MULTI-ROBOT NAVIGATION AND CONTROL FOR ACOUSTIC INSPECTION OF METAL PLATE STRUCTURES

A Thesis
Presented to
The Academic Faculty

By

Brandon Alves

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Computer Science
Department of Interactive Computing

Georgia Institute of Technology

July 2023

MULTI-ROBOT NAVIGATION AND CONTROL FOR ACOUSTIC INSPECTION OF METAL PLATE STRUCTURES

Thesis committee:

Dr. Cedric Pradalier
Department of Interactive Computing
Georgia Institute of Technology

Dr. Seth Hutchinson
Department of Interactive Computing
Georgia Institute of Technology

Dr. Harish Ravichandar
Department of Interactive Computing
Georgia Institute of Technology

Dr. Sehoon Ha
Department of Interactive Computing
Georgia Institute of Technology

Date approved: July 20, 2023

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to all those who have contributed to the realization of this project and the preparation of this report.

I would first like to thank my supervisors, Cédric Pradalier and Olivier Simonin, for their valuable advice and expertise in the field of robotics and multi-robot navigation. Their guidance was essential to the success of this project. I would also like to thank the members of my thesis committee for their help in preparation of this work – Cédric Pradalier, Seth Hutchinson, Harish Ravichandar and Sehoon Ha. I would like to thank the research teams of the INSA Lyon CITI-INRIA and CNRS IRL2958 GT-CNRS laboratories for their collaboration and knowledge sharing.

I would additionally like to express my gratitude to my loved ones for their unconditional support and understanding throughout this intense period. Their presence and their encouragement were an essential source of motivation. Finally, I would like to thank all the people who participated directly or indirectly in the realization of this project and the drafting of this report. Their support and commitment were key to the success of this study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

IMU Inertial Measurement Unit

LIDAR Light Detection and Ranging

mTSP multiple Traveling Salesman Problem

NP Non-deterministic Polynomial time

ROS Robot Operating System

SCC Strongly Connected Components

TSP Traveling Salesman Problem

UGW Ultrasonic Guided Wave

UWB Ultra-Wideband

SUMMARY

This report presents a study on autonomous exploration on metal structures, focusing on the evaluation of three exploration strategies. The objective of this study was to develop effective methods allowing autonomous robot systems to explore a metal structure, in a complete and efficient way, in search of corrosion points. The three strategies evaluated include *Roller Painting*, *Nordic Skiing* and *Polygonal Investigation*, all three based on occupancy grids. The experiments were carried out in simulation using Gazebo and a crawler model developed for the BugWright2 European project. These robots are notably equipped with Ultrasonic Guided Wave (UGW) sensors, specific to our problem. The performances of the different strategies were evaluated in terms of investigation time and accuracy of the mapping obtained. The results obtained demonstrated the effectiveness of each strategy. The *Roller Painting* strategy allowed for a quick but imprecise investigation. The *Nordic Skiing* strategy allowed a slow but rather precise investigation. Finally, the *Polygonal Investigation* strategy made it possible to combine the advantages of the other two strategies. Future perspectives include improving the polygonal exploration strategy by developing more robust methods for collision management. In addition, the extension of this study to experiments with several teams of robots constitutes an interesting avenue for further accelerating the investigation time. This study contributes to research in autonomous investigation and provides indications for the development of effective investigation systems in corroded metallic environments. The results obtained have important implications in various fields, such as service robotics, space exploration and environmental monitoring.

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Introduction

This study is part of the broader context of the BugWright2 European project, which aims to solve the problem of autonomous inspection and maintenance of large metal structures with heterogeneous fleets of mobile robots. In this project, we focus on the development of navigation strategies for a set of mobile robots using UGWs, or Lamb waves, to perform the inspection of metal plates. Indeed, guided waves have the particularity of propagating along a plate by interacting with the material that composes it, and by being affected by changes in geometry related, in particular, to corrosion.

The main problem is therefore to define multi-robot navigation strategies to optimize the acquisition of data allowing to perform a tomography of metallic surfaces. To achieve this objective, we will first carry out a bibliographical search, then set up navigation methods in a simulation environment. Finally, we will consider deployment on different robots depending on the results obtained. This project will be carried out under the supervision of Olivier Simonin (INSA Lyon CITI lab) and Cédric Pradalier (CNRS IRL2958 GT).

The expected contributions of this project are as follows:

- Development of multi-robot navigation strategies for the acoustic inspection of metallic structures.
- Optimization of data acquisition for performing tomography.
- Resolution of coordination and synchronization issues between robots.
- Implementation of navigation methods in a simulation environment.

This report presents the work carried out as part of my master's thesis on navigation and multi-robot control for the acoustic inspection of metal structures. In the first section, we introduce the subject of the report, present the objectives of our project and describe the state of the art. In the second section, we present the methodology used to carry out this project. In the third section, we present the results obtained. In the fourth section, we discuss the results obtained and the limitations of our work. Finally, in the last section, we conclude on the work carried out and present the perspectives for future work.

1.2 Preliminary Definitions

Here, we will explain the preliminary assumptions and definitions that will be used in the remainder of this report. First, we consider a planar environment, bounded and of known size. We are not interested in the location of the robots in the environment, but we assume that each robot is able to know its position in the environment. We also assume that the obstacles are localized in the environment. Only the corrosion areas are not located.

We use *crawler* robots. These robots are equipped with two drive wheels and an idler wheel. An example crawler is shown in Figure 1.1. The pose of the robot is defined by a triple (x, y, θ) where x and y are the coordinates of the robot in the environment and θ is the orientation of the robot in the environment. We assume that the pose of the robot is known. This assumption is justified by the fact that localization is not the subject of our project and that in the context of the BugWright2 project, localization has already been studied for these robots, as explained in section 1.3 and more precisely in [1]. We also assume that the robots are able to synchronize in order to be able to move simultaneously or alternatively. Again, this assumption is justified by the fact that synchronization is not the subject of our project and we found in the literature that this problem has already been studied. We note cr the unit cost of rotation of the robot and ct the unit cost of translation of the robot.

Each robot is either a transmitter or a receiver, or both. Crawlers are equipped with different sensors. Among them :

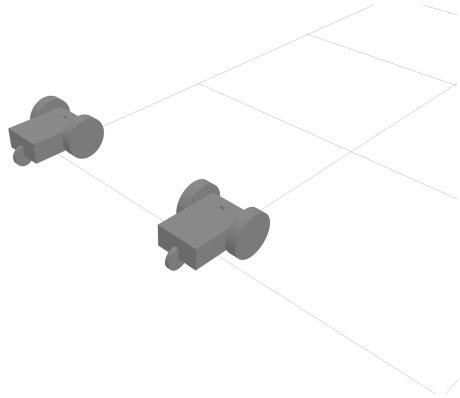


Figure 1.1: Crawler model used for acoustic inspection of metal structures.

- an Inertial Measurement Unit (IMU) sensor
- a piezoelectric or UGW sensor
- a Light Detection and Ranging (LIDAR) sensor

The IMU sensor makes it possible to know the orientation of the robot in the environment. The UGW sensor detects the presence of corrosion on the metal surface by emitting and receiving ultrasonic waves. The LIDAR sensor detects obstacles in the environment. The obstacles considered here are mainly the various robots inspecting the metal surface. The metal surface is considered to be flat and horizontal, and the robots are considered to be able to move on it without difficulty. Corrosion zones are detected by the emission of ultrasonic waves by a robot and the reception of these waves by another robot. Insofar as the wave received by one of the crawlers is altered, then there is a point of corrosion between the transmitter robot and the receiver robot. The detection of these corrosion zones is carried out in real time. The maximum range of ultrasonic waves is noted d_{max}. We approximate the propagation time of ultrasonic waves in the metal surface by zero time. This approximation is justified by the fact that the speed of propagation of ultrasonic waves in the metal surface is very high compared to the speed of the robots.

We use an occupancy grid to model the environment in which robots evolve during the acoustic inspection of metal structures. This grid allows us to represent and categorize the

different states of the areas of the metal surface. The occupancy grid is composed of cells, where each cell corresponds to a small region of the environment. In particular, we used a resolution of 0.05 meters per cell. We use three states to characterize these cells: unknown, empty and occupied. Unknown state refers to areas whose state has not yet been determined or detected. The empty state indicates areas where there is no corrosion detected, i.e. the metal surface is sound. Finally, the occupied state represents the identified corrosion areas, where the presence of defects or deterioration is detected.

By using this occupancy grid, we can track and update in real time the state of different areas of the metal surface during the inspection. This allows us to plan robot movements, optimize their trajectory and ensure full coverage of the surface to be inspected. In addition, this representation gives us a clear view of the state of corrosion of the metal structure, thus facilitating the analysis and evaluation of the results of the inspection.

In the rest of our proposal, we will detail the algorithms and methods used to update the occupancy grid according to the information collected by the robots' sensors. We will also discuss multi-robot navigation strategies that leverage this modeling to optimize data acquisition, with repect to time and accuracy, and improve the efficiency of acoustic inspection.

1.3 Background

We present here the state of the art in the field of multi-robot navigation and control for the acoustic inspection of metal structures. The objective of this literature review was to collect key information, analyze previous work and situate our project in the existing research context. The references and sources cited in this section provide a solid foundation of knowledge and expertise on the subject.

Initially, we were interested in the properties of UGWs and their applications in the field of tomography [2, 3], mapping of robots and metallic structures [1, 4, 5, 6], robots and sensors used in our project [7], multi-robot exploration [8, 9] as well as placement

strategies for detection [10, 11, 12].

In the paper [2], the authors propose a method to infer the geometry of metal plates using Lamb waves. They use beamforming [13] to estimate plate boundaries based on acoustic measurements. Experimental results show accurate inference of plate geometry. However, the authors are content to map the contours of the structures, without proposing a method to map the defects, which is the subject of our problem.

The article [1] presents a FastSLAM-based approach [14] for robotic inspection of metal structures using ultrasound. The authors propose a pioneer edge allocation method for multi-robot exploration, allowing fast and accurate inspection of structures. Our work takes robot localization and mapping as known, and focuses on trajectory planning for the inspection of metal structures. The approach used in this article can therefore be complementary to our work.

In the paper [4], the authors propose a method for mapping metallic structures using UGWs. They combine a Cartesian grid with specific features for defect detection. The experimental results show an accurate mapping of metallic structures. However, the authors are once again content to map the outlines of the structures, without proposing a method for mapping the defects.

The article [5] focuses on the localization of impacts in composite structures using a developed imaging method. The authors use piezoelectric sensors [15] to detect and localize impacts, and a wavelet transform method [16] to analyze acoustic signals. Experimental results show accurate detection and localization of impacts. The sensors used are similar to those used in our project, but these sensors are positioned in a fixed way on the structure, while we want a mobile strategy.

In the paper [3], the authors propose a high-resolution ultrasound tomography method for the quantification of wall thickness. They exploit the dispersive nature of Lamb waves to convert variations in thickness into variations in wave velocity, thus enabling accurate reconstruction of wall thickness. Experimental results show accurate reconstructions of

corrosion defects. This article was interesting to understand the properties of UGWs and their applications in the field of tomography.

The article [7], resulting from the work of the BugWright2 project, presents a magnetic robot system for the inspection and autonomous maintenance of large structures. The authors propose a localization framework based on a grid created from a point cloud, coupled with Ultra-Wideband (UWB) sensors and an IMU. They also incorporate a piezoelectric sensor for UGW detection for precise robot localization and structural feature mapping. It is typically these robots and sensors that are used in our work.

The paper [8] presents a planning algorithm for multi-robot exploration. This algorithm, called *MinPos*, is designed to efficiently allocate boundaries to robots in order to minimize the movement and time needed to explore the environment. It uses advanced optimization techniques to solve this problem effectively. However, our work focuses on structural inspection for flaw detection. We want a detailed inspection of corrosion areas and not a global exploration of the environment.

The article [10] presents strategies for the optimal placement of surveillance cameras in art galleries. The authors propose methods to maximize surveillance coverage while minimizing the number of cameras needed. However, the sensors used in our project are sensors that provide information on a segment only, between a transmitter and a receiver, and not global information like a camera. The sensors used in our project are described in section 1.2.

In the article [9], the authors propose a method for automatically locating and sizing defects in structures using guided wave imaging. They use a convolutional neural network [17] to analyze guided wave signals and estimate defect sizes. The experimental results show the efficiency of the proposed approach to invert both synthetic and experimental data. This approach requires fixed sensors on the structure. We want a mobile approach, not requiring the deployment of sensors on the structure.

The article [6] presents an autonomous on-plate exploration for an inspection robot us-

ing UGWs. The authors propose a localization method based on a mesh created from a cloud of points and use measurements from IMU and UWB sensors. They also integrate a piezoelectric sensor into the system for precise robot location and structural feature mapping. In our approach, the location is assumed to be known. This work can be used for robot localization, although this is not the subject of our project. Nevertheless, the type of robot and sensors used are similar to those used in our project.

The article [11] presents effective measurement planning strategies for remote gas detection with mobile robots. The objective of the study is to optimize the planning of measurements so as to maximize gas detection accuracy while minimizing the time and resources required. The authors propose different approaches for planning measurements, including the use of exploration techniques based on the boundaries of detection zones, the selection of efficient trajectories to cover the environment and the reduction of the number of measurements necessary by using probabilistic models. The type of sensor used has characteristics similar to those of the sensors used in our project. However, our problem imposes movements of pairs of robots. The way to split the investigation into two phases, a rough inspection phase and a refinement phase, is also similar to our approach. However, this first phase is performed by fixed sensors, which is not desirable in our approach. We will also use a Traveling Salesman Problem (TSP) to optimize robot movements between areas of interest.

In the paper [12], the authors propose an efficient measurement planning method for remote gas detection with mobile robots. Their approach is to optimize the planning of measures in order to minimize the time and resources required. To do this, they use a convex relaxation technique in order to solve the optimization problem which allows to minimize the number of necessary measurements, while guaranteeing a complete coverage of the environment. This study is interesting for our problem and could inspire improvements of our approach in the optimization of the TSP used.

In summary, the works presented in this section are interesting for our problem, be-

cause they allowed us to deepen our knowledge of the problems related to guided wave tomography. The articles [11, 12] are the closest to our problem. However, these articles focus on covering the environment without worrying about the quality of the mapping of the areas of interest. Moreover, these items use fixed sensors on the structure for the first rough inspection phase, which is not desirable in our approach. This is why we propose a multi-robot navigation approach for the acoustic inspection of metallic structures in order to optimize the acquisition of data which will allow to carry out the tomography of metallic surfaces.

CHAPTER 2
METHODOLOGY

2.1 Solution Proposal

We present our proposed solution for the acoustic inspection of metal structures using multi-robot navigation strategies. We have developed three specific strategies to optimize data acquisition and enable the tomography of metallic surfaces. These three strategies are:

1. *Roller Painting* navigation strategy
2. *Nordic Skiing* navigation strategy
3. *Polygonal Investigation* navigation strategy

Among these strategies, the first two are non-reactive and can be considered as coarse exploration strategies, the goal being to quickly obtain a global coverage of the surface to be inspected. The third strategy is reactive and makes it possible to optimize the acquisition of data for the realization of the tomography. These three strategies aim to map the metal surface and detect areas of corrosion. We define these three navigation strategies in the following subsections. We also explain how the data structure used for mapping corrosion areas, an occupancy grid, is updated based on information collected by the robots' UGW sensors.

2.1.1 Occupancy Grid Update Process for Mapping

When scanning the surface to be inspected by a pair of transmitter and receiver robots, the transmitter robot emits an acoustic wave in the metal structure, which is then received by the receiver robot. The detection being considered as perfect, the receiver robot receives the wave emitted by the transmitter robot, without quasi-alteration of the power of the signal,

if and only if the line segment between the two robots does not cross a zone of corrosion. It is thus possible to determine whether a corrosion zone is present between the two robots by checking whether the signal received by the receiving robot is sufficiently powerful. Insofar as there is no detection of corrosion between the transmitter and the receiver, then the line segment between the two robots is considered to be free of corrosion. Otherwise, then the points of the line segment between the two robots are considered to be corrosion, with the exception of the points previously perceived to be free of corrosion. The presence of corrosion on the segment is therefore overestimated. The displacement strategies will aim to carry out several measurements, to reduce this overestimation, and approach the real shape of the corrosion.

We now only need to determine which cells of the occupancy grid are crossed by the line segment between the two robots. For this we use Bresenham's segment drawing algorithm [18] which is commonly used to determine the points of a discrete plane that need to be drawn in order to form an approximation of a line segment between two given points. We detail our implementation of this algorithm in section 2.2.

As the metal surface is explored, the occupation grid is updated based on the information gathered by the robots. More precisely, the cells of the occupancy grid that identify corrosion elements are updated with the occupied state, while the cells that identify healthy areas are updated with the empty state. We thus end up with an occupation grid which represents the state of corrosion of the metal surface, with, for each corrosion zone, an approximation of the convex envelope of the corrosion zone.

2.1.2 *Roller Painting* Navigation Strategy

The first navigation strategy we propose is the *Roller Painting* navigation strategy. We chose this name for this strategy because the movement of the robots during this strategy is similar to that of a paint roller when painting a wall. This strategy is based on a rough exploration of the surface to be inspected, where the robots move in a straight line on

parallel trajectories, guaranteeing global coverage of the inspection area. It is therefore a question here of carrying out a grid of the surface to be inspected.

