be connected. Together, these components make COT a practical bridge between operational requirements and the ROS configuration layer. It captures expert knowledge from module owners in a structured, rule-driven form, guides engineers through the configuration process with appropriate human checkpoints, computes parameter values offline so they can be reviewed before deployment and then fires targeted update commands into ROS when the team is ready. Whether the task at hand is a small tweak to an existing use case or the introduction of a completely new inspection scenario, COT provides a consistent and traceable workflow that reduces the risk of misconfiguration and keeps the MAGICIAN system aligned with real-world project goals. A snapshot of the

developed system GUI can be found in Figure 10.

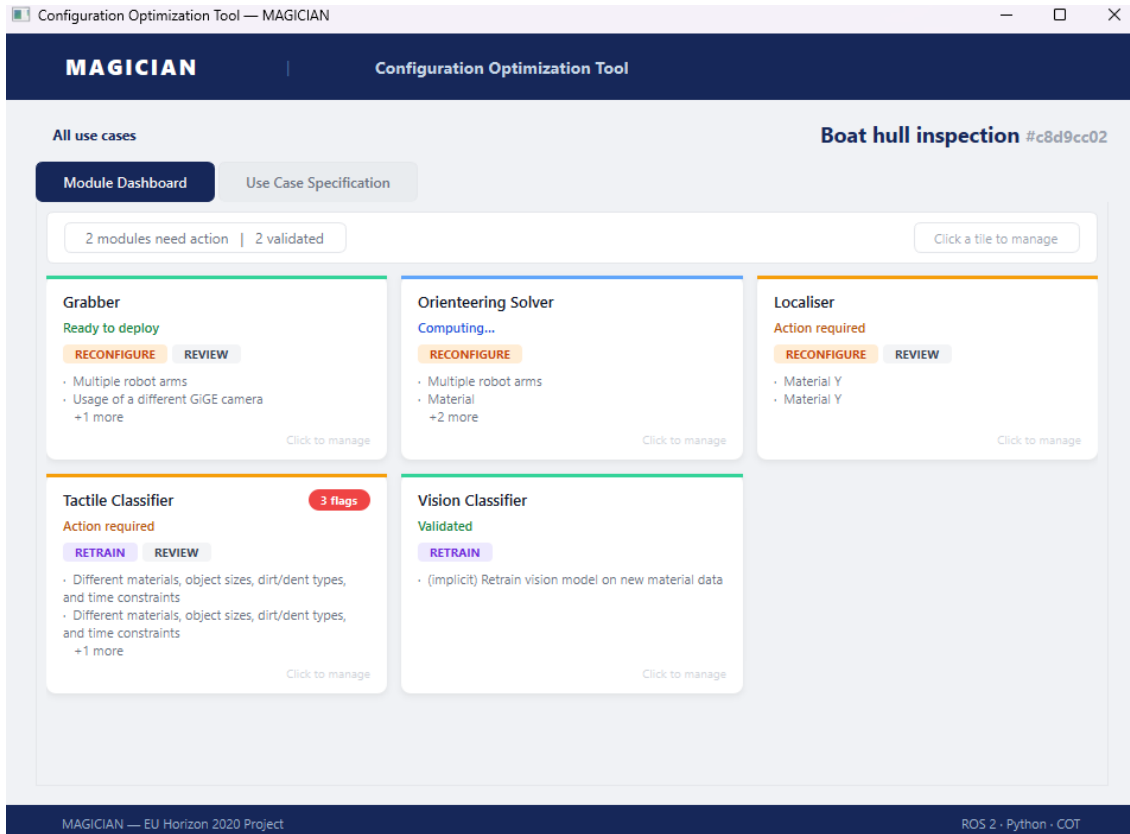


Figure 10: Configuration Optimization Tool GUI.

5 HUMAN MACHINE INTERFACE

5.1 INTERFACE WITH WORK-CELL INFRASTRUCTURE AND HMI

The 12-inch Siemens Comfort Panel has been used as the human-machine interface (HMI) in the demonstrator cell. The device supports industrial communication protocols such as PROFINET enabling integration with system PLC. It provides an intuitive interface for visualization, control, and monitoring of automation processes via the TIA Portal environment (Figure 11).

In the proposed system, the HMI is organized into multiple functional screens, including a system status dashboard, a PLC-ROS communication monitoring page, and dedicated visualization interfaces for process-related operations. This structure allows operators to monitor real-time system states, supervise cross-platform communication, and interact with control functions through a unified interface.



Figure 11: Human-machine interface in the MAGICIAN work cell.

5.2 REPLICABILITY AND DEPLOYMENT CONSIDERATIONS

The proposed system architecture is designed to support replicability and facilitate deployment across different industrial environments. The use of standardized industrial communication protocols, including PROFINET and OPC UA, provides compatibility with common automation infrastructure. This approach enables deployment in different manufacturing setups while maintaining consistent system behaviour and performance under defined operational conditions.

5.3 PLANNED IMPROVEMENTS FOR THE NEXT RELEASE

The next release will focus on completing and validating the integrated workflows under representative operating conditions. The planned activities include further testing of the vibration-damping interface and grinding operations, refinement of the combined visual-tactile inspection workflow, further development of the exchange of welding information, and completion of the remaining calibration and safety-related behaviours. Additional work will address deployment support and the extension of the graphical interfaces. Where relevant, augmented-reality and digital-twin technologies developed through cascade-funding activities will be considered for simulation, monitoring, and

validation.

5.4 GRAPHICAL USER INTERFACE

A new graphical user interface (MaGULcian) was developed to provide an intuitive environment for human–robot interaction throughout the complete workflow. The interface was designed to integrate data visualization, robot-task management, and real-time execution monitoring within a unified framework, enabling seamless coordination between the different operational stages. In particular, the GUI combines 3D scene visualization with interactive control features, enabling operators to supervise and coordinate both virtual and real-world system execution through the same interface (Figure 12).

The software architecture is organized around two principal operational modules: the Inspection Manager and the Cleanup Manager. The Inspection Manager supervises the defect-detection pipeline, allowing operators to initiate scanning procedures in the real world, monitor progress, and receive inspection results. Detected defect candidates are stored and visualized within the virtual environment, where they can be directly selected through the 3D interface and assigned to subsequent robotic operations, enabling a continuous transition between inspection and intervention.

The Cleanup Manager extends this workflow by enabling cleanup operations for detected defects through a dedicated GUI section, where operators can configure and execute grinding tasks for selected defects. Throughout the entire process, defect information is continuously updated and stored directly within the interface, enabling future retrieval and historical tracking.

To improve usability, the GUI employs a color-coded feedback system that enables rapid visual interpretation of information. Colours assigned to defect candidates allow operators to quickly identify their type. In addition, the Overview panel provides direct access to detailed information for each detected defect candidate, enabling efficient inspection and management of associated data.

The main viewer GUI, illustrated in the first figure below, is displayed upon software startup and acts as the primary access point to the system functionalities. The interface allows users to navigate between different projects, load different meshes and restore information quickly.

