

INSA Lyon - Département Informatique

Report
of the
Graduation Project

**Multi-robot navigation and control for acoustic
inspection of metal plate structures [R&D]**

***Navigation et contrôle multi-robots pour l'inspection
acoustique de structures métalliques [R&D]***

Brandon ALVES

Defended June 27, 2023

Project conducted from **January 9, 2023** to **June 30, 2023**

within the reception structure

Laboratoire CITI, équipe CHROMA (INSA & INRIA)
(Villeurbanne, France)

Référent	:	Cédric PRADALIER, Professor	GT Europe
Référent	:	Olivier SIMONIN, Professor	INSA Lyon
Tuteur	:	Mathieu MARANZANA, Lecturer	INSA Lyon

Remerciements

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 research teams of the INSA Lyon CITI-INRIA and CNRS IRL2958 GT-CNRS laboratories for their collaboration and knowledge sharing. I would 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.

Contents

1	Introduction	1
2	Bibliographic studies	2
3	Scientific and technical proposals	4
3.1	Preliminary definitions	4
3.2	Solution proposal	6
3.3	Étude théorique de propriétés de la solution proposée	14
4	Implémentations des algorithmes	16
5	Expérimentations, validations et évaluations	22
6	Bilan personnel	36
7	Conclusion et perspectives	38
	References bibliographiques	41

Annexes

A	Environnements de test	44
B	Comparaison des différentes stratégies de navigation	45
C	Résultat d’investigation	46

1 Introduction

This end-of-study project is part of the broader context of the European BugWright2 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 ultrasonic guided waves (UGW), 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.
- Fixed coordination and synchronization issues between robots.
- Implementation of navigation methods in a simulation environment.

This report presents the work carried out as part of our end-of-study project on navigation and multi-robot control for the acoustic inspection of metal structures. In the first section, we introduce the subject of the report and present the objectives of our project. The second section is dedicated to the bibliographic study, where we summarize the existing research and publications on the subject. In the third section, we propose a solution for multi-robot navigation and control for the acoustic inspection of metallic structures. This section is divided into three subsections: preliminary definitions, proposed solution and theoretical study of the properties of the proposed solution. The fourth section describes the details of the technical implementation of our proposed solution. The fifth section presents the results of our experiments, validations and evaluations. In the sixth section, we make a personal assessment of our experience working on this project. Finally, in the seventh section, we conclude our report by summarizing the results obtained, the limits of the project and the perspectives for future research.

2 Bibliographic studies

This section presents an in-depth study of relevant scientific literature in the field of multi-robot inspection. 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 ultrasonic guided waves and their applications in the field of tomography [1, 2], mapping of robots and metallic structures [3–6], robots and sensors used in our project [7], multi-robot exploration [8, 9] as well as placement strategies for detection [10–12].

The paper [1], 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 [3] 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.

The paper [4], the authors propose a method for mapping metallic structures using UGW waves. They combine a Cartesian grid with specific features for fault 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.

The paper [2], 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 guided ultrasonic waves 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 inertial measurement unit (IMU). They also incorporate a piezoelectric sensor for UGW wave 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 shown in section 3.1.

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 using UGW waves. 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 TSP (Traveling Salesman Problem) 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 4 used.

In summary, the works presented in this section are interesting for our problem, because 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.

3 Scientific and technical proposals

In this section, we present the preliminary definitions, the proposed solution and the theoretical study of the properties of the proposed solution.

3.1 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 robots of the type *crawlers*. These robots are equipped with two drive wheels and an idler wheel. An example crawler is shown in figure 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. We also assume that the robots are able to synchronize in order to be able to move simultaneously or alternatively. 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 :

- an IMU (Inertial Measurement Unit) sensor
- a UGW (Ultrasonic Guided Waves) sensor

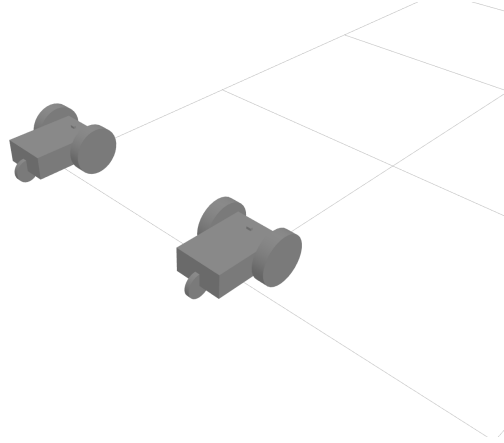


Figure 1: Crawler model used for acoustic inspection of metal structures.

- a LIDAR (Light Detection And Ranging) 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. 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.