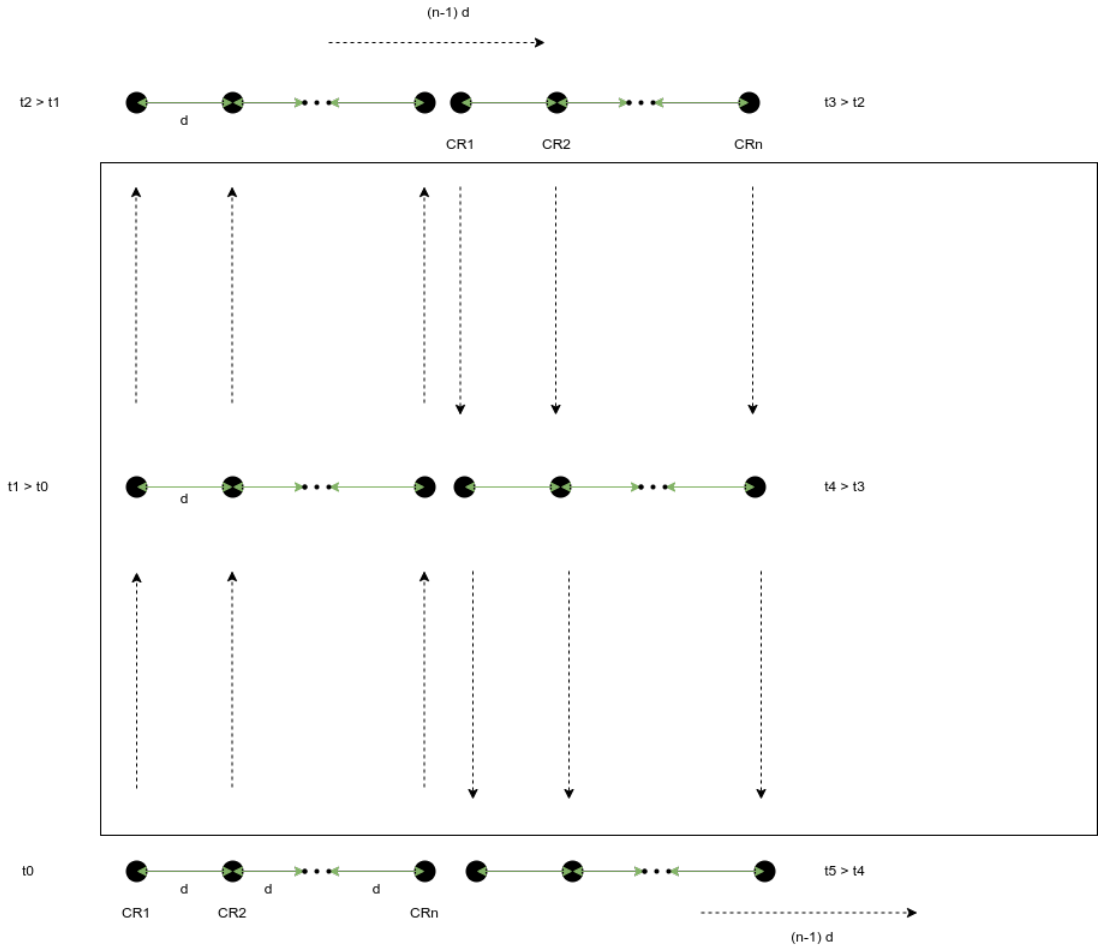


Figure 2.1: *Roller Painting* navigation strategy - vertical phase.

We present in Figure 2.1 a diagram describing the paint roller navigation strategy. This strategy consists of two phases, a vertical movement phase and a horizontal movement phase. The Figure 2.1 shows the first phase of vertical displacement. In order to achieve this strategy, a minimum of $n \in \mathbb{N}$ robots, $n \geq 2$, aligned horizontally and separated by a distance $d < d_{max}$, is used. These robots move vertically, simultaneously, following a parallel trajectory. Once the end of the surface to be inspected has been reached, the robots rotate 90 degrees and move horizontally, simultaneously, by a distance $(n-1) \cdot d$. They then perform a new 90 degree rotation and move again vertically, in a straight line,

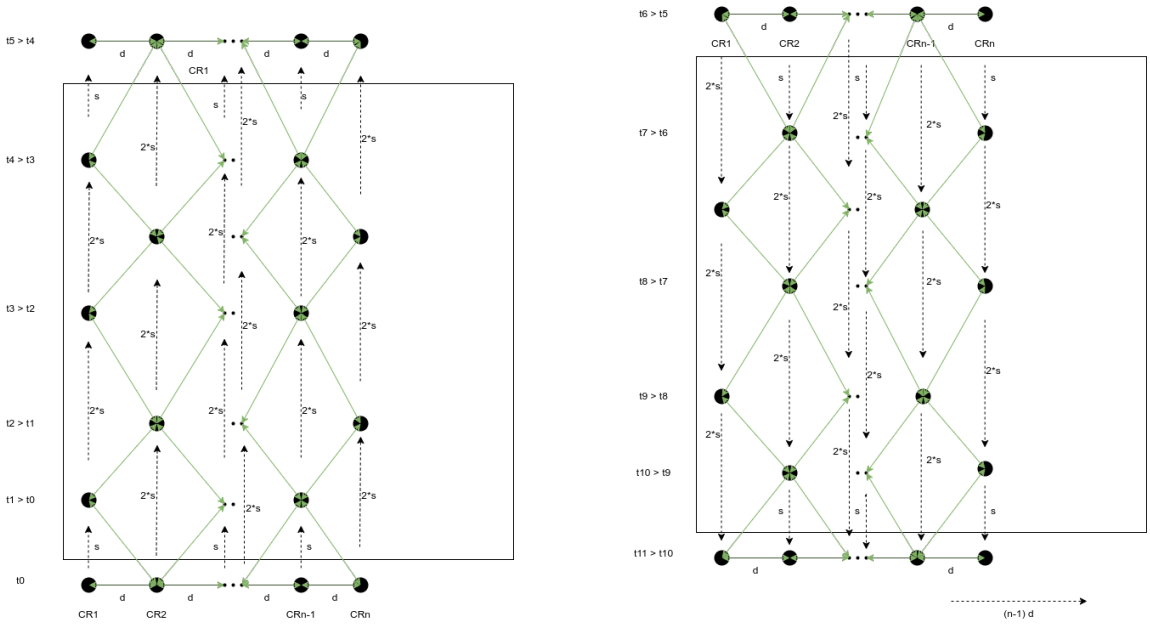
simultaneously, following a path parallel to each other, until they reach the other end of the surface to be inspected. This process is repeated until the metal surface is fully inspected. The same process is then repeated, but this time horizontally.

During this strategy, each robot is both a transmitter and a receiver of UGWs. If the distance separating a robot n_a from a robot n_b, $(n_a, n_b) \in \{1, 2, \ldots, n\}^2$, is less than the maximum propagation distance of the UGWs, d_{max}, then the robot n_a is able to receive the signal emitted by the robot n_b and vice versa. However, it is not necessary for a robot n_k, $n_k \in \{1, 2, \ldots, n\}$, to process signals received from all other robots. Indeed, the robots being aligned, the signals received from the robots n_{k-1} and n_{k+1}, are sufficient for the reconstruction of the state of the metallic surface. The waves emitted by the robots $n_1, n_2, \ldots, n_{k-2}$ and $n_{k+2}, \ldots, n_n$ are not useful for the robot n_k. The robot n_k can therefore ignore these signals and concentrate only on the signals received from the robots n_{k-1} and n_{k+1}. Insofar as the first signals perceived by the robot n_k are those emitted by the robots n_{k-1} and n_{k+1}, due to their proximity, it is possible for the robot n_k to filter signals received from other robots. This constitutes an optimization in terms of processing for each robot.

The fact that the robots move along a parallel trajectory and simultaneously, implies that the rays of the signal emitted by the transmitter robot and received by the receiver robot, always have an orientation of 0 radian for the vertical phase and an orientation of $\frac{\pi}{2}$ radians for the horizontal phase. There is therefore not a large variation in the orientation of the transmitted and received signal. Thus, this strategy will only be able to approach the convex envelopes of the corrosion zones by rectangles. Examples of occupancy grids resulting from the *Roller Painting* navigation strategy, represented as images, where the cells of the grid correspond to the pixels of the images, are shown in Appendix C, Figure C.1.

2.1.3 *Nordic Skiing* Navigation Strategy

The second strategy we propose is the navigation strategy *Nordic Skiing*. We chose this name for this strategy because the movement of the robots during this strategy is similar to the movement of a skier's skis. This strategy still consists of moving in a straight line and following parallel trajectories, but this time the robots move sequentially and no longer simultaneously. In this strategy, we wanted to increase the orientation diversity of the transmitted and received signal rays, in order to approach more precisely the convex envelopes of the corrosion zones.



(a) *Nordic Skiing* navigation strategy - first phase.

(b) *Nordic Skiing* navigation strategy - second phase.

Figure 2.2: *Nordic Skiing* navigation strategy - vertical phase.

Figure 2.2 presents a diagram describing the *Nordic Skiing* navigation strategy. This strategy also consists of two phases, a vertical movement phase and a horizontal movement phase. Figure 2.2 shows the first phase of vertical movement. In order to achieve this strategy, a minimum of $n \in \mathbb{N}$ robots, $n \geq 2$, aligned horizontally and separated by a distance $d < d_{max}$, is used. These robots move vertically, following a parallel path, but sequentially. The odd robots move in a straight line a distance s and stop. The even robots

then move in a straight line a distance of $2 \cdot s$ and stop. This process is repeated until the end of the surface is reached (Figure 2.2a). Then, the robots repeat this same process, in the opposite direction and so that the stopping points of the robots are not the same as those previously (Figure 2.2b). That is to say that this time, it is the even robots which start by moving in a straight line for a distance s and then stop. Then the odd robots move in a straight line for a distance of $2 \cdot s$ and then stop. The robots then move horizontally a distance $(n-1) \cdot d$ and repeat the same process until the metal surface is fully inspected. The same process is then repeated, but this time horizontally. In order for the various receiver robots to be able to receive the signals emitted by the transmitter robots, it is also necessary to impose that s be strictly less than $\frac{d_{max}}{2}$, i.e. $s < \frac{d_{max}}{2}$.

During this strategy, each robot is both a transmitter and a receiver of UGWs. Here, as with the *Roller Painting* navigation strategy, it is not necessary for a robot n_k, $n_k \in \{1, 2, \ldots, n\}$, to process the signals received by robots other than n_{k-1} and n_{k+1}.

The fact that the robots move following a parallel trajectory, but in a sequential way, implies that the rays of the signal emitted by the transmitter robot and received by the receiver robot, have an orientation of greater variation. Thus, this strategy makes it possible to approximate the convex envelopes of the corrosion zones by more diverse and precise shapes than rectangles. Examples of occupation grids resulting from the *Nordic Skiing* navigation strategy, represented in the form of images, where the cells of the grid correspond to the pixels of the images, are shown in Appendix C, Figure C.2 and Figure C.3.

2.1.4 *Polygonal Investigation* Navigation Strategy

The third strategy we propose is the *Polygonal Investigation* navigation strategy. We have seen, previously, that at the end of the realization of the *Roller Painting* navigation strategy, the convex envelope of the corrosion zones was approximated by a rectangle. This approximation is a little more precise for the *Nordic Skiing* navigation strategy. It would be interesting to have a greater degree of precision around potential areas of corrosion.

This is what we propose with the *Polygonal Investigation* navigation strategy. This strategy consists of investigating around potential areas of corrosion, previously detected by one of the two previous navigation strategies. It consists of positioning the robots around the corrosion zones and making them move along a polygonal trajectory, so that the rays of the signal emitted and received have an orientation of greater variation even around these zones.

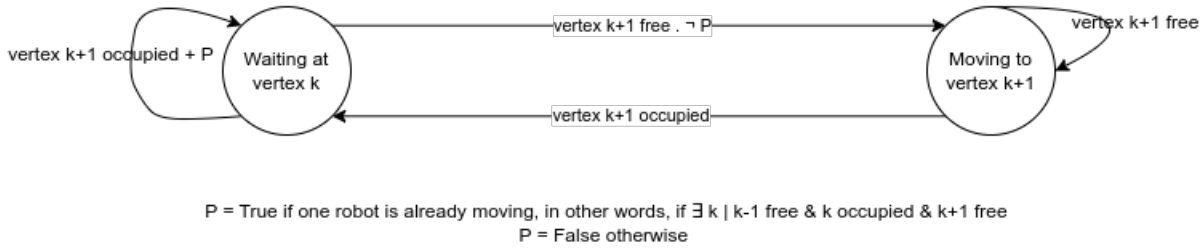
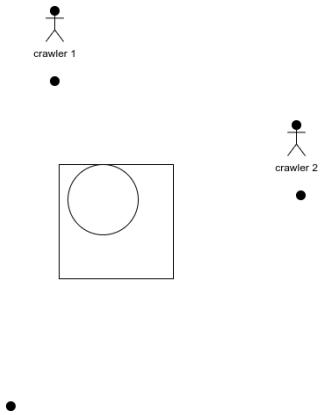


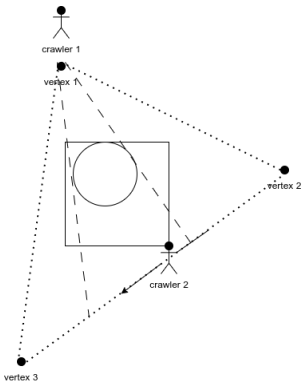
Figure 2.3: *Polygonal Investigation* navigation strategy.

We present in Figure 2.3 a finite state automaton describing the *Polygonal Investigation* navigation strategy. At the beginning of the *Polygonal Investigation* navigation strategy, each $n \in \mathbb{N}$ robots, $n \geq 2$, of $k \in \mathbb{N}$ teams, $k \geq 1$, are positioned on consecutive vertices of a polygon P with $p \in \mathbb{N}$ vertices, $p \geq 3$, enclosing the potential area of corrosion. To ensure the proper functioning of the strategy, it is necessary that the distance d^*, corresponding to the maximum distance between two vertices of the polygon P, be strictly less than d_{max}, i.e. $d^* < d_{max}$. In the latter, each robot has two states. The first consists of waiting and the second consists of moving along the polygonal trajectory, namely traversing the various vertices that make up the polygon. The robot capable of advancing, that is to say, whose next vertex is not occupied by another robot, advances. The others wait until the advancing robot reaches the last free vertex of the polygon. The process is then repeated for each robot of each team until the vertices occupied by the robots are the same as those occupied at the beginning of the *Polygonal Investigation* navigation strategy.

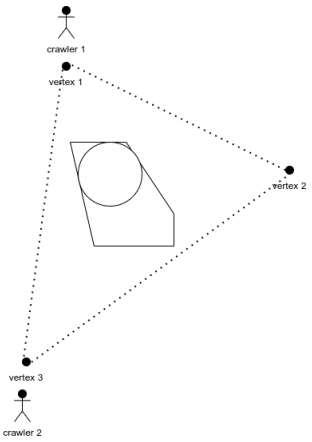
We present in Figure 2.4 an example of the different moving phases of the *Polygonal Investigation* navigation strategy with $k = 1$ team, $n = 2$ robots and $p = 3$ vertices. On these



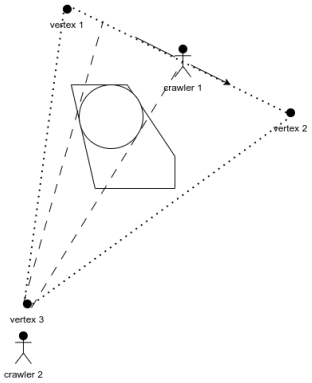
(a) Initial phase, crawlers are positioned on consecutive vertices of the polygon.



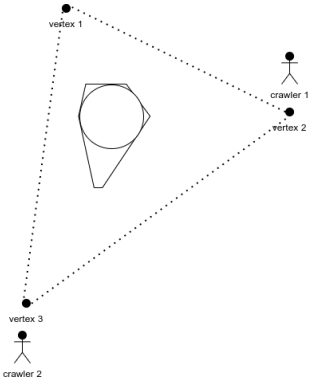
(b) First phase of movement, crawler 2 moves from vertex 2 to vertex 3.



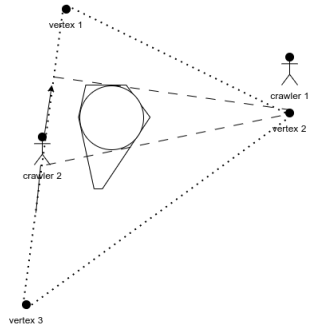
(c) Second phase, crawler 2 reaches vertex 3 and stops.



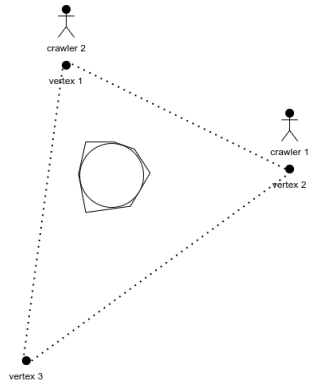
(d) Second phase of movement, crawler 1 moves from vertex 1 to vertex 2.