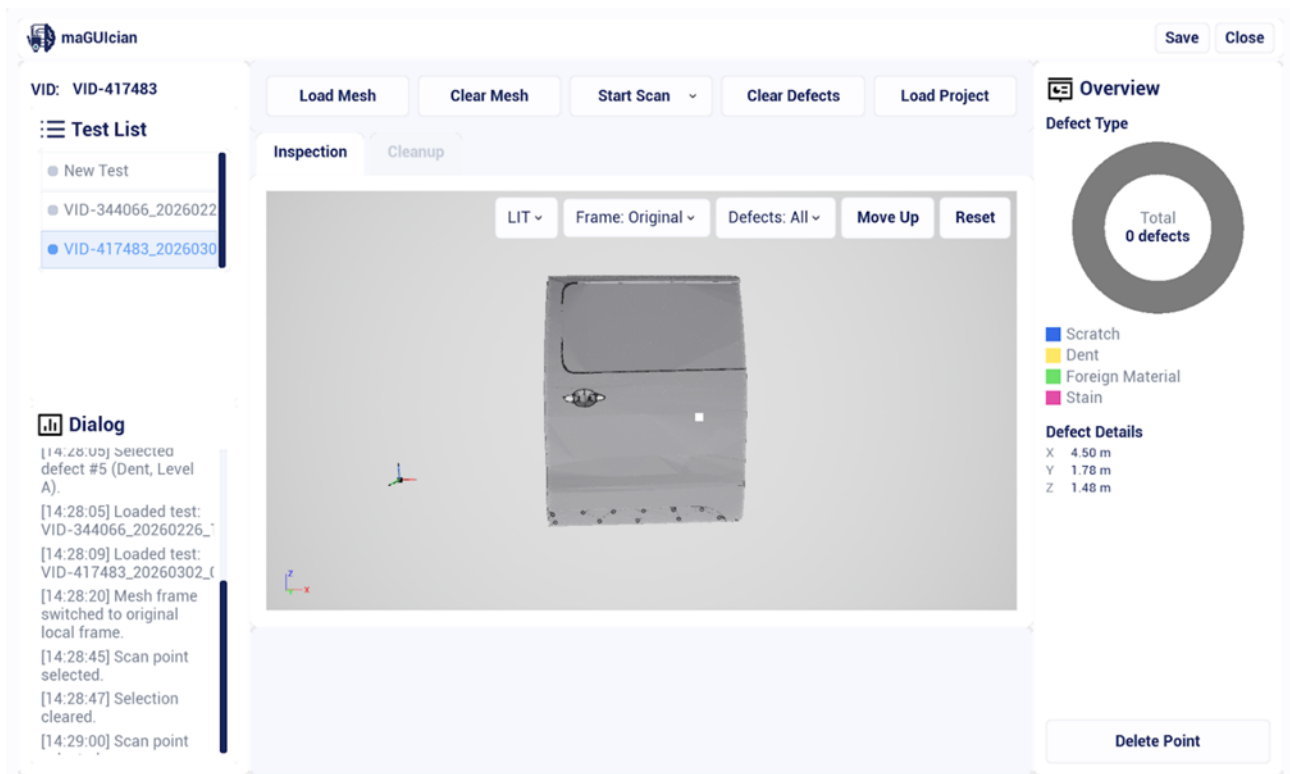


Figure 12: MaGULician main viewer GUI.

The Cleanup Manager section, shown in the figure below, can be accessed through the dedicated Cleanup button. The Overview panel summarizes the detected defects, reporting both the total number of defects and the number already resolved, as shown in Figure 13.

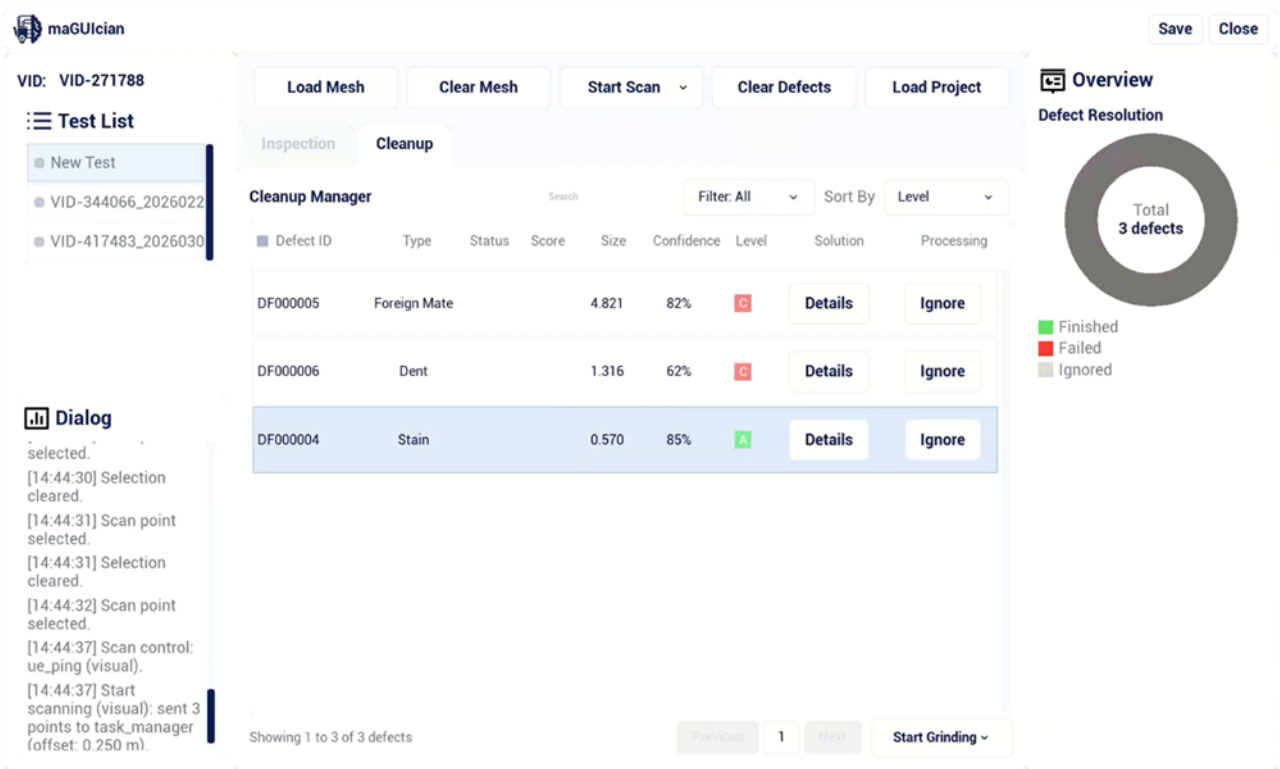


Figure 13: MaGULician Cleanup Manager.

6 PLANNING AND INTEGRATED FUNCTIONALITIES

6.1 TASK AND MOTION PLANNING

The MAGICIAN platform integrates task- and motion-planning components to coordinate the inspection and reworking operations of the Sensing Robot and the Cleaning Robot. These components generate efficient robot actions while considering the geometry of the workpiece, operational constraints, and the possible presence of human operators. The following subsections describe the planning approaches for defect detection, defect reworking, and human-aware motion planning.

6.1.1 DEFECT DETECTION PLANNING

6.1.1.1 VISUAL DEFECT DETECTION PLANNING

The sensing trajectories used to inspect the vehicle surface with the visual sensor mounted on the robot are planned and generated using ergodic control. This strategy enables efficient surface inspection by prioritizing regions where defects are more likely

to be found.

Given a target probability density function (PDF), a trajectory is ergodic if its temporal statistics match the spatial statistics of the target distribution. Specifically, In the ergodic control formulation proposed in [Mat 11] and denoting by $s(t)$ the system trajectory, the ergodic metric is defined as follows:

$$\mathcal{E}(s(t)) = \sum_{k=0}^K \Lambda_k [c_k(s(t)) - \phi_k]^2$$

Where K is the number of Fourier basis functions per dimension and $\Lambda(k) = \frac{1}{(1+|k|^2)^\omega} \cdot \phi_k$ and c_k are respectively the Fourier coefficients of the target distribution $\Phi(x)$ and of the distribution $\mathcal{C}(x)$ representing the fraction of time spent in each region of the search space and defined as:

$$\mathcal{C}(x) = \frac{1}{T} \int_0^T \delta[x - s(t)] dt,$$

Where T is the considered time horizon.