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 status refers to areas whose status 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 status 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 and improve the efficiency of acoustic inspection.

3.2 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.

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 points given. We detail our implementation of this algorithm in section 4.

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.

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.

We present in figure 2 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 shows the first phase of vertical displacement. In order to achieve this strategy, a minimum of, $n \geq 2$ robots, 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 UGW waves. If the distance separating a robot n_a from a robot n_b , $(n_a, n_b) \in \{1, 2, \dots, n\}^2$, is less than the maximum propagation distance UGW waves, 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 , $n_k \in \{1, 2, \dots, n\}$, 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, \dots, n_{k-2} and n_{k+2}, \dots, 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

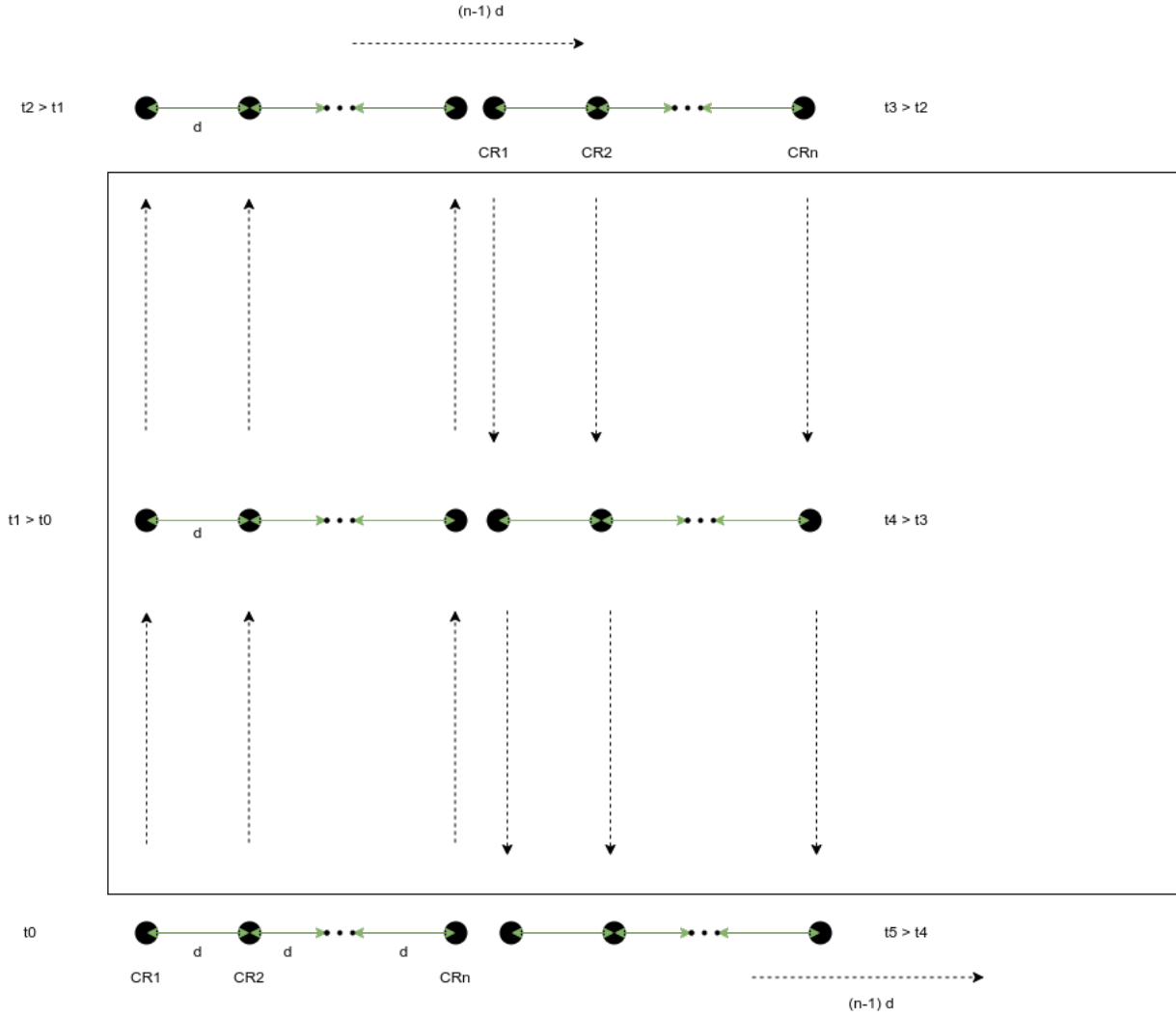


Figure 2: *Roller Painting* navigation strategy.