(e) Third phase, crawler 1 reaches vertex 2 and stops.



(f) Third phase of movement, crawler 2 moves from vertex 3 to vertex 1.



(g) Last phase, crawler 2 reaches vertex 1 and stops.

Figure 2.4: Movement phases of the *Polygonal Investigation* navigation strategy.

different figures, we have represented a corrosion zone by a circle and an approximation of this zone by a square enveloping the circle, as we can see in Figure 2.4a. The objective is to approach the corrosion zone as finely as possible. The corrosion zone is not known in advance, only the shape of the square enveloping the circle is known. Figure 2.4a presents the initial phase of the *Polygonal Investigation* navigation strategy, where the two crawlers are positioned on consecutive vertices of the polygon. Figure 2.4b shows the first moving phase of the *Polygonal Investigation* navigation strategy, while the crawler 2 moves from vertex 2 to vertex 3, until it reaches the latter, as we can see in Figure 2.4c. We can see that after the crawler 2 has reached vertex 3, part of the suspected corrosion area is removed and considered healthy, as we can see in Figure 2.4c. The crawlers keep moving until they reach their initial position, as we can see in Figure 2.4g.

Definition 1 (Phantom zone)**.** *A phantom zone is a corrosion zone detected by one of the navigation strategies, but which is not a corrosion zone. It is a false positive.*

The *Polygonal Investigation* navigation strategy has two advantages. The first is that it quickly eliminates phantom zones (Definition 1). The second is that it makes it possible to approach the convex envelopes of the corrosion zones by more diverse and precise shapes than rectangles due to the great variation in the orientation of the rays of the signal emitted and received by the robots around each vertex of the polygon.

This strategy requires two steps prior to its execution:

1. the extraction of the corrosion zones detected by one of the preceding navigation strategies.

2. determining the order of investigation of corrosion areas.

The first step can be solved using a Strongly Connected Components (SCC) graph decomposition algorithm. A SCC is defined in Definition 2. We then consider our occupation grid, resulting from the exploration of one of the two previously defined strategies, as an undirected graph $G = (V, E)$, where V is the set of vertices of the graph, corresponding to

the cells of the occupancy grid and E is the set of edges of the graph, corresponding to the adjacent cells of the occupancy grid. This problem is well known and there are simple algorithms to solve it, such as Tarjan's algorithm [19], of linear time complexity $O(|V| + |E|)$. We will not look further into this problem and entrust its resolution to the *OpenCV* library.

Definition 2 (SCC)**.** *A strongly connected component of a graph $G = (V, E)$ is a subset C of V such that for any pair of vertices $(u, v) \in C^2$, there is a path from u to v and a path from v to u.*

Definition 3 (Hamiltonian cycle)**.** *A Hamiltonian cycle is a cycle passing through all the vertices of a graph, once and only once.*

Definition 4 (TSP)**.** *Given a graph $G = (V, E)$, where V is the set of vertices of the graph and E is the set of edges of the graph, and a cost function $c : E \to \mathbb{R}$, the TSP consists in finding a Hamiltonian cycle (Definition 3) of minimal cost in G.*

Definition 5 (Multi-depot multiple Traveling Salesman Problem (mTSP))**.** *Given a graph $G = (V, E)$, where V is the set of vertices of the graph and E is the set of edges of the graph, a cost function $c : E \to \mathbb{R}$, and a set of depots $D \subset V$, the multi-depot mTSP is to find a set of cycles of minimum total cost in G, each going through one and only one deposit.*

The second step can be solved using a TSP (Definition 4) algorithm in the case where the number of teams k is equal to 1 and a mTSP (Definition 5) algorithm in the case where the number of teams k is strictly greater than 1. There are several solution paradigms to solve this type of problem. A first is to find an exact solution using an integer linear programming algorithm. A second is to find an approximate solution using a meta-heuristic.

Definition 6 (Non-deterministic Polynomial time (NP) Class)**.** *The class NP is the class of decision problems that can be solved by a non-deterministic algorithm in polynomial time.*

Definition 7 (NP-hard problem)**.** *A problem is NP-hard if it is at least as hard as problems of class NP. In other words, a problem is NP-hard if there is a polynomial reduction algorithm that transforms a problem of class NP into an instance of this problem.*

Definition 8 (NP-complete problem)**.** *A problem is NP-complete if it is both NP and NP-hard.*

The TSP is a NP-complete (Definition 8) problem. It can be treated as an integer linear optimization problem [20, 21]. To do this, we use the formulation presented in Equation 2.1.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{i \in V} x_{ij} = 1 && \forall j \in V \\
& \sum_{j \in V} x_{ij} = 1 && \forall i \in V \\
& \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset V, 2 \leq |S| \leq |V| - 1 \\
& x_{ij} \in \{0, 1\} && \forall i \in V, \forall j \in V
\end{aligned}
\tag{2.1}
$$

The objective function to be minimized from the Equation 2.1 is the sum of the distances between each pair of locations. The first two constraints ensure that each city is visited exactly once. The third constraint ensures that the cycle formed by the cities visited is simple, that is to say, that it does not contain sub-cycles. The last constraint ensures that the decision variables x_{ij} are binary, with $x_{ij} = 1$ if the robot moves from city i to city j and $x_{ij} = 0$ otherwise.

The mTSP is an NP-hard problem (Definition 7) [22]. This one can be solved using meta-heuristics like a genetic algorithm [23, 24]

In the next sections, we will detail each navigation strategy, exposing the specific algorithms and mechanisms used to implement our proposed solution. We will also analyze the performance and results obtained through extensive experimentation and evaluation.

2.2 Algorithm Implementations

In this section, we highlight some of the different technical implementations we have developed to support our multi-robot navigation and control solutions in the context of acoustic inspection of metal structures. We begin by describing our adaptation of Bresenham's line algorithm [18], widely used to determine which points of a discrete plane should be plotted in order to form a segment approximation of line between two given points. Next, we discuss the implementation of the *Roller Painting* algorithm, which allows the robots to move simultaneously, following parallel trajectories. We continue with the implementation of the *Nordic Skiing* algorithm, which allows the robots to move alternately, following parallel trajectories, thus modifying the orientation of the vector representing the direction of movement of the wave transmitted and received by the pair of robots. Additionally, we look at the implementation of the *Polygonal Investigation* algorithm, which allows robots to examine suspected areas of corrosion more precisely. Finally, we present Cohen's κ algorithm [25], used to assess the quality and reliability of the acoustic inspection results. We discuss in detail our implementation of this algorithm, which provides quantitative measures to evaluate the performance of robots in the inspection of metal structures. Each of these technical implementations contributes to the efficiency and accuracy of our multi-robot navigation and control approach, and will be discussed in detail in the following subsections.

Bresenham's Line Algorithm

We use Bresenham's line algorithm to determine the points of the line segment between the two robots. The algorithm is presented in algorithm 1. The part adapted to our problem is between lines 12 and 17 of the latter. At this point, we check if the signal strength is sufficiently impaired and if the point of the line segment between the two robots has not already been perceived as free of corrosion. If so, then the considered point is marked

Algorithm 1: Process of updating the occupancy grid using Bresenham's line algorithm.

Data: $P_1 \in \mathbb{R}^2$, $P_2 \in \mathbb{R}^2$, $pw \in \mathbb{R}$, $threshold \in \mathbb{R}$, G:
$l \times w \rightarrow [\text{UNKNOWN}, \text{EMPTY}, \text{OCCUPIED}]$, $l \in \mathbb{N}$, $w \in \mathbb{N}$
with P_1 and P_2 the two points to connect, pw the power of the UGW, $threshold$ the threshold above which the power of the UGW is considered undistributed and G the grid to update.
Result: The updated grid.

$p_0 \leftarrow$ from_position_to_grid_coordinate(P_1)
$p_1 \leftarrow$ from_position_to_grid_coordinate(P_2)
if *is_out_of_grid*(p_0) ***or*** *is_out_of_grid*(p_1) **then**
    return
end
$dx \leftarrow p_1.x - p_0.x$
$dy \leftarrow p_1.y - p_0.y$
$sx \leftarrow \text{sign}(dx)$
$sy \leftarrow \text{sign}(dy)$
$err = dx - dy$
while $p_0 \neq p_1$ **do**
    if $pwd \leq threshold$ ***and*** $G(p_0) =$ *UNKNOWN* **then**
        $G(p_0) \leftarrow$ OCCUPIED
    end
    else if $pwd > threshold$ **then**
        $G(p_0) \leftarrow$ EMPTY
    end
    $e2 \leftarrow 2 \times err$
    if $e2 > -dy$ **then**
        $err \leftarrow err - dy$
        $p_0.x \leftarrow p_0.x + sx$
    end
    if $e2 < dx$ **then**
        $err \leftarrow err + dx$
        $p_0.y \leftarrow p_0.y + sy$
    end
end

as corrosion, modeled by the value `OCCUPIED`. If the signal strength is not sufficiently altered, then the considered point is marked as being free of corrosion, modeled by the value `EMPTY`. Once all the points of the segment have been traversed, the grid G is updated with the new information. Bresenham's line algorithm thus contributes to the construction

of the occupation grid which makes it possible to locate the corrosion zones detected by the robots during the acoustic inspection of metal structures.

Roller Painting and *Nordic Skiing* Algorithms

We present in this subsection the implementations of the *Roller Painting* and *Nordic Skiing* algorithms. Their source code is available on GitLab, here[1] and here[2].

The implementations of these algorithms have been made using the Python programming language and the Robot Operating System (ROS) libraries. In these implementations, we use the ROS Task Manager [26] framework to manage the tasks of the robot inspectors. First, we initialize the ROS node and create a task client. Then, we retrieve the necessary parameters such as the speed of the crawlers, the crawler identifier, the distance between the crawlers, the overlap or the dimensions of the surface to be inspected.

The algorithms are then executed following a sequence of precise movements. For each crawler, we define vertical and horizontal trajectories using an iterative loop and mathematical calculations. Crawlers move along defined paths, using task client functions such as `AlignWithTarget` and `FollowLine` to maintain a precise path.

During the execution of the algorithms, the crawlers synchronize using the `SetStatusSync` and `WaitForStatusSync` functions of the task client. This ensures that the crawlers perform the movements in a coordinated manner and position themselves correctly to cover the entire metal surface. At the end of each movement step, the status is updated and synchronization is performed with the corresponding partner.

The implementation of the two algorithms *Roller Painting* and *Nordic Skiing* allow crawlers to explore the metal surface in a methodical and complete way. Using vertical and horizontal trajectories, crawlers traverse the surface overlapping previously inspected areas to ensure optimal coverage. Here we have used a 10 cm overlap between the different

[1] https://gitlab.georgiatech-metz.fr/bugwright2/bugwright2-ws/blob/cr_nav_strat/bugwright_ws/src/floor_nav/missions/peinture_au_rouleau.py

[2] https://gitlab.georgiatech-metz.fr/bugwright2/bugwright2-ws/blob/cr_nav_strat/bugwright_ws/src/floor_nav/missions/ski_nordique.py

vertical and horizontal trajectories.

Once the algorithms are completed, the execution time is recorded, providing an indication of how long it took to inspect the metal surface. The occupancy grid is also recorded in order to calculate the inspection score. This implementation is an essential step in our proposed solution for the acoustic inspection of metal structures and guarantees complete and effective coverage of the surface to be inspected.

Polygonal Investigation Algorithm

We present in this subsection the implementation of the *Polygonal Investigation* algorithm. The corresponding source code is available on GitLab [3] .

The implementation of this algorithm was also carried out using the Python programming language and the ROS libraries. In this implementation, we still use the ROS Task Manager [26] framework to manage the tasks of the inspector robots. First, we initialize the ROS node and retrieve the different parameters and more particularly the map of potential corrosion zones, on which we base ourselves for the inspection, which come from one of the two coarse strategies. Next, we extract the SCC from the map using the `connectedComponentsWithStats` function from the *OpenCV* library. This function uses Bolleli's spaghetti algorithm [27] to extract SCC from an image. For each of these components, we retrieve its center and its dimensions. Next, we construct a $p \in \mathbb{N}$ sided polygon around each center of a component. To do this, we place p points on an ellipse centered on the center of the component and whose axes are the dimensions of the component. We therefore have for each potential zone of corrosion a polygon with p sides which surrounds it. All that remains is to find the shortest path that passes through all the polygons. For this, we use the *Gurobi* library to solve a simple TSP in the case where the number of teams of robots $k = 1$. When $k > 1$, we use the genetic algorithm proposed by Elad Kivelevitch [28] to solve the multi-depot mTSP.

[3] https://gitlab.georgiatech-metz.fr/bugwright2/bugwright2-ws/blob/cr_nav_strat/bugwright_ws/src/floor_nav/missions/investigation_polygonale.py

Once the algorithms are completed, the run time is recorded, providing an indication of how long it will take to inspect the various potential areas of corrosion. The occupancy grid is also recorded in order to calculate the inspection score. This implementation is an essential step in our proposed solution for the acoustic inspection of metal structures and allows us to investigate potential areas of corrosion in an efficient way.

Algorithm for Calculating Cohen's κ

Assessing the quality and reliability of acoustic inspection results is essential to ensure accurate measurements of the condition of metal structures. In this subsection, we present the algorithm for calculating Cohen's κ [25], introduced in algorithm 2, a statistical measure commonly used to assess the agreement between the results obtained by the robots and a human reference.

κ	Interpretation
< 0	Disagreement
$0.00-0.20$	Very Low agreement
$0.21-0.40$	Low agreement
$0.41-0.60$	Moderate agreement
$0.61-0.80$	Strong agreement
$0.81-1.00$	Almost perfect agreement

Table 2.1: Interpretation of Cohen's κ according to Landis and Koch.

Cohen's κ calculation algorithm is based on the notion of concordance and discordance between the results of inspections carried out by robots and those carried out by human inspectors (ground truth). It takes into account the positive, negative, false positive and false negative results obtained during the acoustic inspection. This information is used to calculate the value of the Cohen coefficient, noted κ, with $\kappa = \frac{p_o - p_e}{1 - p_e}$, where p_o is the observed rate of agreement and p_e the expected agreement rate.

The algorithm proceeds in several steps. First, the results of the inspections carried out by the robots and the actual distributions of the corrosion zones are compared for each zone inspected. To do this, we compare the values of each cell of the occupancy grid, obtained at

Algorithm 2: Cohen's κ algorithm.

Data: I_0: $l \times w \times 3 \rightarrow [0..255]$, I: $l \times w \times 3 \rightarrow [0..255]$, $l \in \mathbb{N}$, $w \in \mathbb{N}$
with I_0 the ground truth image and I the image to score.
Result: $\kappa \in [0, 1]$

```
TP ← 0
TN ← 0
FP ← 0
FN ← 0
for i ← 1 to l do
    for j ← 1 to w do
        if is_label_1(I_0(i, j)) then
            if is_label_1(I(i, j)) then
                TP ← TP + 1
            end
            else
                FN ← FN + 1
            end
        end
        else
            if is_label_1(I(i, j)) then
                FP ← FP + 1
            end
            else
                TN ← TN + 1
            end
        end
    end
end
f_c ← ((TN+FN)(TN+FP)+(FP+TP)(FN+TP)) / (TP+TN+FN+FP)
κ ← (TP+TN−f_c) / (TP+TN+FN+FP−f_c)
```

the end of the inspection by the robots, with those of the ground truth. Having modeled the different test environments, we know the true distribution of the corrosion zones. Then, the results are grouped into four categories: positive agreement, negative agreement, positive discrepancy (false positives), and negative discrepancy (false negatives). These categories are used to calculate the observation and agreement rates observed between the robots and the true distribution of the corrosion zones. Cohen's κ is then calculated from observed observation and agreement rates, taking into account the possibility of concordance due to

chance. The closer Cohen's κ is to 1, the greater the agreement between the robot results and the ground truth results. On the other hand, a κ close to 0 indicates a low level of agreement, while a negative κ suggests a discrepancy between the results. An interpretation of Cohen's κ according to Landis and Koch is presented in Table 2.1.

We implemented this algorithm in our project, using the results of the acoustic inspections carried out by the robots and the maps composed of corrosion zones as a basis for comparison. This implementation allows us to obtain quantitative measures to evaluate the performance of our multi-robot navigation and control approach in the inspection of metal structures. In the next sections, we will detail the results obtained thanks to the application of this algorithm of the calculation of Cohen's κ.

2.3 Experiments

In this section, we present the experiments we conducted to validate and evaluate our different multi-robot navigation and control strategies in the context of acoustic inspection of metal structures. These experiments aim to demonstrate the efficiency, precision and reliability of our system in detecting and locating corrosion zones.

To carry out these steps, we chose to perform our experiments using *Gazebo*, a well-established simulation environment in the field of robotics. We started by building several test maps. These maps model a flat surface on which are placed simple geometric shapes, rectangles and circles, and more complex shapes, polygons between 3 and 8 vertices. These different geometric shapes represent the corrosion areas that we want to detect and locate. We present in Appendix A, in Figure A.1 the maps we built for our experiments. Each of these cards is sized 6 meters by 6 meters. The number of corrosion zones varies between 5, 8, 11, 15, 20 and 30 zones. The size and location of corrosion areas are randomly generated. For the maps of 5, 8, 11 and 15 zones, we generated 5 different maps in order to have more representative results. We did not allow ourselves to generate several maps for the 20 and 30 zone maps, the polygonal investigation time being too long.