The planning of the ergodic trajectory is formulated as an optimal control problem in which the ergodic metric is minimized while also penalizing the control effort $u(t)$:

$$J(s(t)) = q\mathcal{E}(s(t)) + \int_0^T \frac{1}{2} u(\tau)^T R u(\tau) d\tau.$$

The proposed framework assumes spatial correlation among consecutively inspected surfaces. The objective is to guide the inspection process by prioritizing the most informative regions while still ensuring sufficient exploration coverage. One of the main strengths of ergodic control is its ability to balance exploitation and exploration. Unlike strategies that focus exclusively on hotspot regions, ergodic control guides the trajectory proportionally to the target distribution, ensuring both thorough coverage and attention to high-probability regions. The proposed sensing trajectory planning algorithm is based on an active sensing framework, goal is to guide the robot trajectory so as to maximize the amount of information acquired during the inspection process. In robotics, a common approach to active sensing is to formulate a closed-loop framework in which a belief state is maintained and updated over consecutive iterations to guide the exploration process, as introduced for example in [Mil 16]. The loop is closed by incorporating the measurements collected during trajectory execution to update the belief state and guide subsequent sensing actions.

In Figure 14, the diagram of the proposed active sensing framework is presented.

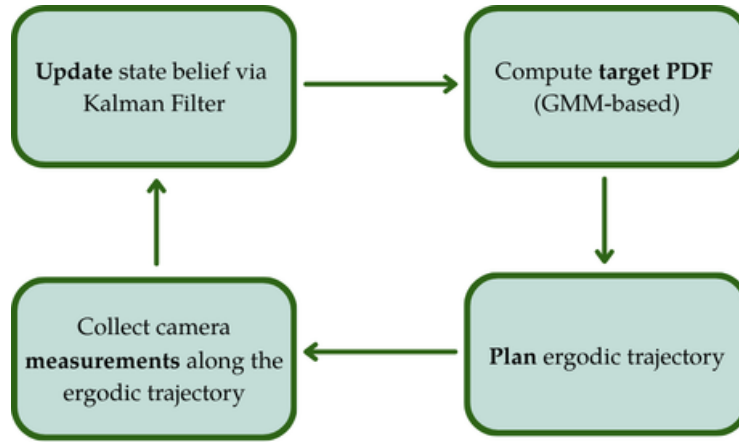


Figure 14: Diagram of the proposed active sensing framework.

In the proposed framework, the belief state is used to construct the target PDF employed for ergodic trajectory planning. Specifically, the inspection area is partitioned into N cells, and the belief state is modeled as an N -dimensional multivariate Gaussian distribution which for the l -th inspected vehicle is described as $N(\hat{x}_l, P_l)$. The i -th element of the state vector represents the estimated portion of the i -th cell affected by defects. The belief state is iteratively updated through a Kalman filter using the measurements acquired during the inspection process.

The target PDF used to guide the ergodic controller is modelled as a Gaussian Mixture Model (GMM), hereafter denoted as $\Phi(x)$. A key aspect of the proposed framework lies in the construction of the target PDF, which is designed to guide the trajectory toward regions where defects are more likely to be detected. Assuming spatial correlation of defects across consecutively inspected vehicles, regions in which defects were detected during previous inspections should be assigned higher priority. At the same time, it is essential to preserve sufficient exploration of areas that have been visited less frequently in the past. Consequently, the target PDF must jointly encode both exploitation and exploration objectives.

The GMM $\Phi(x)$ is modelled such that the Gaussian components have fixed means located at the centroids of the inspection cells and fixed covariance matrices. The information encoded in the belief state is incorporated into the PDF through the component weights. Specifically, for the $(l+1)$ -th inspected surface, the vector of weights is computed according to the following expression:

$$w_{l+1} = \beta P_{l+1}^{-1} x_{l+1} + (1 - \beta) P_{l+1} \mathbf{1}$$

in which $\mathbf{1}$ is the vector with all ones. The first term in the expression assigns higher values to cells with larger state estimates and lower uncertainty, corresponding to regions that have been visited in the past and where defects have been detected; this term therefore promotes exploitation. Conversely, the second term assigns higher values to cells with higher uncertainty, i.e., lower confidence in the state estimate, which corresponds to less frequently visited regions, thereby encouraging exploration. Tuning the parameter $\beta \in [0,1]$, it is possible to obtain a trade-off between exploitation and

exploration. The weights are further normalized and the GMM associated with the $(l + 1)$ -surface is obtained with the following formula:

$$\Phi(x) = \sum_{i=1}^n w_{l+1,i} N(x|\mu_i, \Sigma_i)$$

To prevent the sensing trajectory to visit areas outside of the mesh or holes in the surface, zero values are assigned weights of the Gaussian components corresponding to cells in these regions.

During trajectory execution, measurements of the inspected surface are acquired through the visual sensor mounted on the robot end-effector. The collected measurements are stored and forwarded to subsequent modules, such as the reworking module, and exploited as prior information to guide the exploration of subsequent vehicles. To this end, the acquired data are processed offline to construct the measurement vector employed by the Kalman filter for updating the belief state. The trajectory is computed by solving the nonlinear optimization problem using IPOPT, with the problem formulation implemented within the CasADi framework.

The vehicle mesh is mapped onto a two-dimensional plane through a one-to-one transformation. From the selected inspection area, the minimum bounding rectangle is computed and used as the domain for the ergodic control algorithm. This step is required because the adopted ergodic metric assumes a rectangular Euclidean domain. The 2D ergodic trajectory (Figure 15) obtained as result of the optimization problem is then mapped back onto the three-dimensional mesh space through the inverse transformation (Figure 16). The generated 3D trajectory is subsequently provided to the motion planner.

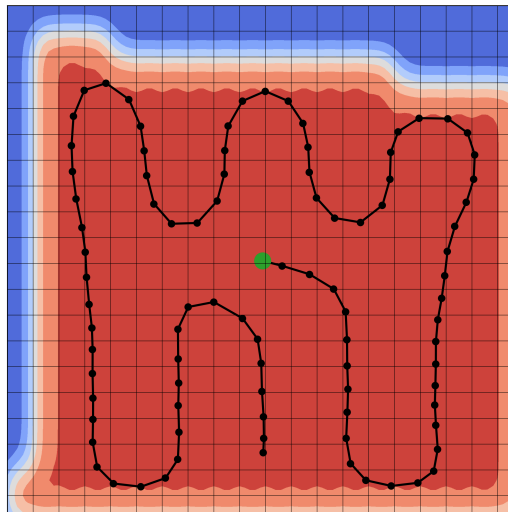


Figure 15: 2D ergodic trajectory generated for the inspection of the first surface in the sequence, considering a target PDF for ergodic control that approximates a uniform distribution.