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 hulls 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 19.

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 hulls of the corrosion zones.

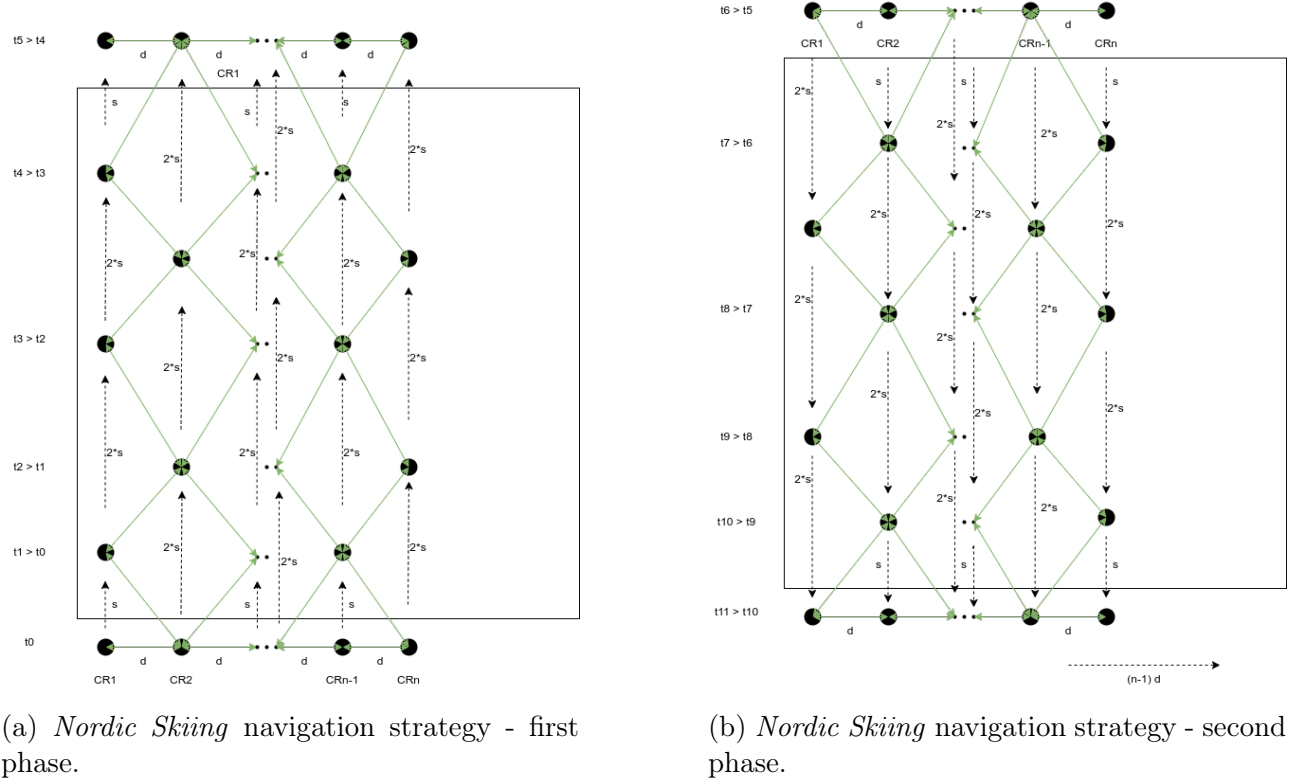


Figure 3: *Nordic Skiing* navigation strategy.

Figure 3 presents a diagram describing the navigation strategy *Nordic Skiing*. This strategy also consists of two phases, a vertical movement phase and a horizontal movement phase. Figure 3 shows the first phase of vertical movement. In order to achieve this strategy, a minimum of, $n \geq 2$ robots, 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 3a). 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 3b). 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 UGW waves. Here, as with the *Roller Painting* navigation strategy, it is not necessary for a robot n_k , $n_k \in \{1, 2, \dots, n\}$, 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 navigation strategy *Nordic Skiing*, 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 20 and figure 21.

***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 navigation strategy *Roller Painting*, the convex hull of the corrosion zones was approximated by a rectangle. This approximation is a little more precise for the navigation strategy *Nordic Skiing*. It would be interesting to have a greater degree of precision around potential areas of corrosion. This is what we propose with the navigation strategy *Polygonal Investigation*. 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.