We also simulated the UGW sensor by exploiting the simulation of a UWB sensor. This UWB sensor makes it possible to emit a pulse and to receive it. By measuring the signal strength, we are able to know whether the signal has passed through an object or not. The behavior of this UWB sensor is therefore similar to that of the UGW sensor, namely that it makes it possible to detect the presence of an object between two points, but not to locate it.

We evaluated the performance of the three navigation strategies in terms of Cohen's κ and inspection time. For the *Roller Painting* and *Nordic Skiing* strategy, we only used two robots. For these two strategies, we varied the distance d between the robots. For the *Nordic Skiing* strategy, we also varied the stride s between the robots. For the *Polygonal Investigation* strategy, we vary the number of robots n, the number of teams k and the number of sides p of the polygons used. We use the result of the *Roller Painting* navigation strategy as a starting point for the *Polygonal Investigation* strategy. We justify this choice by the fact that this strategy is the fastest and least accurate and therefore the most likely to benefit from an improvement from the *Polygonal Investigation* strategy, without reaching inspection times too long. We therefore vary the parameter d of this strategy. We summarize the experimental parameters used for each strategy in Table 2.2.

Strategy	Parameter	Values
Roller Painting	n	2
	d	1, 2, 3, 6 (meters)
Nordic Skiing	n	2
	d	1, 2, 3, 6 (meters)
	s	1, 2, 3, 6 (meters)
Polygon Investigation	initial strategy	*Roller Painting*
	d	1, 2, 3, 6 (meters)
	n	2
	k	1
	p	4, 6

Table 2.2: Experimental settings used for each navigation strategy.

During these simulations, we expect to have certain results. Among them, we expect the *Roller Painting* strategy to be the fastest, but also the least accurate. Conversely, we

expect the *Polygonal Investigation* strategy to be the most accurate, but also the slowest. We also expect the d parameter to have an impact on the accuracy and inspection time of the *Roller Painting* and *Nordic Skiing* strategies. A low d distance should provide better accuracy, but should also increase inspection time. Moreover, we expect that the s parameter will also have an impact on the accuracy and inspection time of the *Nordic Skiing* strategy. A low s stride should provide better accuracy, but should also increase inspection time. We also expect the p parameter to have an impact on the accuracy and inspection time of the *Polygonal Investigation* strategy. A low number of sides p should provide better accuracy, but should also increase inspection time. Next, we expect the parameters k and n to have an impact on the inspection time of the *Polygonal Investigation* strategy. A high number of teams k or a high number of robots n should allow to obtain a lower inspection time. Finally, we expect the number of corrosion zones to have an impact on the inspection time of the *Polygonal Investigation* strategy, but not on the *Roller Painting* and *Nordic Skiing* strategies. The higher the number of corrosion areas, the higher the inspection time should be for the *Polygonal Investigation* strategy. Finally, we expect the number of corrosion zones to have an impact on the accuracy of the different strategies. The greater the number of corrosion areas, the lower the accuracy should be. Indeed, the higher the number of corrosion zones, the higher the probability of phantom zones appearing for the *Roller Painting* and *Nordic Skiing* strategies. For the *Polygonal Investigation* strategy, the higher the number of corrosion zones, the greater the probability that two distinct corrosion zones were confused into one during the *Roller Painting* or *Nordic Skiing* strategies.

The different results from the different simulations carried out are available in Appendix C. On these images, it is possible to see in black the real areas of corrosion and in blue the areas detected as having corrosion by the various navigation algorithms.

CHAPTER 3

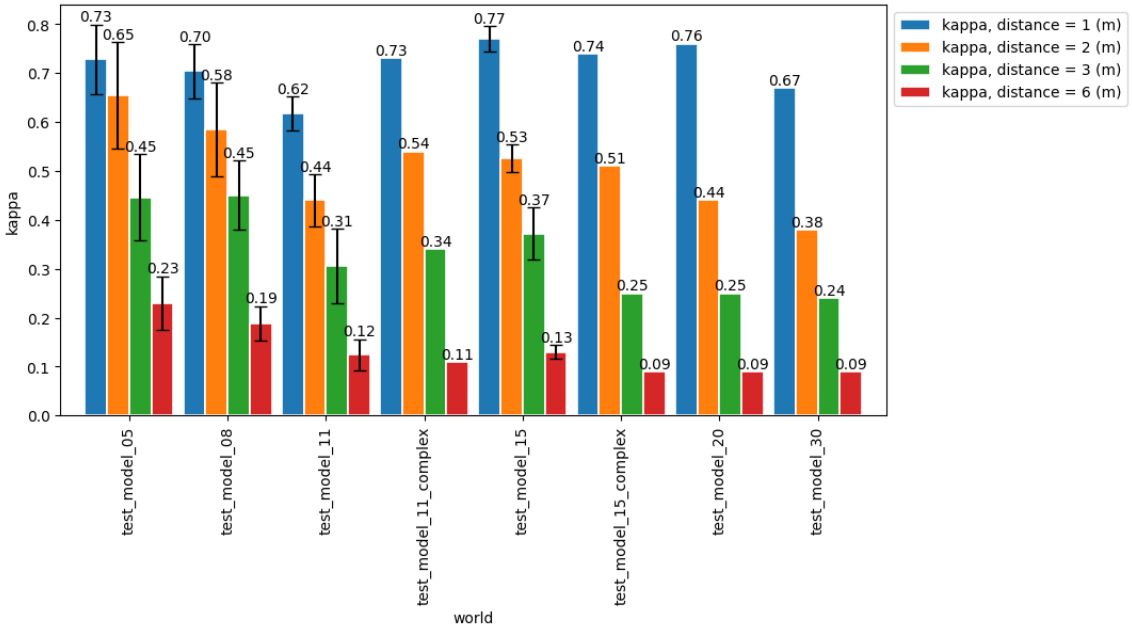
RESULTS

Roller Painting Navigation Strategy

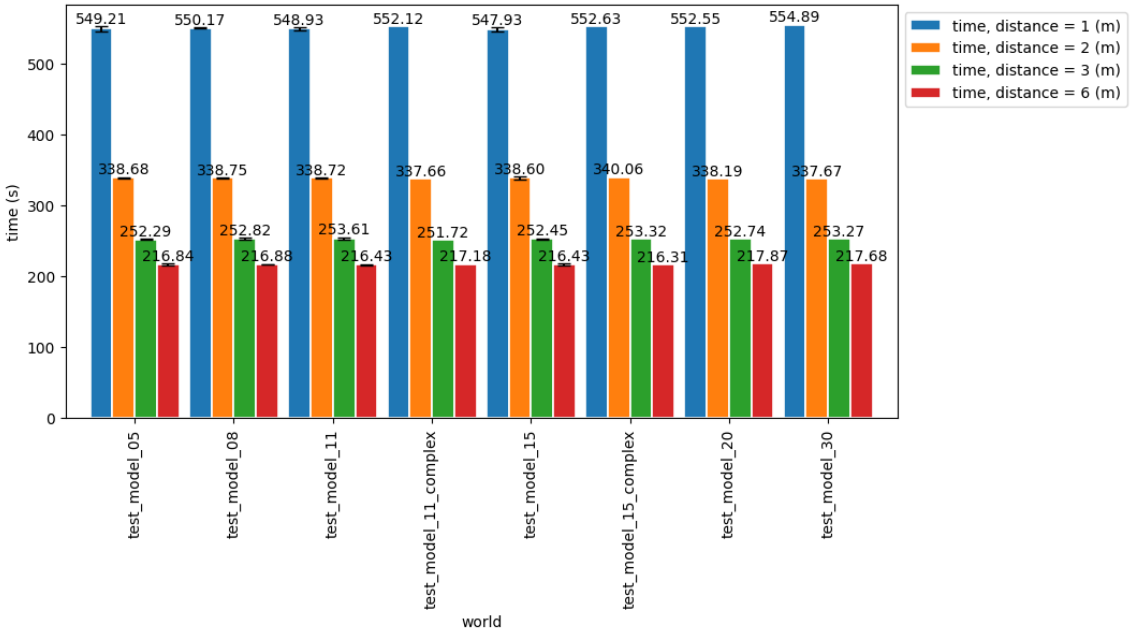
We summarize in Figure 3.1a the evolution of the Cohen score as a function of the density of the world for each value of d. We also summarize in Figure 3.1b the evolution of the inspection time according to the density of the world for each value of d.

First, we can observe that the Cohen score generally decreases with the number of corrosion zones. There are exceptions, notably for the map composed of 15 corrosion zones, where the Cohen score is higher than for the maps composed of 5, 8 and 11 corrosion zones. This is explained by the fact that in the maps composed of 5, 8 and 11 corrosion zones, we have introduced corrosion zones with elongated shapes unlike the map composed of 15 corrosion zones where the corrosion zones are all circles . Indeed, elongated corrosion zones have a greater probability of causing phantom zones to appear, illustrated in Figure 3.3, than circular corrosion zones. These phantom zones are areas free of corrosion which are detected by the crawlers. These are therefore false positives which reduce the Cohen score. These phantom zones are also more likely to appear when the density of the world is high and therefore the corrosion zones are closer to each other, or when the distance d between the two crawlers is high. This is what we can observe in Figure 3.2a where the Cohen score decreases when the distance d between the two crawlers increases. We observe that there seems to be a linear relationship between the Cohen score and the distance d between the two crawlers.

Then, we observe that the execution time of the *Roller Painting* algorithm is constant for each value of the number of corrosion zones. This was expected because the algorithm in question is an *a priori* algorithm and therefore does not depend on the number of corrosion zones. On the other hand, the execution time depends on the distance d between



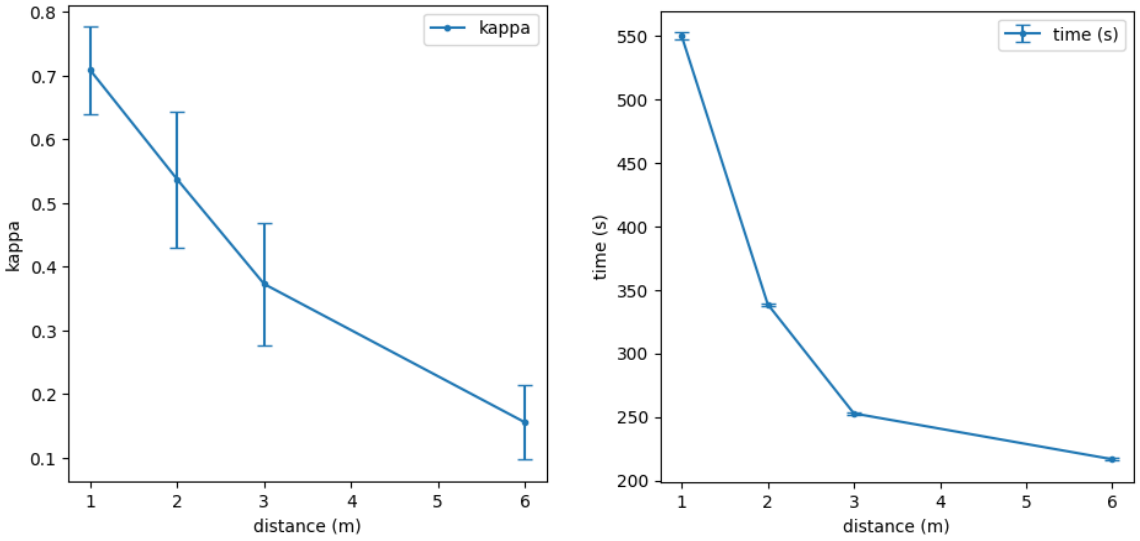
(a) κ according to the density of the world.



(b) Runtime based on world density.

Figure 3.1: Evolution of Cohen's κ and the execution time of the *Roller Painting* algorithm as a function of the density of the world and the distance between the robots.

the two crawlers. As we can see in Figure 3.2b, the execution time increases when the distance d between the two crawlers decreases. This is because the greater the distance d, the fewer moves the crawlers have to make to cover the map. There does not seem to be a



(a) κ depending on the distance between the two crawlers.

(b) Runtime depending on the distance between the two crawlers.

Figure 3.2: Evolution of Cohen's κ and the execution time of the *Roller Painting* algorithm as a function of the distance between the two crawlers.

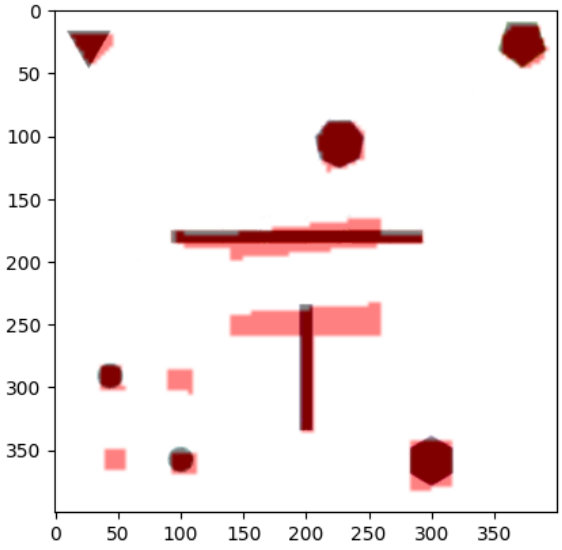


Figure 3.3: Example of a phantom zone located at the bottom left of the map.

linear relationship between the execution time and the distance d between the two crawlers. However, we would have expected that there would be a linear relationship between the execution time and the distance d between the two crawlers. It would be interesting to check if there was no bias introduced during the implementation of the algorithm.

We have also introduced two maps with more complex shapes than the base maps. These are visible in the Appendix A, in Figure A.1g and Figure A.1h. Unfortunately, we could not, for the sake of time, vary the position of the corrosion zones, as we did with the low density maps. However, there seems to be no significant difference between complex shaped maps and simple shaped maps. For example, for the maps with 15 forms of corrosion and the map with 15 complex forms of corrosion, the Cohen score only varies by 0.02 on average for a distance $d = 1$ and by 0.04 on average for a distance $d = 6$.

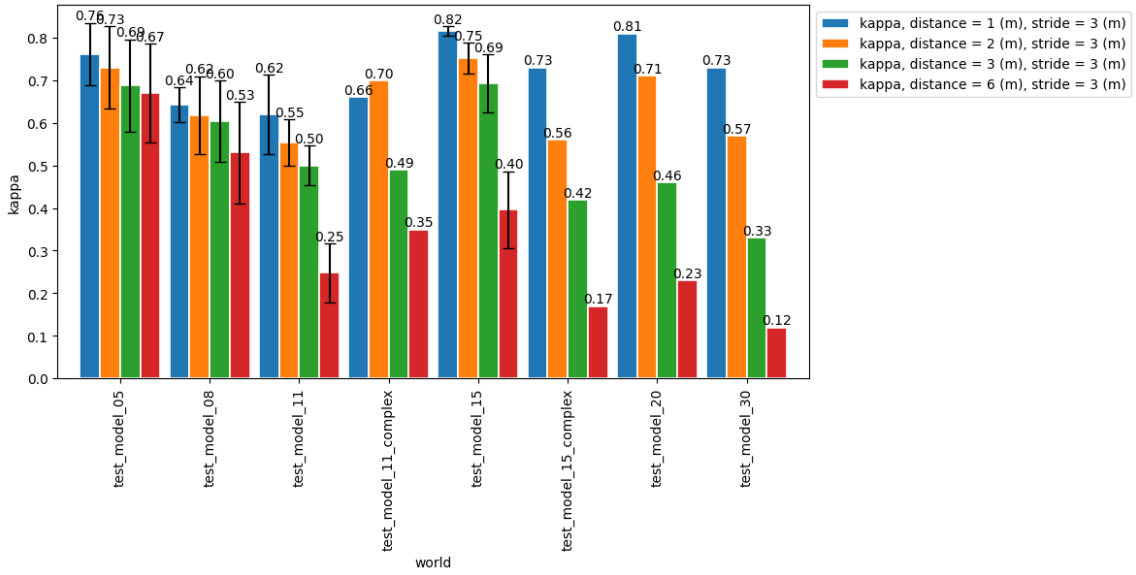
In the rest of this report, we will consider a distance $d = 3$ meters between the two crawlers for the *Roller Painting* algorithm.