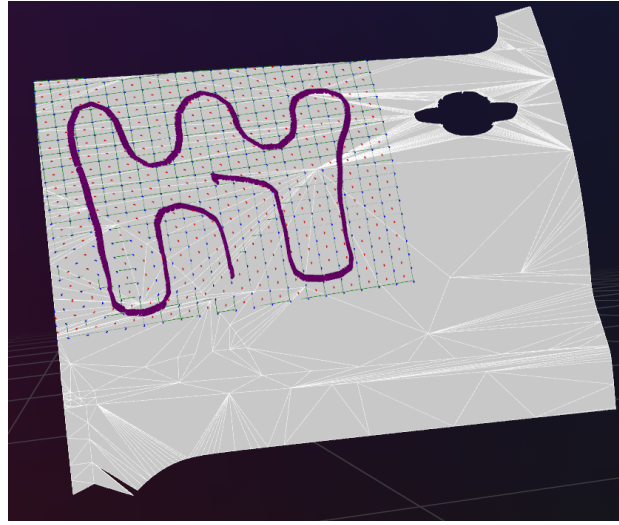


Figure 16: The corresponding trajectory mapped back onto the 3D coordinates of the mesh frame.

6.1.1.2 TACTILE DEFECT DETECTION PLANNING

The tactile inspection strategy proposed integrates two main components: a probabilistic-based belief estimator that provides the next possible defective locations on the surface, and an ergodic control problem that provides the next trajectory.

Ergodic control provides a natural trade-off between exploration and exploitation encoded in one single metric ε that measures the distance between the time-averaged statistics of a trajectory and a reference PDF that represents our current belief of where the defects are located. The objective of ergodic control is to being able to spend more time on more probable defective locations and less time on less probable locations but still visit those locations. The ergodic control problem in our pipeline is described as an optimization problem subject to some dynamics $\dot{x} = u$ related to the end-effector of the robot with inputs u and trajectories x , with weights γ and R , and a time horizon t_f .

$$\min \quad \gamma \varepsilon + \int_0^{t_f} u^T R u^T d\tau$$

This problem has been encoded using the CasADi library along with the IPOPT solver, and it has been parametrized such that in our framework it is possible to modify quantities of interest like the horizon time, maximum velocity, dimension of the search space, etc. The solution of this problem is the ergodic trajectory to be performed by the robot in the search space.

The execution of the trajectory on the robot relies on an admittance controller that ensures the contact of the tactile sensor with the surface by setting a reference constant force. So, whenever the sensor is in contact with a defect, there will be perturbations/spikes in the measurements and that information can be used to update our belief of defect locations.

After the execution of the ergodic trajectory on the surface and the collection of tactile data, there is a preprocessing step to separate “useful” (force on defective locations)

information, to find this kind of information, needed for this PDF estimator to work properly, it is intended to rely on the probability output of the classifier module to serve as a filter given a certain threshold.

The PDF estimator is the main component of this pipeline, and it works through the sense of iterations; one iteration is such that the robot has solved and executed an ergodic trajectory given a prior PDF and is ready to provide the next belief. The input for this component is the force-position data collected and pre-processed, and the output is a probability map of the next defective locations belief modelled as a Gaussian Mixture Model (GMM); however, in this framework, there is no need to know a priori the number of defects we are looking for, that is computed “on the fly” by evaluating the number of possible gaussian components through the “k-means” clustering algorithm and the “silhouettes” distance-based criterion.

Once the number of defects in the search space has been determined, the objective is to drive the GMM estimation to the real and unknown one, which ideally, not only provides the location of the defects, but also provides an idea of their shape (assuming they are gaussian-like). The algorithm finishes when all the defects have been “found” (meaning, that their gaussian estimation component has reached a certain threshold on the total spread related to the covariance matrix), or when a maximum number of iterations is reached, in such a case, maybe not all the defects have reached the termination condition but this framework is still able to provide the next estimation.

A few nice characteristics of this approach are the next: the removal of gaussian components whenever they reach the termination condition in order to stop spending time in already “found” defects and focus on the others; and the “double-check” behaviour that executes a second ergodic trajectory on a given iteration if there is no useful information to give the next update. The first one is especially useful if the defects have different levels of severities because of the forgetting factor that serves to avoid exploiting information where it is no longer needed and don’t lose track of the low severity level defects. The second one is a nice feature to have when the coverage of the trajectory is poor and/or the planning time is limited.

6.1.2 DEFECT REWORKING PLANNING

6.1.2.1 DEFECT SELECTION AND SCHEDULING

Given the list of potential defects from the Sensing Robot, the defect selection and scheduling task can be formulated as a graph optimisation problem. 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 and the total reward is maximised.

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. A complete mathematical formulation of the orienteering problem can be found 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 have to be derived by three different modules, as visualized in Figure 17.

To create the travel times matrix (T_{xy}), as visualized in Figure 17 as the “Distance Matrix Calculator”, we need to follow these detailed steps: 1. Collect Defect Locations, 2. Calculate Euclidean Distance. The details are reported in D4.2.

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 17, 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).

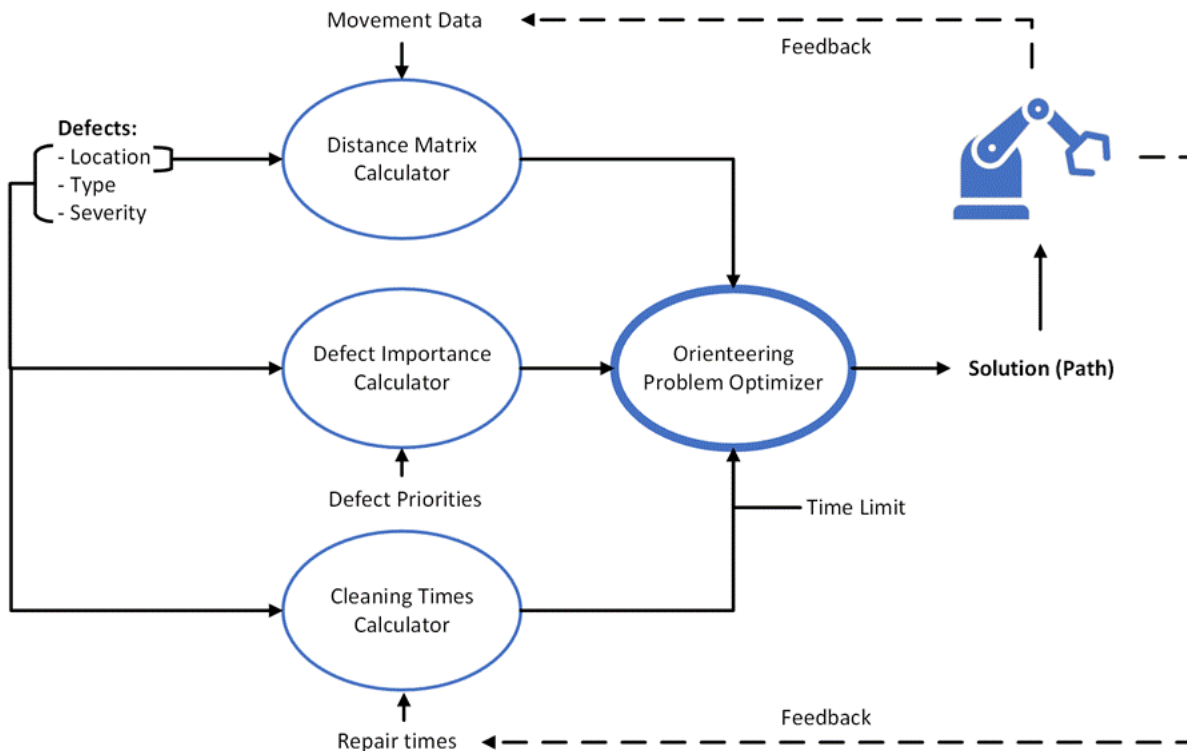


Figure 17: Path Optimizer Framework.