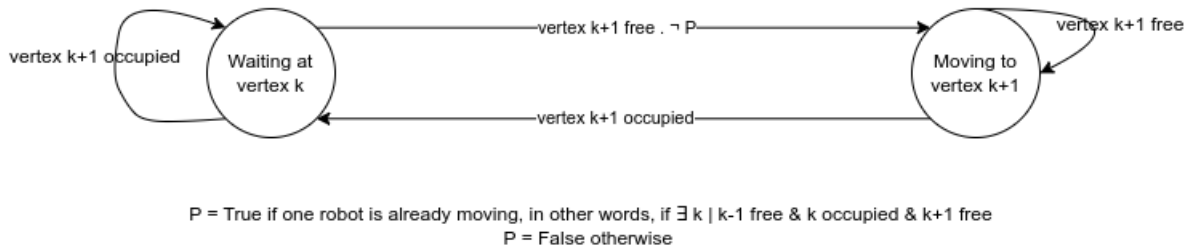


Figure 4: *Polygonal Investigation* navigation strategy.

Nous présentons sur la figure 4 un automate à états finis décrivant la stratégie de navigation *investigation polygonale*. Au début de la stratégie de navigation *investigation polygonale*, chaque

$n \in \mathbb{N}$ robots de $k \in \mathbb{N}$ équipes sont positionnés sur des sommets consécutifs d'un polygone englobant la zone potentielle de corrosion. Dans cette dernière, chaque robot a deux états. Le premier consiste à attendre et le second consiste à se déplacer en suivant la trajectoire polygonale, à savoir parcourir les différents sommets constituant le polygone. Le robot capable d'avancer, c'est-à-dire, dont le sommet suivant n'est pas occupé par un autre robot, avance. Les autres attendent jusqu'à ce que le robot qui est en train d'avancer atteigne le dernier sommet libre du polygone. Le procédé est ensuite répété pour chaque robot de chaque équipe jusqu'à ce que les sommets occupés par les robots soient les mêmes que ceux occupés au début de la stratégie de navigation *investigation polygonale*.

Nous présentons à la figure 5 un exemple des différentes phases de déplacement de la stratégie de navigation, *investigation polygonale* avec $k = 1$ équipe, $n = 2$ robots et un polygone à 3 sommets. Sur ces différentes figures, nous avons représenté une zone de corrosion par un cercle et une approximation de cette zone par une forme enveloppant le cercle. L'objectif est d'approcher le plus finement possible la zone de corrosion. La zone de corrosion n'est pas connue à l'avance, seule la forme enveloppant le cercle est connue. La figure 5a présente la phase initiale de la stratégie de navigation *investigation polygonale*. La figure 5b présente la première phase de déplacement de la stratégie de navigation *investigation polygonale*, pendant que le crawler *CR2* avance. Nous pouvons voir qu'après que le crawler *CR2* ait avancé, une partie de la zone suspectée comme étant de la corrosion est éliminée et considérée comme saine, comme nous pouvons le voir à la figure 5c. Les crawlers continuent de se déplacer jusqu'à atteindre leur position initiale, comme nous pouvons le voir à la figure 5g.

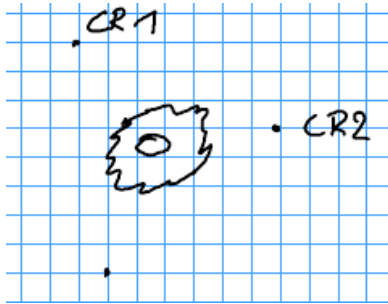
Definition 1 (Zone fantôme). Une zone fantôme est une zone de corrosion détectée par une des stratégies de navigation, mais qui n'est pas une zone de corrosion. Il s'agit d'un faux positif.

La stratégie de navigation *investigation polygonale* a deux avantages. Le premier est qu'elle permet de rapidement éliminer les zones fantômes 1. Le second est qu'elle permet d'approcher les enveloppes convexes des zones de corrosion par des formes plus diverses et précises que des rectangles du fait de la grande variation de l'orientation des rayons du signal émis et reçu par les robots autour de chaque sommet du polygone.

Cette stratégie nécessite deux étapes préalables à son exécution :

1. l'extraction des zones de corrosion détectées par une des stratégies de navigation précédentes.
2. la détermination de l'ordre d'investigation des zones de corrosion.