Nordic Skiing Navigation Strategy

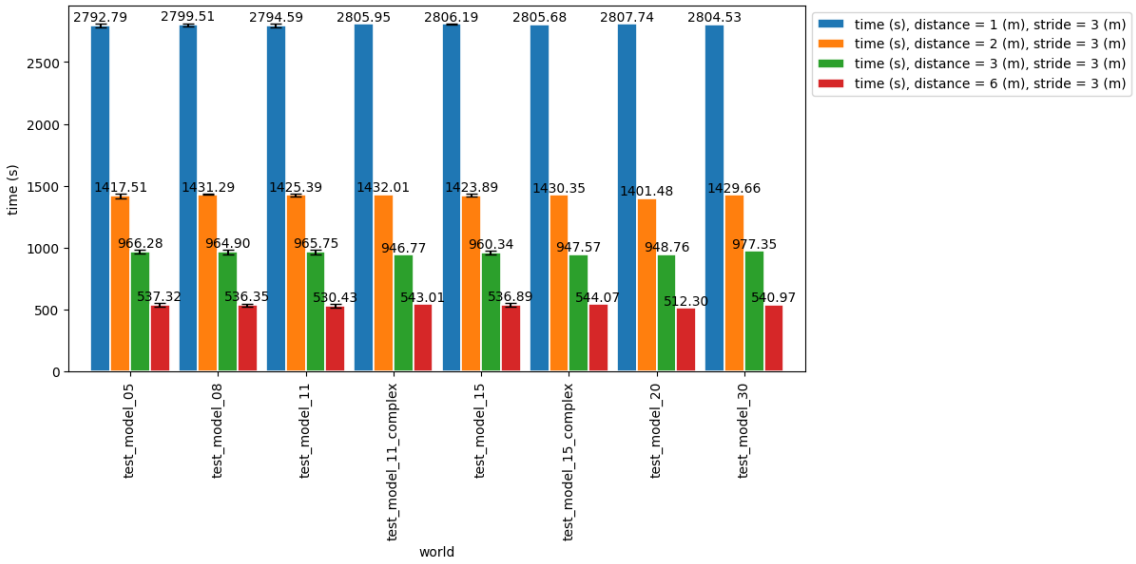
We are now going to analyze the results obtained for the *Nordic Skiing* algorithm. As for the *Roller Painting* algorithm, we varied the density of the world and the distance d between the two crawlers, but also the stride s used between the two crawlers. The Figure 3.4 presents the evolution of the Cohen score and the execution time of the *Nordic Skiing* algorithm as a function of the density of the world for different values of the distance d between the two crawlers and a stride $s = 3$ meters.

We have very similar results to those obtained for the *Roller Painting* algorithm. Indeed, we observe in Figure 3.4a that the Cohen score generally decreases when the density of the world increases. Moreover, the execution time of the *Nordic Skiing* algorithm, observed in Figure 3.4b, is constant for each value of the density of the world.

We can observe in the Figure 3.5 the evolution of the Cohen score and the execution time of the *Nordic Skiing* algorithm according to the density of the world for different values of the stride s between the two crawlers, and a distance $d = 3$ meters between the crawlers. In the Figure 3.5a, we observe that the Cohen score is the lowest for large values of densities and large values of d, as for $d = 6$ meters and the maps with 30 and 20 corrosion areas. This is explained by the fact that for large values of densities and d, the probability



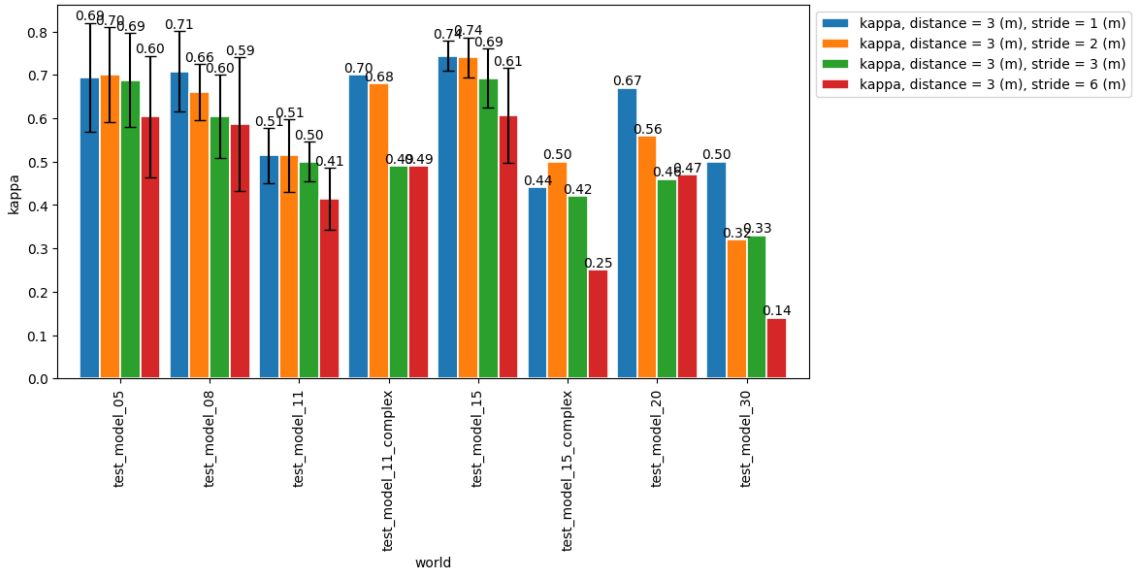
(a) κ according to the density of the world.



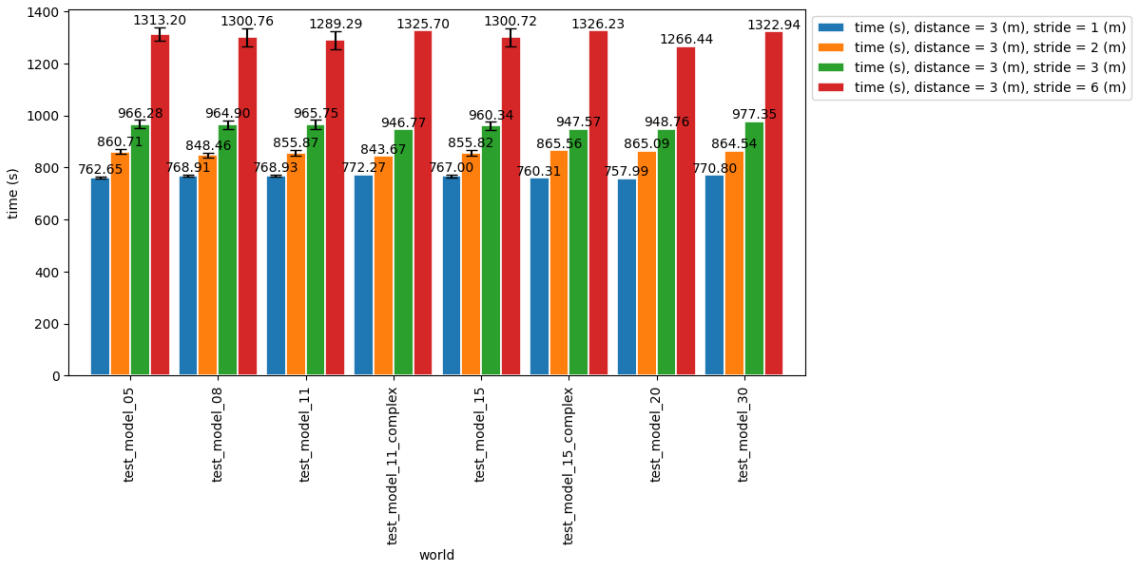
(b) Runtime based on world density.

Figure 3.4: Evolution of Cohen's κ and the execution time of the *Nordic Skiing* algorithm as a function of the density of the world for different values of the distance between the two crawlers.

that the signal rays cross corrosion zones is higher. There is therefore a greater chance that phantom zones will be created, which lowers Cohen's score. The elongated shapes of the corrosion zones are also a factor that lowers the Cohen score as studied previously. This is why we observe that the Cohen score is the highest for the map with the smallest density



(a) κ according to the density of the world.



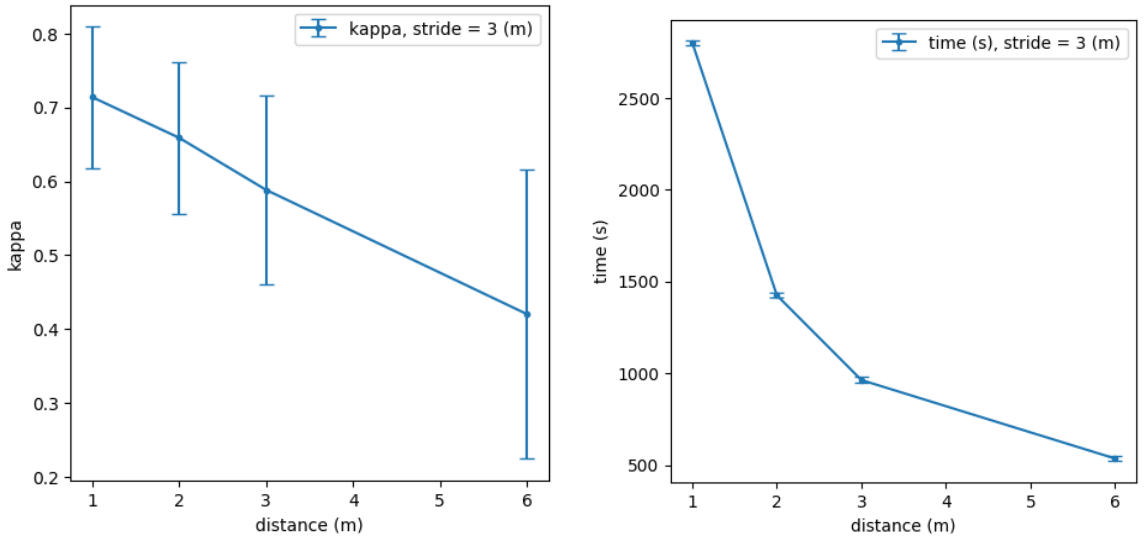
(b) Runtime based on world density.

Figure 3.5: Evolution of Cohen's κ and the execution time of the *Nordic Skiing* algorithm according to the density of the world for different values of the stride between the two crawlers.

and without elongated forms of corrosion, that is to say the map with 15 corrosion zones.

In Figure 3.5b, we observe that the execution time of the *Nordic Skiing* algorithm is constant for each value of the density of the world. This was expected as for the *Roller Painting* strategy. However, we observe that the execution time varies with the stride s

used. We would have rather expected the execution time to remain constant with the stride of the crawlers. Indeed, regardless of the value of the stride, the vertical and horizontal distance to be covered by the crawlers remains the same. This significant difference in execution time is due to the way we implemented the *Nordic Skiing* algorithm, which is not optimal. We did not make the crawlers stop at the ends of the plates, but we made them continue by a value of the stride s, in addition, for simplicity of implementation, without thinking that the impact on the time of execution would be significant.

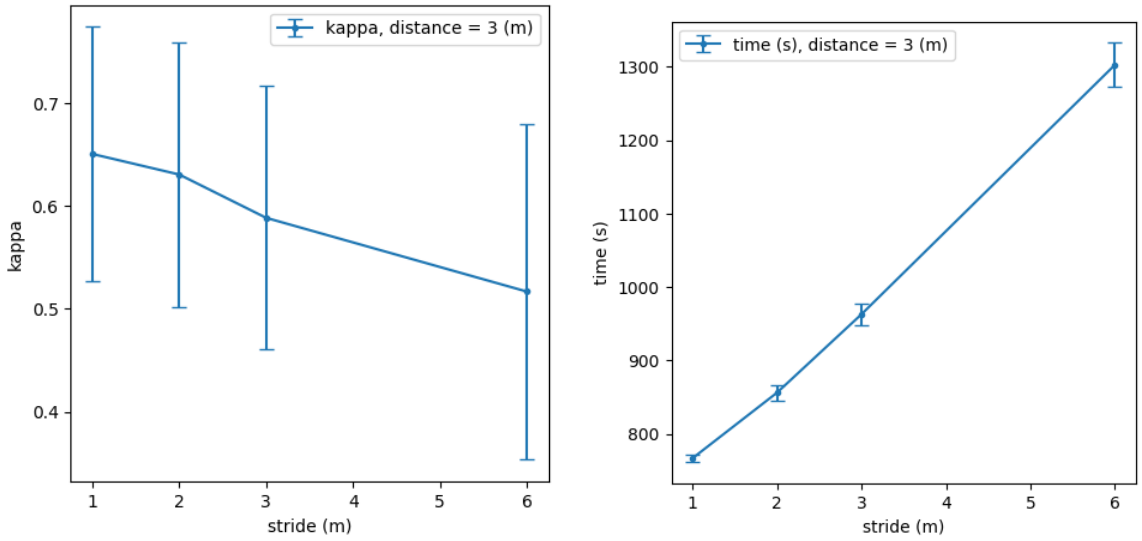


(a) κ according to the distance between the two crawlers.

(b) Runtime according to the distance between the two crawlers.

Figure 3.6: Evolution of Cohen's κ and the execution time of the *Nordic Skiing* algorithm according to the distance between the two crawlers.

In the Figure 3.6, we observe the evolution of the Cohen score and the execution time of the *Nordic Skiing* algorithm according to the distance which separates the two crawlers for a stride of 3 meters. The score seems, as for the strategy *Roller Painting*, to follow a linear relation with the distance which separates the two crawlers. Execution time also appears to follow a linear relationship with the distance between the two crawlers. The fact that the curve in Figure 3.6b is not a straight line is due to the fact that the smaller the distance between the crawlers, the greater the number of rotations that the crawlers must perform. However, the rotation time is not negligible in the execution times of the algorithms.



(a) κ according to crawler stride.

(b) Runtime according to crawler stride.

Figure 3.7: Evolution of Cohen's κ and the execution time of the *Nordic Skiing* algorithm according to the crawler stride.

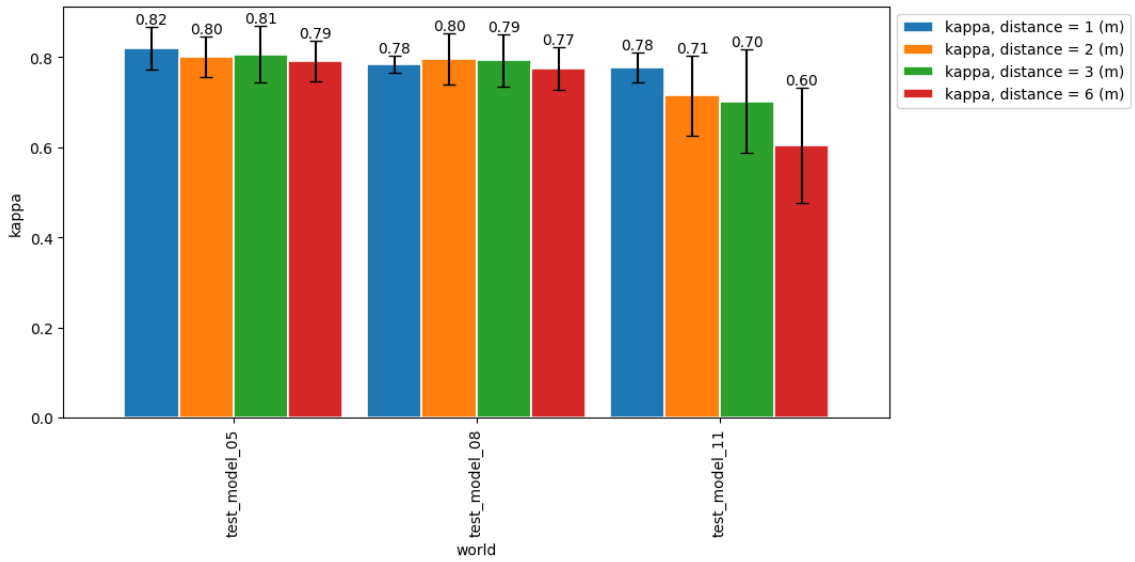
In the Figure 3.7, we observe the evolution of the Cohen score and the execution time of the *Nordic skiing* algorithm according to the crawlers' stride s for a distance $d = 3$ meters. The score seems to follow a linear relationship with the stride of the crawlers. The smaller the stride, the higher the score. This is consistent with what we explained previously. The larger the stride, the greater the chance of creating phantom zones and therefore of reducing Cohen's score. It should be noted however that there is a large variation in the score for the different values of the stride. It therefore seems that the impact of the value of the stride on the score is rather weak unlike the impact of the value of the distance on the score. Execution time seems to follow a linear relationship with crawler stride. As explained previously, the latter should have been constant, but our implementation makes the execution time depend on the stride of the crawlers.

Again, it seems that the score and execution time are not affected by whether the shapes are complex or not.

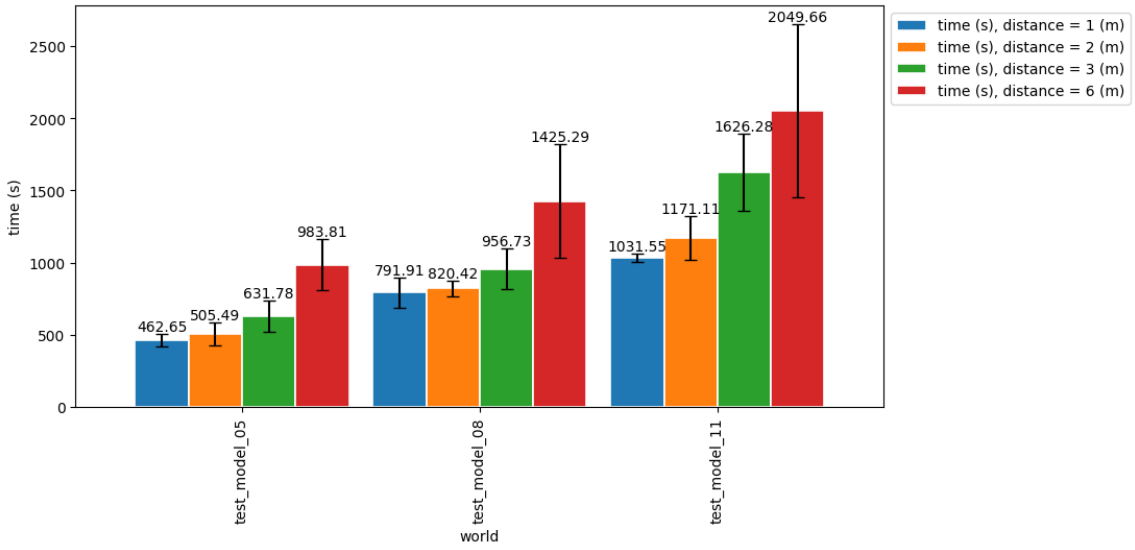
In the rest of this report, we will consider a distance $d = 3$ between the two crawlers and a stride $s = 3$ for the *Nordic Skiing* algorithm.