6.1.2.2 TASK PLANNER IMPLEMENTATION

The algorithm has been implemented and integrates within the MAGICIAN framework and is currently undergoing testing and benchmarking. Consideration is also being given to a stochastic generalisation of the problem to account for false positives and to multi-agent extensions.

Given the critical importance of rapid computation in real-time applications, particularly in robotic systems, exploring methods that effectively balance speed and solution quality is essential.

Whilst exact optimisation methods, such as those from Gurobi and SCIP, offer precision for Integer Linear Programming (ILP), their high computational demands render them impractical for real-time scenarios where solutions are required within seconds.

To address MAGICIAN's real-time needs, heuristic and meta-heuristic methods have been adopted. These approaches, including nearest neighbour, greedy algorithms, local search, and learning-based approach, prioritise speed and simplicity, generating sufficiently good approximate solutions more rapidly than exact methods.

Depending on the problem size, the module selects an appropriate algorithm: a Branch-and-Bound approach for instances with up to 7 defects, a greedy heuristic known as Profit Ratio Search for larger cases, and a learning-based meta-heuristic for extension of other potential use cases. For a detailed description of the OP, please refer to D4.2, which

includes a mathematical notation of the problem.

- **Branch-and-Bound (B&B):** A systematic algorithm for finding optimal solutions to optimization problems, including combinatorial and integer programming. In the process of solving the OP it considers the following steps:
 - Initialization: Starting with the initial problem and setting up the best-known solution and a list of subproblems.
 - Branching: Dividing the problem into smaller subproblems by making decisions on variables.
 - Pruning: Eliminating subproblems that cannot yield a better solution than the current best.
 - Selection: Choosing the next subproblem to explore.
 - Iteration: Repeating the above steps until all subproblems are explored or a stopping criterion is met.

While powerful, B&B can be computationally intensive for large instances and is often used with heuristics to improve efficiency.

- **Profit Ratio Search (PRS):** A simple greedy meta-heuristic that is used for solving optimization problems, such as the Orienteering Problem. It operates by starting from an initial node and repeatedly selecting the node with the best profit ratio to add to the path. This process continues until all nodes have been visited, or when the time limit has been reached. In the process of solving the OP it considers the following steps:
 - Initialization: Start with the initial node, and initialize the path, profit, and total time
 - Calculate Profit Ratio: Compute the profit ratio for each node. That is the profit earned at a node divided by the travel time to that node plus time spent at that node.
 - Select Next Node: Choose the node with the highest profit ratio.
 - Update Solution: Update the path, profit, and total time.
 - 2OPT Optimization: Use the 2OPT algorithm to optimize the path by reducing travel time.
 - Iteration: Continue until all nodes are visited or the time limit is reached.

While easy to implement and fast, PRS does not guarantee an optimal solution, as it can sometimes lead to suboptimal routes due to its greedy nature.

- **DEcoupled discrete-Continuous optimization with Service-time-guided Trajectory (DeCoST):** A reinforcement-learning based meta-heuristic that is used for solving OP with variable profits and time windows (OPTWVP), which is a more generalized problem that can be applied to other use cases. DeCoST applies a two-stage search which effectively decouple the searching process while enabling efficient and learnable coordination. The reinforcement learning agent maximizes the overall reward and learns the trade-off between travel time and service time allocation and better guide the policy of the initial service time assignment. In the process of solving OP, it considers:
 - Feasible Solution Generation: Generate M feasible defect removal

- sequence solutions with the node trajectory and an initial service time.
- Service time optimization: Optimize the service time using a service time optimization algorithm.
- Decision: return the best solution.

The solution outlined above has proved practical for our use and is tested in the prototype. In addition, following the DeCoST methodology, we plan to design a human-aware Task and Motion Planner (TAMP) for defect reworking planning. This planner will integrate task-level decisions, such as defect selection and scheduling, with motion-level feasibility and human-related constraints, enabling safer and more efficient human-robot collaboration during the reworking process.

6.1.3 HUMAN-AWARE MOTION PLANNING

The new implementation of the motion-planning module follows the roadmap-manipulation approach described in [Hu 22] but implements it directly in a ROS2 package through a parallelized time-variant A* planner. The planner operates on an orientation-augmented roadmap in which each search state includes the roadmap node, end-effector orientation, inverse-kinematic solution, and arrival-time layer. Edge feasibility is evaluated lazily during graph expansion using the current human pose and, in proactive mode, the predicted human trajectory. Similarly to [Hu 22], temporal information is embedded in the search process: the traversal time of each edge depends on when it is executed and on the predicted robot-human clearance. Motions close to the human are slowed down or rejected if they violate the stop or collision distance, while safer transitions can be executed at higher velocity. Successor expansion is parallelized in deterministic batches: multiple worker threads evaluate candidate transitions and collision checks independently after which the results are merged in a fixed order before updating the global A* queue. This preserves repeatability while reducing the average planning latency from the previous value of about 0.25 s to approximately one camera frame, around 0.04 s, almost matching the rate at which new human poses are published.

Plan generation and replanning are handled by a receding-horizon supervision logic that combines predictive replanning ideas from [Ole 24] with a lazy roadmap strategy related to [Jai 04], while avoiding the overly conservative behaviour typical of purely reactive methods such as [Cef 19; Ot 16]. At each 33 ms planning cycle, the ROS2 node builds a consistent snapshot of the robot, roadmap, static environment, current human state, and human-motion predictions. The plan manager maintains at most one asynchronous backup planning process from the latest useful data, discarding candidates that arrive too late or are based on outdated human observations. During nominal execution, newer backup plans can replace the active trajectory when they improve the remaining execution time or refresh the motion according to updated human predictions. If a future violation is predicted with sufficient margin, the system schedules a temporally consistent replacement; if the violation is imminent or the stop distance is already reached, the robot enters collision-avoidance mode and can generate

an emergency reactive plan from the stopped state. This architecture enables faster and more reliable replanning, reduces sensitivity to spurious predictions, and synchronizes planning with the perception frame rate, allowing the robot to adapt both path and velocity continuously while avoiding unnecessary stops.