Polygonal Investigation Navigation Strategy

We then tested the *Polygonal Investigation* algorithm on worlds composed of 5, 8 and 11 corrosion zones. The inspection strategy is based on the results of the *Roller Painting* strategy. As explained previously, we justify this choice by the fact that the *Roller Painting* strategy is the fastest of the *a priori* strategies that we have implemented.



(a) κ according to the density of the world.



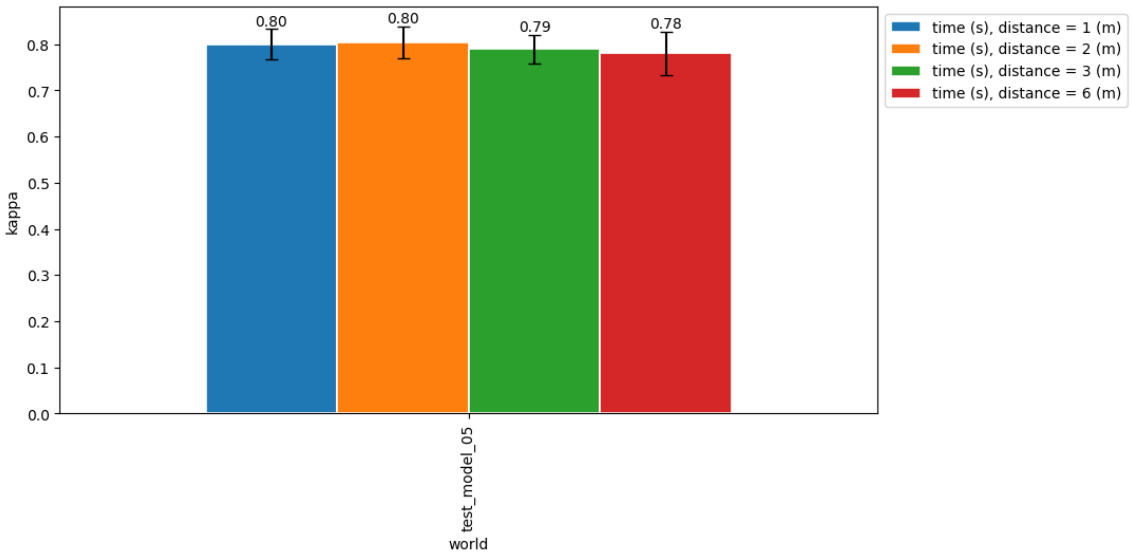
(b) Runtime based on world density.

Figure 3.8: Evolution of Cohen's κ and *Polygonal Investigation* algorithm runtime according to the world density for different distances between crawlers with a 4-sided polygon.

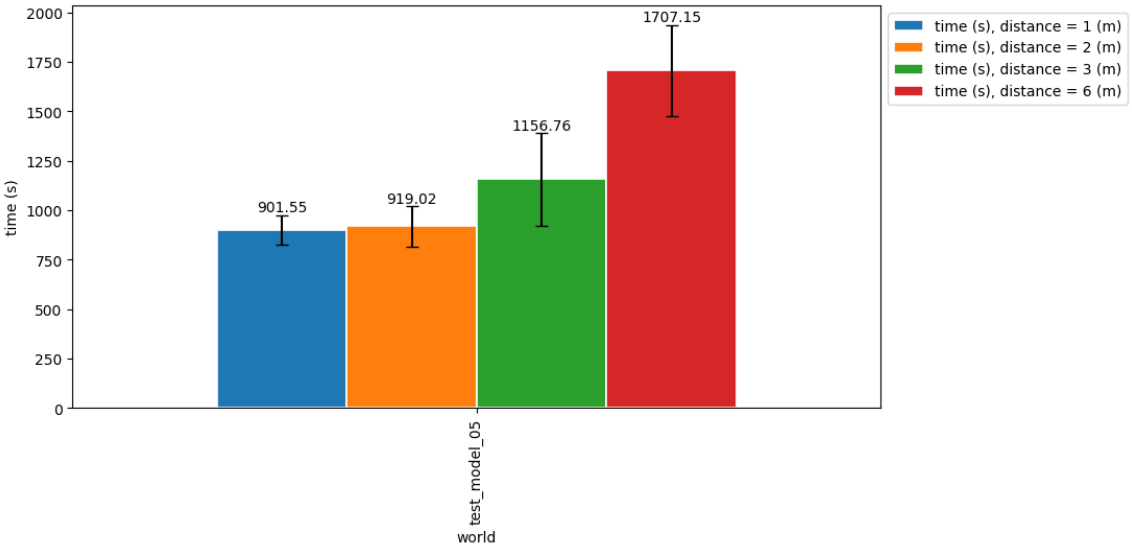
The Figure 3.8a shows the evolution of the Cohen score according to the density of the world for each value of d used in the *Roller Painting* strategy. We used a 4-sided investigation polygon. First, we observe that Cohen scores are relatively independent of the distance between crawlers for maps with 5 and 8 corrosion zones. This is an exciting result, because it means that we can use the *Polygonal Investigation* strategy based on the results of the *Roller Painting* strategy using a large distance between the crawlers, and therefore, a very fast *Roller Painting* strategy. Nevertheless, we can observe that for maps with 11 corrosion zones, the Cohen score is impacted by the distance between the crawlers, when the latter increases. We attribute this result to the fact that this map has elongated corrosion zones very close to each other, having the effect of blocking certain rays emitted and received during the polygonal inspection of an area. Polygonal inspection is therefore naturally impacted by the density of the world. However, we can imagine, when repairing metal structures, that it is more convenient to merge corrosion areas close to each other into a single corrosion area, although this is not considered in our problem.

Figure 3.8b shows the evolution of execution time according to the world density for each value of d used in the *Roller Painting* strategy. We used a 4-sided investigation polygon. We observe that the execution time increases with the density of the world in a linear way. This is an expected result, because the *Polygonal Investigation* algorithm has linear complexity as a function of the number of corrosion zones, the latter consisting in traversing all the potential corrosion zones and inspecting them. We also observe that the execution time increases with the distance between the crawlers. Indeed, the greater the distance between the crawlers, the greater the number of phantom zones at the end of the *Roller Painting* navigation strategy, and therefore the greater the number of potential corrosion zones. However, these phantom areas are quickly processed by the *Polygonal Investigation* algorithm. For example, for map 5 with 11 corrosion areas, we get 12 potential corrosion areas with a distance of 1 meter between crawlers after the *Roller Painting* strategy, versus 20 potential corrosion areas with a distance of 6 meters between the crawlers. However, we

observe an execution time of 1027 seconds for the first configuration versus 1616 seconds for the second configuration. So we have for a 67% increase in the number of potential corrosion areas, a 57% increase in execution time. The performance gain is not very large, but is still significant.



(a) κ according to the density of the world.



(b) Runtime based on world density.

Figure 3.9: Evolution of Cohen's κ and *Polygonal Investigation* algorithm runtime according to the world density for different distances between crawlers with a 6-sided polygon.

We also varied the size of the investigation polygon of the *Polygon Investigation* strat-

egy. We present in Figure 3.9a the evolution of the Cohen score as a function of the density of the world for map 5, for a polygon with 6 vertices. First, we do not observe a significant improvement in the Cohen score when the size of the investigation polygon increases. On the contrary, we observe an average decrease, although very weak, of the score. In theory, increasing the size of the investigation polygon should make it possible to better approach the convex envelop of the corrosion zones, and therefore to obtain a better Cohen score. However, we are limited in our implementation by the resolution used for the discretization of the map. For a more precise resolution, we should observe an improvement in the Cohen score.

We present in the Figure 3.9b the evolution of the execution time according to the density of the world for map 5, for a polygon with 6 vertices. Here, we naturally observe an increase in execution time when the size of the investigation polygon increases.

We would also have liked to vary the number of robots used for the polygonal investigation as well as the number of robot teams. However, we did not find an simple way to manage the collisions between the robots. The problem of managing collisions between moving entities is not trivial, and can be considered as a research topic in itself. It would be interesting in future work to implement such a solution and to analyze the performance of the *Polygonal Investigation* algorithm with these different parameters. Indeed, the execution time of the *Polygonal Investigation* algorithm should decrease when k and n increase.

In section 4.1, we will compare the performance of the *Polygonal Investigation* algorithm with those of the *Roller Painting* and *Nordic Skiing* algorithms. To do this, we will consider an investigation polygon, for the polygonal investigation, with $p = 4$ vertices.

CHAPTER 4

DISCUSSION

4.1 Comparisons and Discussion of Results

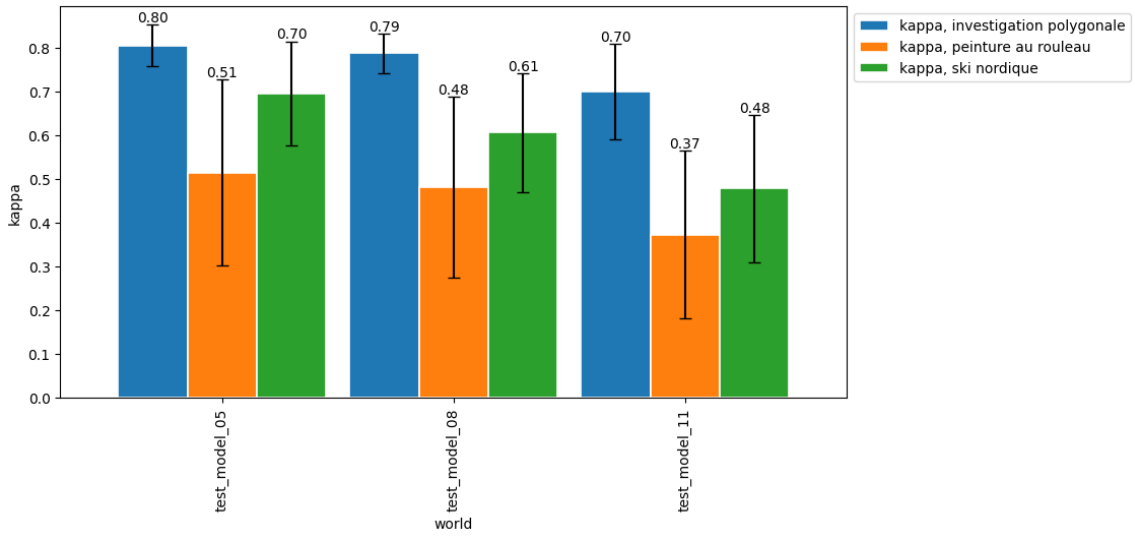
In this subsection, we compare the performance of different multi-robot navigation strategies.

We present on the Figure 4.1 the evolution of the Cohen score and the execution time according to the density of the world, for each algorithm. In this figure, we have represented the scores and execution times obtained, on average, for the different values of d and s, for the sake of readability. We therefore obtain, on average, $d = 3$ meters and $s = 3$ meters. A more detailed version, for each value of d, is available in Appendix B, in Figure B.1.

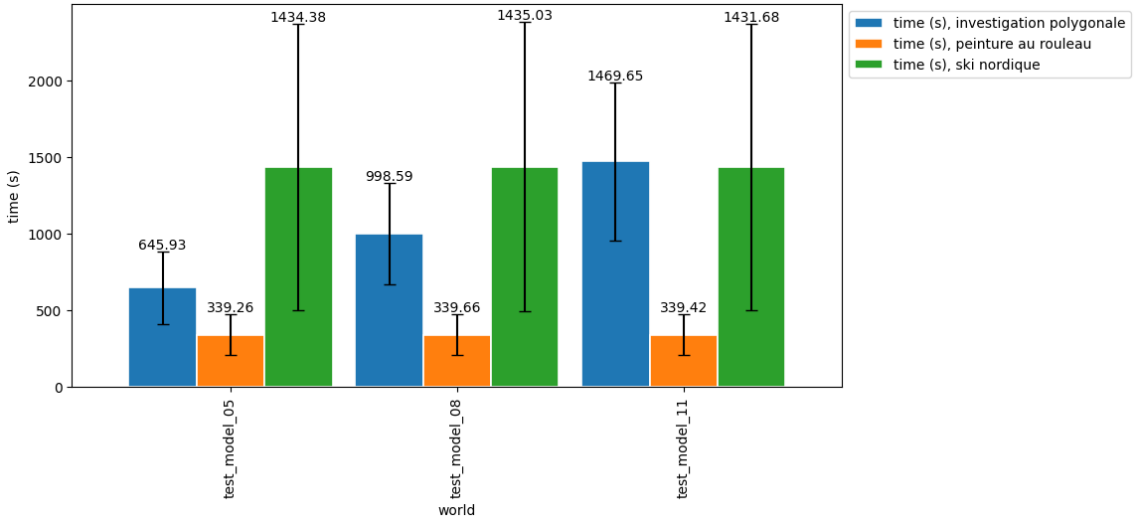
We can observe on the Figure 4.1a that the Cohen score obtained by the algorithm *Polygonal Investigation* is higher than those obtained by the *Roller Painting* and *Nordic Skiing* algorithms. Only the *Polygonal Investigation* strategy made it possible to obtain a Cohen score considered as *almost perfect agreement* (greater than 0.8) according to Landis and Koch.

We can observe on the Figure 4.1b that the execution time of the *Roller Painting* algorithm is lower than those obtained by the *Polygonal Investigation* and *Nordic Skiing* algorithms. For low map densities like those used in our experiments, the *Nordic Skiing* algorithm is the slowest. This result should be qualified. Indeed, for higher densities, the *Polygonal Investigation* algorithm becomes slower than the *Nordic Skiing* algorithm, as we can already almost see for the map with 11 corrosion zones. This is due to the fact that the *Polygonal Investigation* has a linear dependence on the number of corrosion zones, while the two other algorithms do not depend on the number of corrosion zones.

The Table 4.1 presents the performance gain provided by the *Polygonal Investigation*



(a) κ according to the density of the world.



(b) Runtime based on world density.

Figure 4.1: Evolution of Cohen's κ and the execution time of the different algorithms according to the density of the world for different distances between the crawlers.

	Gain in performance *Polygonal Investigation*	
compared to	**Cohen's κ**	**Runtime**
roll paint	+68.39%	+305.80%
Nordic skiing	+27.92%	-3.92%

Table 4.1: Performance gain provided by the *Polygonal Investigation* strategy compared to the *Roller Painting* and *Nordic Skiing* strategies.

strategy compared to the *Roller Painting* and *Nordic Skiing* strategies. Here, we considered the investigation time of the *Polygonal Investigation* strategy to be equal to the sum of the

investigation times of the *Polygonal Investigation* strategy and the *Roller Painting* strategy since the first strategy is based on the second to explore the corrosion zones. We can observe that the *Polygonal Investigation* algorithm achieves a Cohen score 68.39% higher than that obtained by the *Roller Painting* algorithm, although it is much slower than this last. On the other hand, the *Polygonal Investigation* algorithm makes it possible to obtain a Cohen score 27.92% higher than that obtained by the algorithm *Nordic Skiing*, while being faster than the latter.

The *Polygonal Investigation* strategy is therefore the best strategy in terms of performance, in our experiments, as well as a trade-off between the Cohen score and the execution time. The performance gain provided by the *Polygonal Investigation* strategy is explained by the fact that this strategy permits to greatly vary the vector of the signal emitted and received by the robots, and therefore to approach the convex envelope of the corrosion zones as closely as possible.

4.2 Theoretical Study of Properties of the Solution Proposal

Nordic Skiing Navigation Strategy

Proposition 1. *The angle of the signal emitted and received by the robots, for the Nordic Skiing navigation strategy, varies between* $-\tan^{-1}(\frac{s}{d})$ *and* $\tan^{-1}(\frac{s}{d})$.

We can observe the result of the Proposition 1, relying on the properties of trigonometry and the definition of the tangent function. We explain in Figure 4.2 the process undertaken to find $\alpha = -\tan^{-1}(\frac{s}{d})$. This proposal allows us to quantify the range of the orientation of the signal emitted and received by the robots.

Polygonal Investigation Navigation Strategy

Proposition 2 (Completeness)**.** *The Polygonal Investigation navigation strategy, defined by a polygon,* P *allows to cover all points inside the polygon* P.

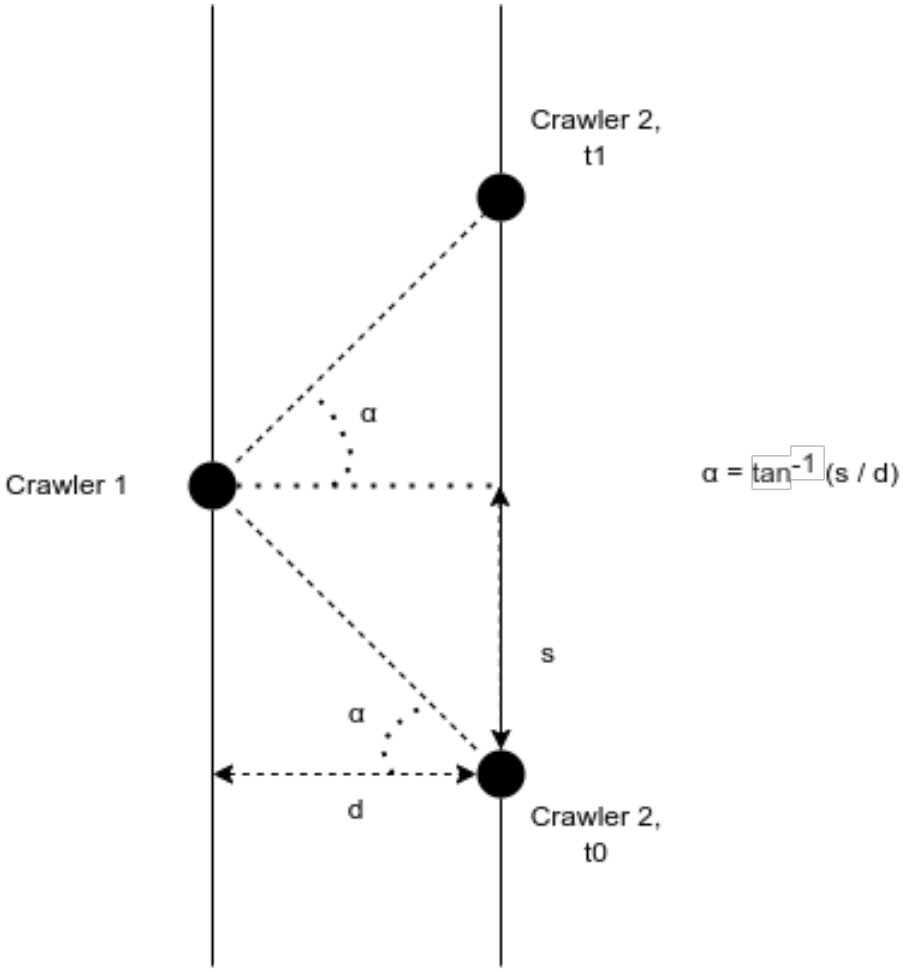


Figure 4.2: Orientation of the transmitted and received signal for the *Nordic Skiing* navigation strategy.

Proof. Let's prove Proposition 2.

- Let P be a convex polygon with p vertices used for the *Polygonal Investigation* navigation strategy.
- For simplicity, we consider a strategy with 2 robots, but the proof remains similarly the same for $n > 2$ robots.
- Consider the robot r_1 at vertex s_1 of polygon P and the robot r_2 at vertex s_2 of polygon P.
- Let p be a point inside the polygon P.

- Then, there exists a point p' such that p is on the segment $[s_1, p']$ and p' is on the edge of the polygon P, by definition of the convexity of P.
- By definition of the *Polygonal Investigation* navigation strategy, the robot r_2 moves on the contours of the polygon P and therefore in particular on the point p'.
- We therefore have for any point p inside the polygon P, there is a pair of positions for the robots r_1 and r_2 such that p is on the segment formed by the points where robots r_1 and r_2 are located.
- So all points inside the polygon P are covered by the *Polygonal Investigation* navigation strategy.

□

Proposition 3. *The approximation of the convex envelope of the corrosion zones at the end of the Polygonal Investigation, with a polygon with p vertices, $p \in \mathbb{N}$, is a polygon of at most $2p$ vertices.*

Proof. Let us give an intuition of the proof of the Proposition 3.

- We have, for each vertex of the polygon, there are two straight lines passing through this point and touching the corrosion zone without crossing it.
- We therefore have, for a vertex of the polygon, at most two lines which participate in the construction of the approximation of the convex envelope and therefore, of which a segment of these lines is an edge of the approximation of the convex envelope.
- We therefore have, for a vertex of the polygon, at most two edges of the approximation of the convex envelope.
- We therefore have, for a polygon with p vertices, at most $2p$ edges of the convex envelope approximation.

- We therefore have, for a polygon with p vertices, at most $2p$ vertices of the convex envelope approximation.

□

Conjecture 1. *If the number of vertices p of the polygon P, used during the Polygonal Investigation algorithm, tends towards infinity, and therefore approximate a circle, then the approximation of the convex envelope of the corrosion zone at the end of the Polygonal Investigation strategy is the convex envelope itself.*

CHAPTER 5
CONCLUSION

In conclusion, this study made it possible to implement and evaluate three strategies for performing autonomous muli-robot exploration in complex environments. The results obtained demonstrate the effectiveness of these approaches in solving the problem of inspecting metal surfaces and highlight their respective characteristics.

The first strategy, the *Roller Painting* strategy, made it possible to obtain satisfactory results in very short exploration times. The distance between the robots can be adjusted to optimize the results depending on the density of the inferred corrosion zones. This strategy is particularly suitable for use prior to the *Polygonal Investigation* strategy, because it allows to quickly get a rough view of potential corrosion areas.

The second strategy, the *Nordic Skiing* strategy, provided better results than the *Roller Painting* strategy, but with longer exploration times. This approach is more robust than the roller painting strategy, because it varies the orientation of the emitted and received UGW rays, thus making it possible to approach the corrosion zones more finely.

Finally, the third strategy, the *Polygonal Investigation* strategy, yielded the best results, in terms of accuracy. This reactive strategy makes it possible to refine the location of the corrosion zones based on the results of one of the two previous strategies. However, this strategy is more sensitive to collisions between robots, which can lead to less satisfactory results in some cases.

In terms of prospects, several areas of development can be envisaged. First of all, it would be interesting to deepen the polygonal exploration by looking for more robust methods for the management of collisions with the different robots. This would make this strategy more reliable and usable in a wide range of complex environments. The extension of this study to experiments with several teams of robots is also a promising avenue for

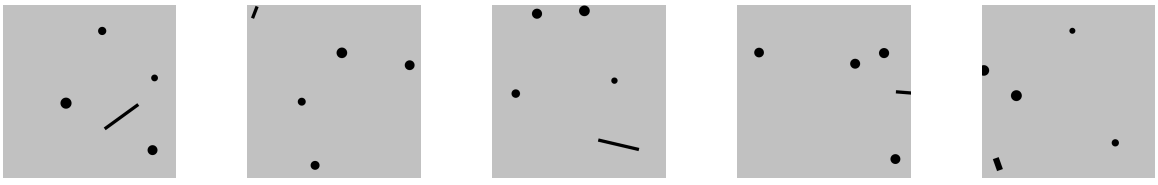
further accelerating the process of exploration and investigation. Finally, the deployment of this approach on a real system would be an important step to validate the results obtained in simulation and demonstrate the effectiveness of this approach in an industrial context.

In conclusion, this study has highlighted the advantages and limitations of three autonomous exploration strategies. The results obtained open the way to many development prospects, particularly with regard to improving the efficiency and robustness of existing approaches. These advances could have a significant impact in various fields, such as service robotics, space exploration or environmental monitoring.

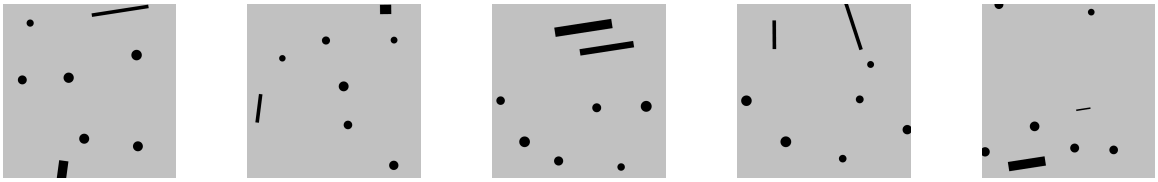
Appendices

APPENDIX A

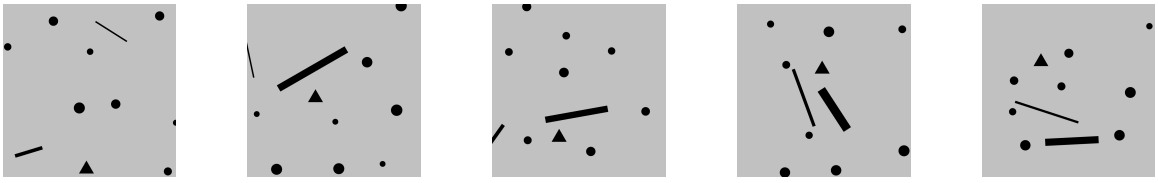
TEST ENVIRONMENTS



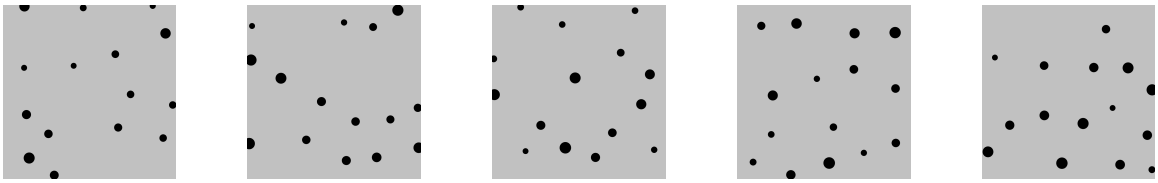
(a) Test worlds with 5 corrosion zones.



(b) Test worlds with 8 corrosion zones.

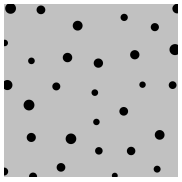


(c) Test worlds with 11 corrosion zones.

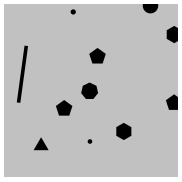


(d) Test worlds with 15 corrosion zones.

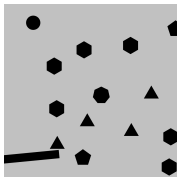
(e) Test worlds with 20 corrosion zones.



(f) Test worlds with 30 corrosion zones.



(g) Test worlds with 11 complex corrosion areas.

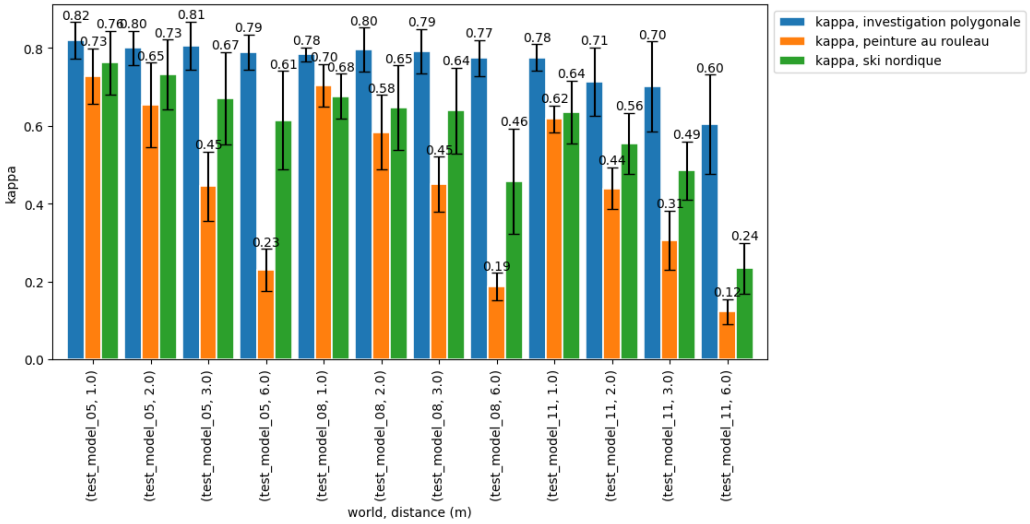


(h) Test worlds with 15 complex corrosion areas.

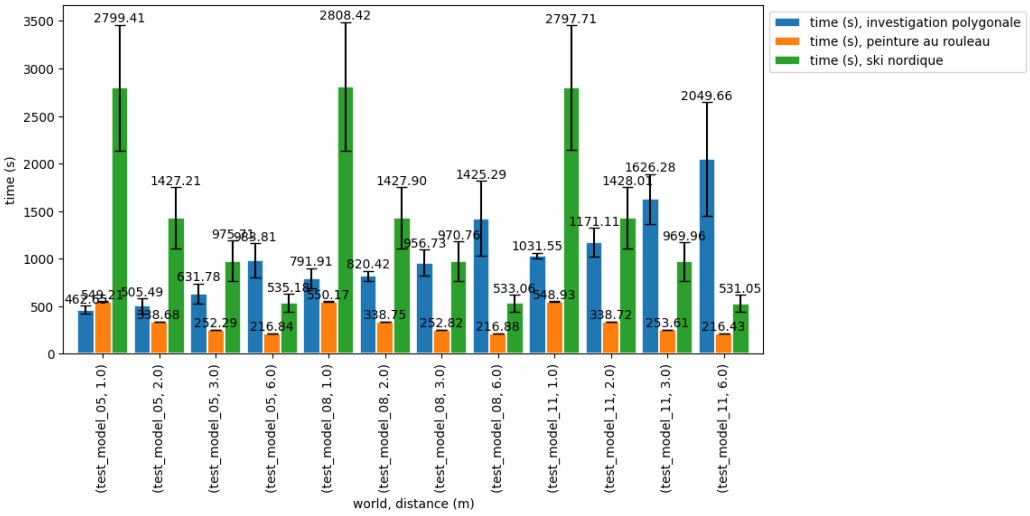
Figure A.1: Different test environments.

APPENDIX B

COMPARISON OF DIFFERENT NAVIGATION STRATEGIES



(a) κ according to the density of the world.

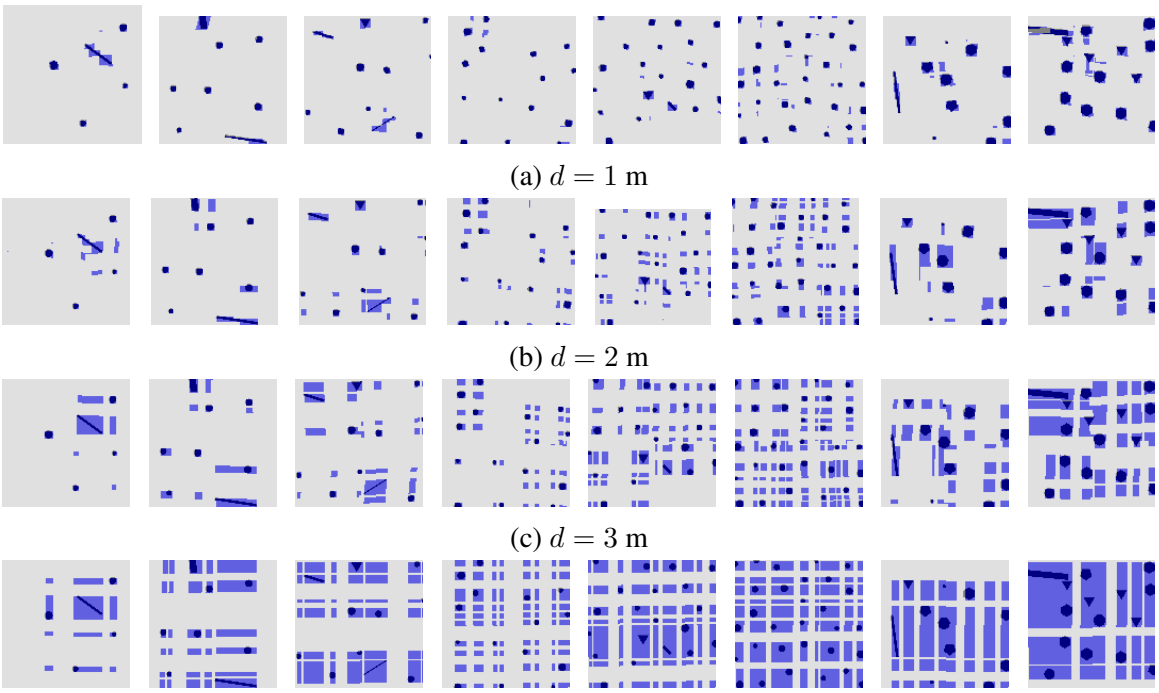


(b) Runtime based on world density.

Figure B.1: Evolution of Cohen's κ and the execution time of the different algorithms according to the density of the world for different distances between the crawlers.