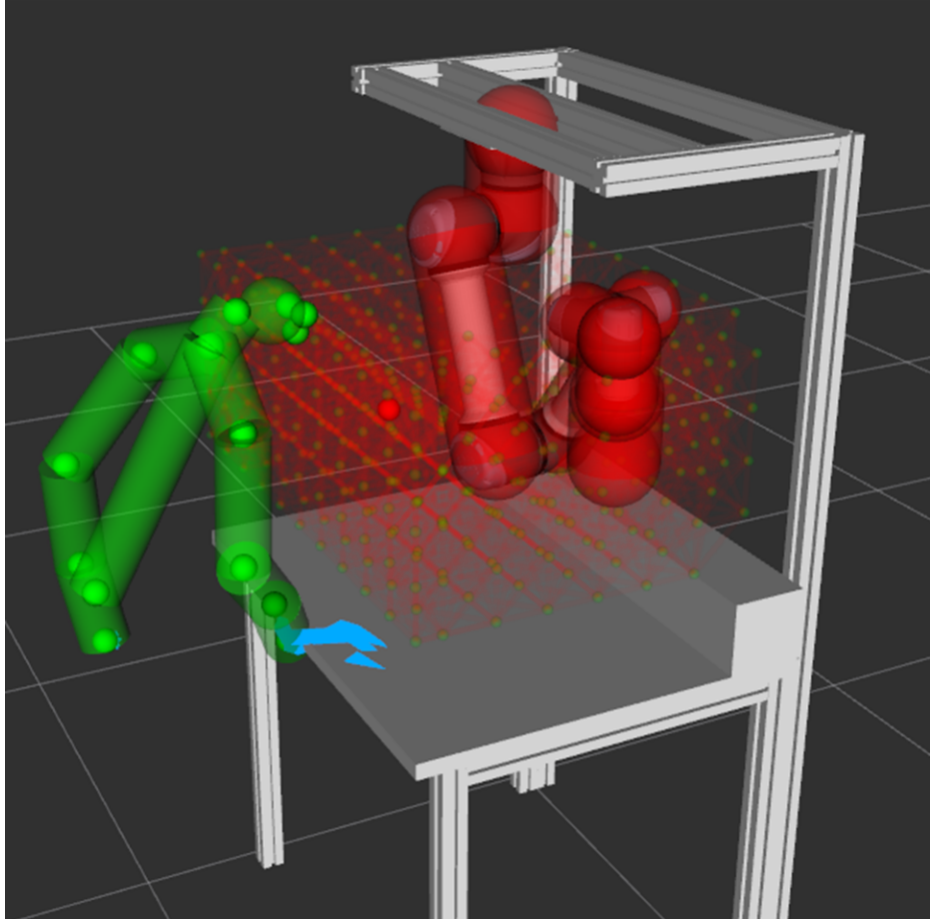


Figure 18: Representation of the detected human being (in green) and of the robot (in red), with the graph nodes depicted as 3D red points in space.

The planner was evaluated in a deliberately demanding simulated work cell, which is illustrated test-setup image Figure 18. The setup uses a ceiling-mounted UR5e robot operating above a table, where the robot repeatedly executes simulated pick-and-place tasks across the tabletop. At the same time, a simulated human moves inside the robot workspace; the human motion trajectories are taken from the HA4M dataset, providing realistic motion patterns for testing prediction-based planning and replanning. This scenario is intentionally stricter than the expected MAGICIAN project setup, where the planner will be adapted to the Doosan robot. In the present evaluation, the human can move through the same constrained workspace used by the robot for manipulation, making collision avoidance, timing adaptation, and stop/replan decisions more challenging than in the target deployment.

The evaluation was carried out on 195 successful tasks per method. The new planner keeps the computation time close to the perception update rate: the human-aware methods required only about 0.04-0.07 s on average for planning, compared with approximately 0.25 s in the previous implementation. The no-human baseline completed tasks in 9.33 s on average, while human-aware execution increased the duration, as expected, because the robot adapts its motion around the human. Among the human-aware strategies, the reactive method required 18.19 s on average and produced about 4.63 stops per task. The predictive variants reduced this value to about 3.31-3.48 stops per task, with the best mean execution time obtained by the 2 s prediction strategy at 16.65 s. All human-aware methods adapted the path to the human in every evaluated task and reported zero collisions, while maintaining a mean robot-human distance of about 0.19-0.20 m. Overall, the results show that the new implementation preserves safety while reducing unnecessary stops and bringing replanning latency down to the camera-frame timescale.

6.2 INTEGRATED WORKFLOWS

6.2.1 VISUAL DEFECT DETECTION

The visual defect detection module has been integrated into the ROS2 framework to operate with the Doosan robot and the visual sensor mounted on the robot end-effector. The trajectory executed by the robot to inspect the surface and detect defects is planned to use the ergonomic control approach introduced in Section 6.1.1.1. The proposed algorithm guides the robot toward regions of the surface where defects are more likely to occur. The approach assumes that defects are spatially correlated across consecutively inspected vehicles. By exploiting this information, measurements acquired from defects detected on one vehicle can be used as prior information to guide the exploration of subsequent vehicles. For the first vehicle in the inspection sequence, no prior information is assumed to be available; therefore, the surface is explored uniformly, as shown in the example of the plot in Figure 15.

During trajectory execution, measurements are acquired using the visual sensor. The sensor grabber captures images at a frequency of 20 Hz, and for each acquired image, the visual classifier publishes ROS2 messages containing defect detection information. Specifically, the classifier provides, for each detected defect in the image, its coordinates in the image frame, defect severity, defect class, and detection confidence. Each published ROS2 message includes a header containing the timestamp corresponding to the image acquisition time.

The messages generated by the visual sensor are processed offline and synchronized with the robot data describing the end-effector pose using the corresponding timestamps. This synchronization enables the reconstruction of the 3D coordinates of the detected defects in the mesh-link reference frame. The Figure 19 shows the ergonomic trajectory projected onto the mesh frame (purple spheres), together with the defects detected by the visual sensor during trajectory execution (yellow spheres). In the current

version of the vision classifier, a significant number of false positives are observed. The resulting 3D defect coordinates are then published as ROS2 messages and forwarded to the subsequent modules of the processing pipeline.

In addition, the detected defects are processed to construct the measurement vector used to update the belief state through a Kalman filter. The updated belief state is then employed as prior information to guide the exploration of corresponding surface regions on subsequent vehicles. The Figure 20 shows the updated probability density function (PDF) and the corresponding ergodic trajectory over the 2D rectangular search space. In this example, the parameter β used in the computation of the Gaussian mixture model (GMM) weights is selected to promote an exploitative behaviour, encouraging revisits to regions where defects were identified in previous iterations.

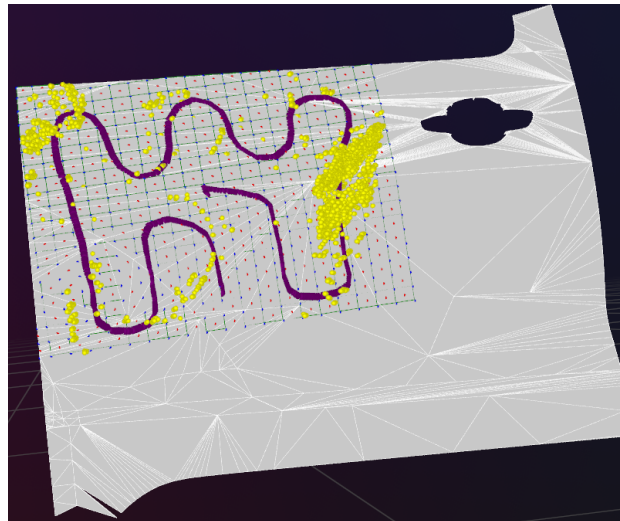


Figure 19: Ergodic trajectory projected onto the mesh frame (purple spheres), together with the defects detected by the visual sensor during trajectory execution (yellow spheres).

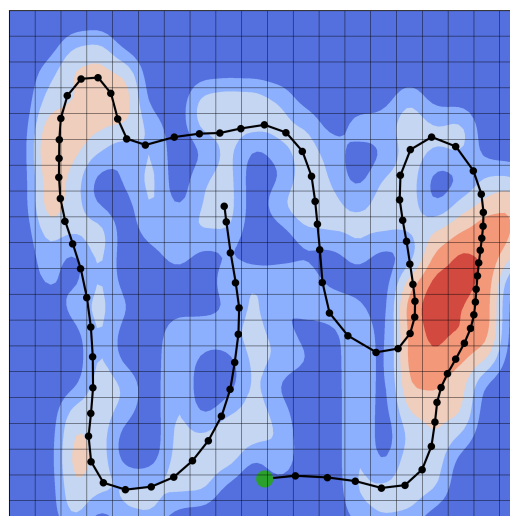


Figure 20: The updated probability density function (PDF) and the corresponding ergodic trajectory over the 2D rectangular search space.