APPENDIX C

INVESTIGATION RESULTS



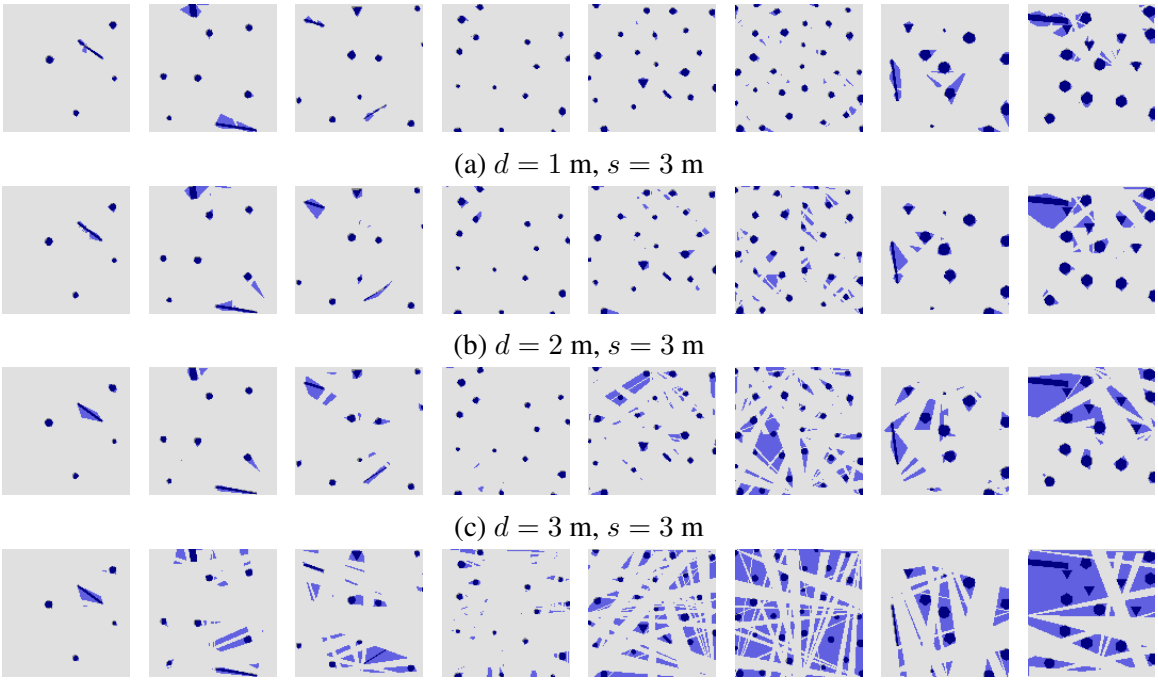
(a) $d = 1$ m

(b) $d = 2$ m

(c) $d = 3$ m

(d) $d = 6$ m

Figure C.1: Overlay of the investigation maps with the mapping of the corrosion zones obtained for the different test worlds, for the *Roller Painting* method.



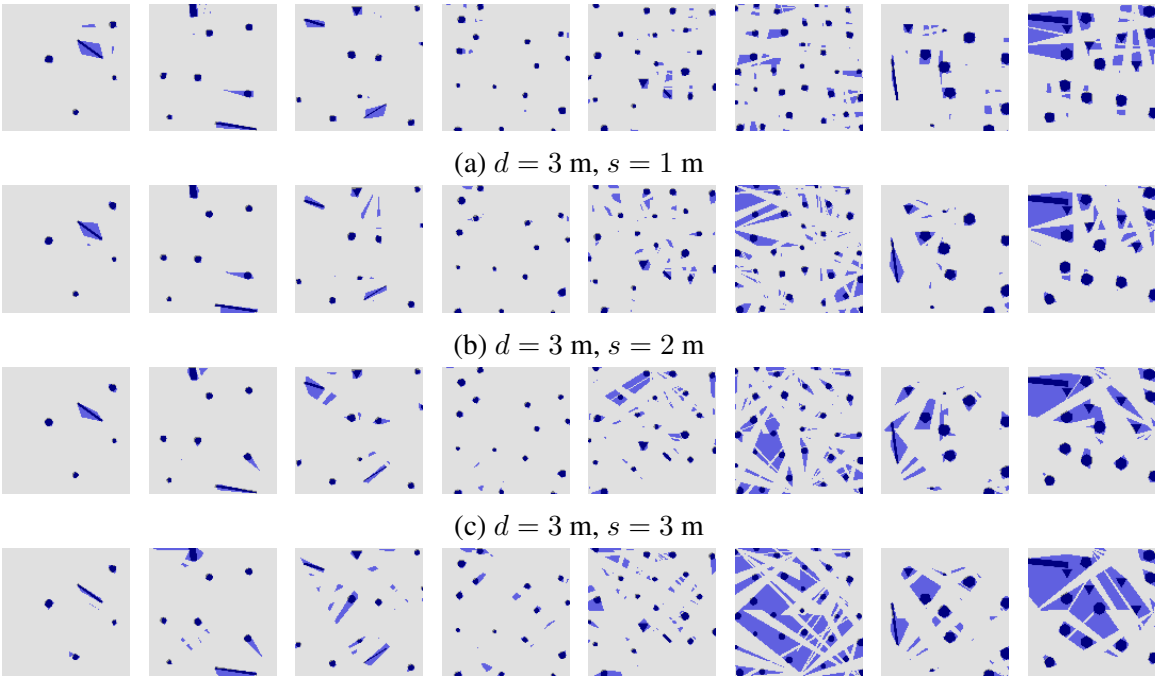
(a) $d = 1$ m, $s = 3$ m

(b) $d = 2$ m, $s = 3$ m

(c) $d = 3$ m, $s = 3$ m

(d) $d = 6$ m, $s = 3$ m

Figure C.2: Overlay of the investigation maps with the mapping of the corrosion zones obtained for the different test worlds, for the *Nordic Skiing* method - 1.



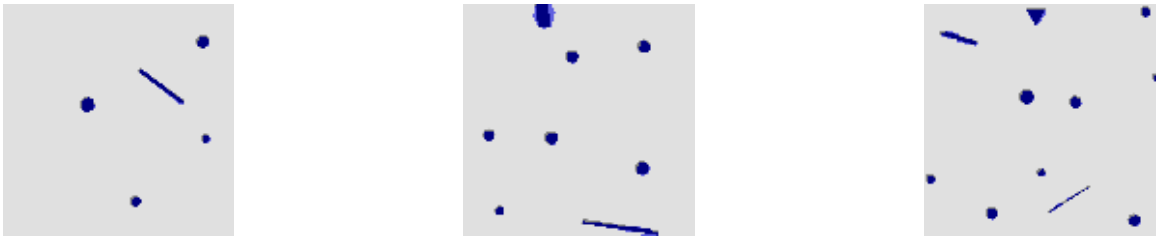
(a) $d = 3$ m, $s = 1$ m

(b) $d = 3$ m, $s = 2$ m

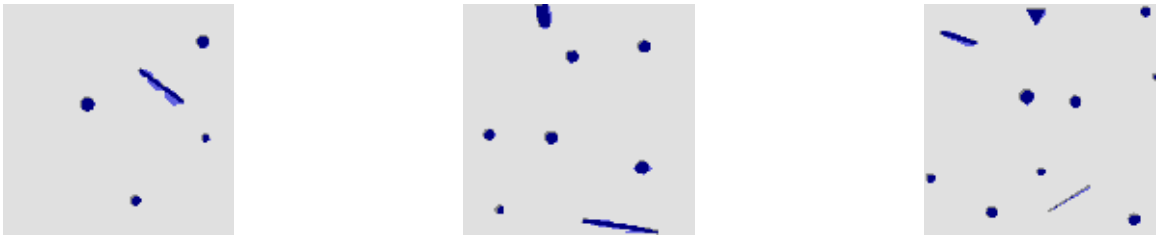
(c) $d = 3$ m, $s = 3$ m

(d) $d = 6$ m, $s = 3$ m

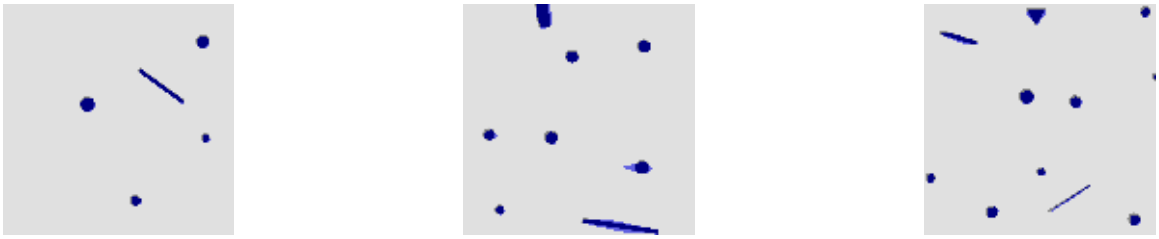
Figure C.3: Overlay of the investigation maps with the mapping of the corrosion zones obtained for the different test worlds, for the *Nordic Skiing* method - 2.



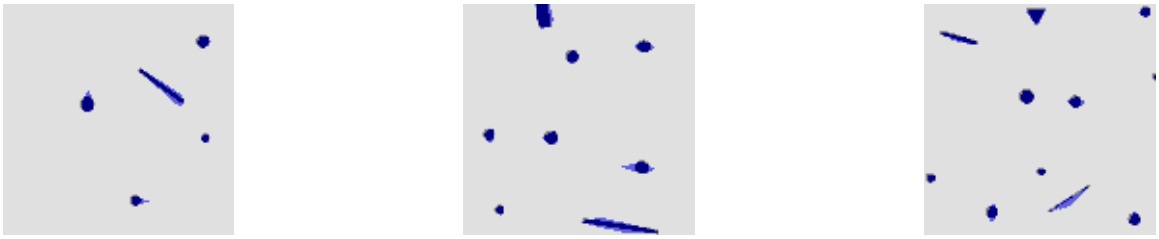
(a) $k = 1$, $n = 2$, $p = 4$, $d = 1$ m



(b) $k = 1$, $n = 2$, $p = 4$, $d = 2$ m

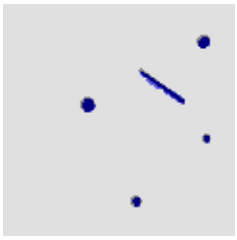


(c) $k = 1$, $n = 2$, $p = 4$, $d = 3$ m

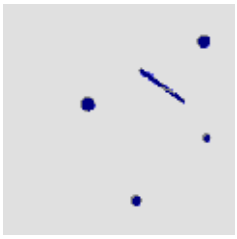


(d) $k = 1$, $n = 2$, $p = 4$, $d = 6$ m

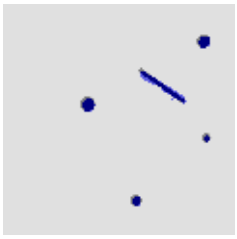
Figure C.4: Overlay of the investigation maps with the map of the corrosion zones obtained for the different test worlds, for the *Polygonal Investigation* method - 1.



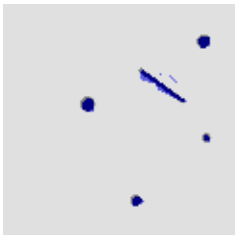
(a) $k = 1$, $n = 2$, $p = 6$, $d = 1$ m



(b) $k = 1$, $n = 2$, $p = 6$, $d = 2$ m



(c) $k = 1$, $n = 2$, $p = 6$, $d = 3$ m



(d) $k = 1$, $n = 2$, $p = 6$, $d = 6$ m

Figure C.5: Overlay of the investigation maps with the mapping of the corrosion zones obtained for the different test worlds, for the *Polygonal Investigation* method - 2.

REFERENCES

[1] O.-L. Ouabi, P. Pomarede, M. Geist, N. F. Declercq, and C. Pradalier, "A fastslam approach integrating beamforming maps for ultrasound-based robotic inspection of metal structures," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2908–2913, 2021.

[2] O.-L. Ouabi, P. Pomarede, N. F. Declercq, N. Zeghidour, M. Geist, and C. Pradalier, "Learning the propagation properties of rectangular metal plates for lamb wave-based mapping," *Ultrasonics*, vol. 123, p. 106 705, 2022.

[3] P. Huthwaite and F. Simonetti, "High-resolution guided wave tomography," *Wave Motion*, vol. 50, no. 5, pp. 979–993, 2013.

[4] O.-L. Ouabi, A. Ridani, P. Pomarede, N. Zeghidour, N. F. Declercq, M. Geist, and C. Pradalier, "Combined grid and feature-based mapping of metal structures with ultrasonic guided waves," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5056–5062.

[5] M. Faisal Haider, A. Migot, M. Y. Bhuiyan, and V. Giurgiutiu, "Experimental investigation of impact localization in composite plate using newly developed imaging method," *Inventions*, vol. 3, no. 3, 2018.

[6] A. Ridani, O.-L. Ouabi, N. F. Declercq, and C. Pradalier, "On-plate autonomous exploration for an inspection robot using ultrasonic guided waves," in *2021 European Conference on Mobile Robots (ECMR)*, 2021, pp. 1–6.

[7] G. Chahine, P. Schroepfer, O.-L. Ouabi, and C. Pradalier, "A magnetic crawler system for autonomous long-range inspection and maintenance on large structures," *Sensors*, vol. 22, no. 9, 2022.

[8] A. Bautin, O. Simonin, and F. Charpillet, "MinPos : A Novel Frontier Allocation Algorithm for Multi-robot Exploration," in *ICIRA - 5th International Conference on Intelligent Robotics and Applications - 2012*, C.-Y. Su, S. Rakheja, and H. Liu, Eds., ser. Lecture Notes in Computer Science, The original publication is available at www.springerlink.com, vol. 7507, Montréal, Canada: Springer, Oct. 2012, pp. 496–508.

[9] R. Miorelli, C. FISHER, A. Kulakovskyi, B. CHAPUIS, O. Mesnil, and O. d'Almeida, "Defect sizing in guided wave imaging structural health monitoring using convolutional neural networks," *NDT & E International*, vol. 122, p. 102 480, Jun. 2021.

[10] M. Couto, C. Souza, and P. De Rezende, "Strategies for optimal placement of surveillance cameras in art galleries," *GraphiCon 2008 - International Conference on Computer Graphics and Vision, Proceedings*, Jan. 2008.

[11] M. A. Arain, E. Schaffernicht, V. H. Bennetts, and A. J. Lilienthal, "The right direction to smell: Efficient sensor planning strategies for robot assisted gas tomography," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4275–4281.

[12] M. A. Arain, M. Cirillo, V. H. Bennetts, E. Schaffernicht, M. Trincavelli, and A. J. Lilienthal, "Efficient measurement planning for remote gas sensing with mobile robots," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3428–3434.

[13] Wikipedia contributors, *Beamforming — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Beamforming&oldid=1151960654, [Online; accessed 15-June-2023], 2023.

[14] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data," *Journal of Machine Learning Research*, vol. 4, May 2004.

[15] Wikipedia contributors, *Piezoelectric sensor — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Piezoelectric_sensor&oldid=1154129092, [Online; accessed 15-June-2023], 2023.

[16] ——, *Wavelet transform — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Wavelet_transform&oldid=1147185762, [Online; accessed 15-June-2023], 2023.

[17] ——, *Convolutional neural network — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=1159408824, [Online; accessed 15-June-2023], 2023.

[18] ——, *Bresenham's line algorithm — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Bresenham%27s_line_algorithm&oldid=1155124335, [Online; accessed 15-June-2023], 2023.

[19] ——, *Tarjan's strongly connected components algorithm — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Tarjan%27s_strongly_connected_components_algorithm&oldid=1148118528, [Online; accessed 15-June-2023], 2023.

[20] M. Diaby, "The traveling salesman problem: A linear programming formulation of," *CoRR*, vol. abs/cs/0609005, Jan. 2006.

[21] Gurobi, *Traveling salesman problem*, https://colab.research.google.com/github/Gurobi/modeling-examples/blob/master/traveling_salesman/tsp_gcl.ipynb#scrollTo=-W_zXnoRaauo, [Online; accessed 15-June-2023].

[22] K. Sundar and S. Rathinam, "Generalized multiple depot traveling salesmen problem - polyhedral study and exact algorithm," *Computers & Operations Research*, vol. 70, pp. 39–55, 2016.

[23] D. Singh, M. K. Singh, T. Singh, and R. Prasad, "Genetic algorithm for solving multiple traveling salesmen problem using a new crossover and population generation," *Computación y Sistemas*, vol. 22, Jul. 2018.

[24] A. Kiraly and J. Abonyi, "Optimization of multiple traveling salesmen problem by a novel representation based genetic algorithm," in *Intelligent Computational Optimization in Engineering: Techniques and Applications*, M. Koppen, G. Schaefer, and A. Abraham, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 241–269, ISBN: 978-3-642-21705-0.

[25] Wikipedia contributors, *Cohen's kappa — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=Cohen%27s_kappa&oldid=1130024730, [Online; accessed 15-June-2023], 2022.

[26] *ROS Task Manager*, https://dream.georgiatech-metz.fr/research-projects/ros-task-manager/, [Online; accessed 31-May-2023].

[27] F. Bolelli, S. Allegretti, L. Baraldi, and C. Grana, "Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling," *IEEE Transactions on Image Processing*, vol. PP, pp. 1–1, Oct. 2019.

[28] E. Kivelevitch, *MDMTSPV_GA - multiple depot multiple traveling salesmen problem solved by genetic algorithm*, https://www.mathworks.com/matlabcentral/fileexchange/31814-mdmtspv_ga-multiple-depot-multiple-traveling-salesmen-problem-solved-by-genetic-algorithm, [Online; accessed 2-June-2023].