6.2.2 TACTILE DEFECT DETECTION

The tactile defect detection module has been integrated into the ROS2 framework to work on the Doosan robot using the pipeline solution described in Section 6.1.1.2. This module computes the ergodic trajectory to be executed based on a prior PDF of a 2D surface, however, in order to work on the real car body, the module relies on a projection module that maps a portion of the target mesh to a 2D domain for the trajectory planning and then transform it to the 3D scenario for the execution.

For the ergodic planner, 3 different time horizons are being considered, namely, 10, 15 and 20 seconds. This consideration has a direct effect on the desired level of coverage of the consider area. For ALT integration, the time horizon considered was 10 seconds for a faster computation in testing.

After real execution, the force-position data recorded along the ergodic trajectory is then filtered by the probability of having a defect given by the classifier module. If there is useful data (defective related force), then a computation of the next PDF and a new ergodic trajectory can be performed. Figure 21 shows the resulting test perform in ALT over a portion of a door.

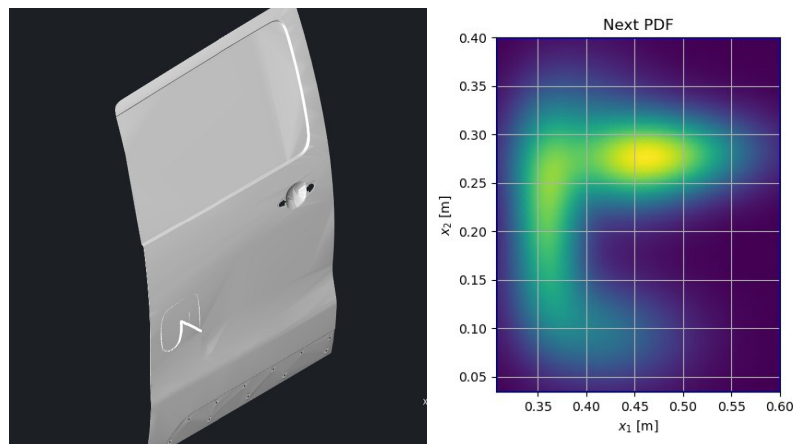


Figure 21: Ergodic trajectory executed on a door, and next PDF computed with force-position data.

In the case when there is no useful data, the algorithm computes a “double-check” ergodic exploration round to be performed. This is especially useful when the level of coverage is not enough.

The tactile defect detection module accuracy depends on the geometric uncertainty of the sensor footprint and the idea of the defect spread to be found (if available). Then the result is a set of gaussians components that shows where the defects are located in the 2D projection.

6.2.3 COMBINED VISUAL-TACTILE INSPECTION

The two sensing modalities developed in the MAGICIAN project, tactile and visual, have different computational costs and performance, depending on the type of surface that

the robot inspects, its transparency, and its position within the workspace. Sometimes it can be convenient to use visual inspections, sometimes tactile inspection, sometimes the combination of the two modalities.

Therefore, we have developed a sensor selection policy which will complement the ergodic search. Specifically. The policy outlined below decides the sensing action to take for every patch explored by the ergodic trajectory.

6.2.3.1 PROBLEM DEFINITION AND SPATIAL MODELLING

The environment is discretised into N distinct pixels, where each pixel i is associated with a latent state $m_i \in \{0,1\}$, with $m_i = 1$ indicating the presence of a defect.

The robot operates under a strict total time budget $B > 0$, which is found subtracting from takt time T , the time needed by the robot to travel between the different locations. At each sequential decision step t , the robot travels across a sensing location x_t (which defines a localised patch of pixels W_t and a specific sensing mode $s_t \in \{V, T, VT, \emptyset\}$, representing visual, tactile, joint visual-tactile sensing and null action. Executing a chosen action incurs a location- and mode-dependent time cost $\tau_{s_t}(x_t) > 0$.

The spatial correlation and underlying distribution of the defects are modelled via a continuous Gaussian Mixture Model (GMM) source prior. The parameter set θ describes the spatial coordinates, spreads, and mixing weights of the underlying defect-generating sources.

Let $\mathcal{X} = \bigcup_{t \in [0, T]} x_t$ be the set of locations covered by the ergodic search trajectory for a given iteration. Let $IG_s(x_t)$, represent the quantity of information gained by using the (possibly empty) sensing action s on location x at time t .

The problem can be modelled as a variant of the classic knapsack problem: select for each point an action s_t , such that the sum of the information gained is maximised subject to the sum of all the time spent (quantified by $\tau_{s_t}(x_t)$) remains below the budget B .

As detailed next, the key problem is how to quantify the information gain.

6.2.3.2 THE NECESSITY OF PARTICLE FILTERING FOR PARAMETER ESTIMATION

To execute this adaptive policy, the robot must maintain a running posterior probability distribution over the GMM parameters θ given the accumulated data $\mathcal{D}^{(h-1)}$ accumulated up to the processed item number $h - 1$. A Sequential Monte Carlo method, specifically a Particle Filter, is utilised rather than classical analytical filters (such as the standard Kalman Filter) due to three key factors:

- **Highly Non-Linear and Non-Gaussian Likelihoods:** The mathematical relationship mapping the continuous GMM source parameters θ to the discrete, noisy, patch-based binary observations Z is highly non-linear.

- Analytical Intractability: Because the sensing modalities (Visual, Tactile, and Joint VT) introduce distinct, complex likelihood models, calculating the exact mathematical posterior distribution $p(\theta|\mathcal{D})$ is impossible to solve in closed form.
- Flexible Non-Parametric Representation: A particle filter circumvents analytical constraints by approximating the continuous probability distribution using a cloud of J discrete, weighted samples ("particles") $\{\theta^{(h)}, w^{(h)}\}$. Each particle embodies a concrete hypothesis of the environment's true layout, allowing the filter to track multiple competing spatial hypotheses simultaneously.

6.2.3.3 OVERCOMING STATIC PARAMETER DEGENERACY VIA THE LIU–WEST FILTER

While standard particle filters excel at tracking dynamic, time-evolving states, they encounter a severe theoretical bottleneck when applied to static parameters like θ . Because the true defect configuration is fixed and does not change as the robot moves, there is no physical process model to naturally transition particles forward in time.

In a standard bootstrap particle filter, the resampling step duplicates highly probable particles and discards improbable ones. Without a mechanism to introduce variation, this process rapidly strips diversity from the system. Within a few iterations, the entire cloud collapses into identical carbon copies of a single sample, a phenomenon known as particle degeneracy. The filter becomes falsely overconfident, ceases to explore the parameter space, and can no longer learn from new sensor data.

To resolve this, the framework implements a Liu–West (LW) particle filter. The LW filter introduces a specialized kernel-smoothing mechanism during the resampling phase that artificially injects diversity while strictly preserving the statistical properties of the distribution. Rather than adding arbitrary random noise—which would cause the total variance of the particle cloud to artificially bloat and drift over time—the LW filter employs a balanced Shrinkage and Perturbation strategy:

- Shrinkage: Prior to perturbation, each particle's parameter vector θ_j is drawn slightly closer to the current mean ($\bar{\theta}$) of the entire particle cloud using a multiplier $a \in (0,1)$:

$$\widetilde{\theta}_j = a\theta_j + (1 - a)\bar{\theta}$$

This effectively compresses the cloud inward, reducing its overall variance.

- Perturbation: The filter then adds a carefully scaled component of random Gaussian noise to this shrunken position.

By mathematically balancing the shrinkage factor against the variance of the injected noise, the Liu–West framework guarantees that the total variance of the particle cloud remains perfectly preserved. This ensures degeneracy is avoided, particle diversity is sustained, and the filter maintains a true, uncorrupted representation of uncertainty.

6.2.3.4 WEIGHT-ENTROPY INFORMATION GAIN AND COMPUTATIONAL OPTIMISATIONS

Under the particle filter representation, the uncertainty of the robot's belief state can be elegantly characterised by the entropy of the discrete particle weight distribution:

$$H_w(\mathcal{D}_{t-1}) := - \sum_{j=1}^J w_j \log w_j$$

with J being the number of particles.

When evaluating a prospective action, the Weight-Entropy Information Gain IG_w measures the expected drop in this entropy conditional upon receiving a hypothetical sensor observation Z :

$$IG_w(a) := \mathbb{E}_{Z \sim p(Z|a, \mathbb{D}_{t-1})} [H_w(\mathcal{D}_{h-1}) - H_w(\mathcal{D}_{h-1} \cup \{(x, s, Z)\})]$$

This metric explicitly quantifies how effectively a candidate action will focus the posterior probability mass onto a smaller subset of correct parameter hypotheses.

6.2.3.5 IMPLEMENTATION DETAILS FOR REDUCING SAMPLE AND CANDIDATE EVALUATION LOAD

Computing an exact expectation across all possible measurement outcomes Z is computationally prohibitive. To achieve operational efficiency, the implementation introduces two vital algorithmic steps to reduce both sample generation and candidate action evaluation:

- **Single-Observation Particle Approximation:** Rather than simulating a massive array of prospective measurements to calculate the expectation, a Monte Carlo approximation is used where exactly one hypothetical observation $Z^{(h)} \sim p(Z|x, s, \theta^{\{(h)\}})$ is drawn per particle. The resulting entropy drops are then averaged. This bounds the evaluation cost per action strictly to $\mathcal{O}(J)$, where J is the number of particles.
- **Candidate Action Shortlisting (Pre-filtering):** Evaluating IG_w even with this sample reduction remains expensive when conducted across an entire workspace grid with multiple sensing modes. To bypass this bottleneck, the system computes a rapid pre-filter metric: the variance of the local detection probability across the particles, $\text{Var}_\theta[q_s(x; \theta)]$, and then:
 - If $\text{Var}_\theta[q_s(x; \theta)] = 0$, all particles are in perfect agreement regarding the state of that region, meaning the prospective information gain is guaranteed to be zero.
 - For non-zero values, the expected information gain scales linearly with this local variance.

By evaluating this local variance first, the robot can instantaneously discard

uninformative areas of the workspace, shortlisting a tiny fraction of highly uncertain candidate actions for full weight-entropy evaluation. This dual-layer optimisation drastically reduces the computational footprint, making real-time, active Bayesian sensor fusion fully tractable.

6.2.4 DEFECT REMOVAL

Following the Task Planner implementation described above, the defect removal process acts as the execution layer of the overall defect reworking pipeline. The Sensing Robot first provides a defect list, including the detected defect information and corresponding locations. These defects are visualised in the GUI, allowing the operator to inspect the detected defects and monitor the planned removal process.

Based on the defect list provided by the Sensing Robot, the Task Planner computes an optimised defect removal sequence. This sequence determines the order in which the defects should be handled, considering the planning criteria introduced in the previous section, such as travel cost, cleaning time, defect priority, and time constraints.

Once the optimal or near-optimal defect removal sequence has been generated, the planned tasks are transmitted to the Defect Removal module through ROS2 message communication. For each selected defect, the system sends the corresponding event information together with its sequence number. The event defines the defect removal action to be triggered, while the sequence number indicates the execution order within the planned removal route.

After receiving the ROS2 message, the Defect Removal module executes the removal operation according to the assigned sequence.

In this way, the planning result generated by the Task Planner is converted into executable robotic actions, enabling the system to perform defect removal in a structured, traceable, and coordinated manner.

6.3 INTERACTION WITH WELDING INFORMATION

Data generated during the welding process can be integrated into the proposed ergonomic control framework for visual defect detection, providing prior information to guide the sensing robot toward regions where welding spatters are more likely to be present. In particular, the welding software can acquire information regarding the severity of the spatters generated during welding operations. The spatter severity is represented by a discrete value ranging from 0 to 3, where 0 indicates the absence of spatters and 3 corresponds to the highest spatter severity. Only external spatters located on the vehicle surface are of importance for the project. However, based on the information available from the welding software, it is not possible to distinguish between internal and external spatters. Consequently, all welding spots for which the welding software reports the presence of spatters are considered as potentially associated with surface defects and are therefore relevant for inspection.

The welding software transmits the spatter severity information to the PLC, which subsequently publishes ROS messages associating each welding spot on the vehicle with the corresponding spatter severity value. In addition, a lookup table will be available to associate each welding spot with its corresponding 3D coordinates in the mesh reference frame.

For the first vehicle in the inspection sequence, the prior information provided by the welding process is used to initialize the state estimate defining the belief state introduced in Section 6.1.1.1. For subsequently inspected vehicles, the welding-process prior information is incorporated during the prediction step of the Kalman filter. As a result, when constructing the Gaussian Mixture Model (GMM) used to guide the ergodic exploration, as described in Section 6.1.1.1, higher probability values are assigned also to regions corresponding to welding spots where the welding process identified potential spatters. Consequently, the generated ergodic trajectory will tend to prioritize the inspection of these regions as well.

7 SYSTEM OPERATION AND NEXT STEPS

At the time of writing, further work is required to complete the validation of the integrated solution. The main open issues concern the end-effector-specific calibration procedures, the experimental validation of the vibration-damping interface during grinding, the completion of combined visual-tactile inspection tests, the refinement of the defect-removal workflow, and the integration of welding information into the inspection-planning process. Further testing at the ALT demonstrator will also be used to assess the operation of the system under representative work-cell conditions.

8 BIBLIOGRAPHY

- [Hu 22] "T-PRM: Temporal Probabilistic Roadmap for Path Planning in Dynamic Environments." In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 10320–10327. DOI: 10.1109/IROS47612.2022.9981739 (with Luca Bartolomei, Ruben Mascaro and Margarita Chli.)
- [Jai 04] "A PRM-based motion planner for dynamically changing environments." In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vol. 2, pp. 1606–1611. DOI: 10.1109/IROS.2004.1389625 (with Thierry Siméon.)
- [Cef 19] "A general framework for task-constrained motion planning with moving obstacles." In: Robotica, Vol. 37, No. 3, pp. 575–598. DOI: 10.1017/S0263574718001182 (with Giuseppe Oriolo.)
- [Ole 24] "Scenario-based model predictive control with probabilistic human predictions for human–robot coexistence." In: Control Engineering Practice, Vol. 142, Article 105769. Elsevier (with Sergey Soltan, Zarema Balgabekova, Alberto Bemporad and Matteo Rubagotti.)

[Ot 16] "RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning." In: The International Journal of Robotics Research, Vol. 35, No. 7, pp. 797–822. SAGE Publications (with Emilio Frazzoli.)

[Mat 11] I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," Physica D: Nonlinear Phenomena, vol. 240, no. 4, pp. 432–442, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S016727891000285X> (with I. Mezić).

[Mil 16] "Ergodic exploration of distributed information," IEEE Transactions on Robotics, vol. 32, no. 1, pp. 36–52, 2016 (with Y. Silverman, M. A. Maclver, and T. D. Murphey)