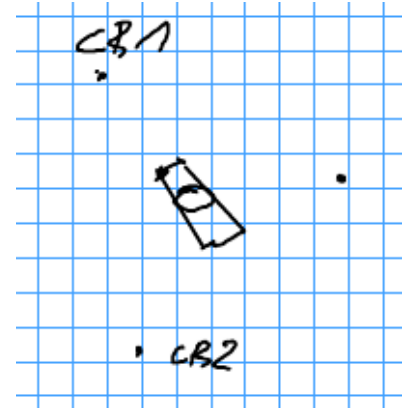
La première étape peut être résolue en utilisant un algorithme de décomposition de graphe en composantes fortement connexes. Une composante fortement connexe est définie à la définition 2. Nous considérons alors notre grille d'occupation, résultante de l'exploration d'une des deux stratégies définies préalablement, comme un graphe non orienté $G = (V, E)$, où V est l'ensemble des sommets du graphe, correspondant aux cellules de la grille d'occupation et E sont les arêtes du graphe, correspond aux cellules adjacentes. Ce problème est bien connu



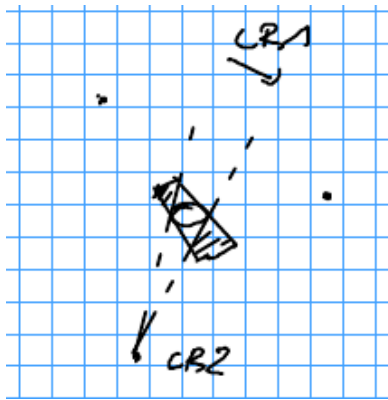
(a) Phase initiale.



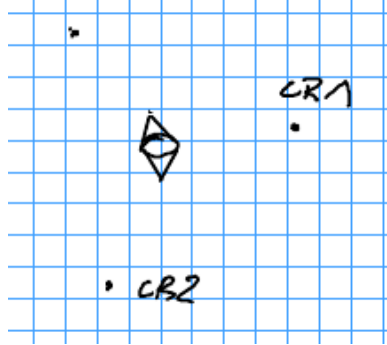
(b) Première phase de déplacement.



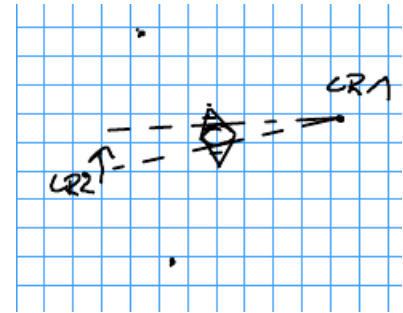
(c) Seconde phase.



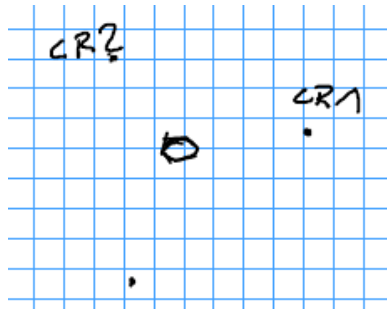
(d) Seconde phase de déplacement.



(e) Troisième phase.



(f) Troisième phase de déplacement.



(g) Dernière phase.

Figure 5: Phases de déplacement de la stratégie de navigation *investigation polygonale*.

et il existe des algorithmes simples pour les résoudre, comme l'algorithme de Tarjan [19], de complexité temporelle linéaire $O(|V| + |E|)$. Nous ne nous pencherons pas plus sur ce problème et confions sa résolution à la bibliothèque *OpenCV*.

Definition 2 (Composante fortement connexe (SCC)). Une composante fortement connexe d'un graphe $G = (V, E)$ est un sous-ensemble C de V tel que pour tout couple de sommets $(u, v) \in C^2$, il existe un chemin de u à v et un chemin de v à u .

Definition 3 (Cycle hamiltonien). Un cycle hamiltonien est un cycle passant par tous les sommets d'un graphe, une et une seule fois.

Definition 4 (Problème du voyageur de commerce (TSP)). Étant donné un graphe $G = (V, E)$, où V est l'ensemble des sommets du graphe et E est l'ensemble des arêtes du graphe, et une fonction de coût $c : E \rightarrow \mathbb{R}$, le problème du voyageur de commerce consiste à trouver un cycle hamiltonien 3 de coût minimal dans G .

Definition 5 (Problème du voyageur de commerce multiple avec multiples dépôts (mTSP)). Étant donné un graphe $G = (V, E)$, où V est l'ensemble des sommets du graphe et E est l'ensemble des arêtes du graphe, une fonction de coût $c : E \rightarrow \mathbb{R}$, et un ensemble de dépôts $D \subset V$, le problème du voyageur de commerce multiple avec multiples dépôts consiste à trouver un ensemble de cycles de coût total minimal dans G , chacun passant par un et un seul dépôt.

La seconde étape peut être résolue en utilisant un algorithme de TSP 4 (*Travelling Salesman Problem*) dans le cas où le nombre d'équipes k est égale à 1 et un algorithme de mTSP 5 (*(multiple depot) multiple Travelling Salesman Problem*) dans le cas où le nombre d'équipes k est supérieur strictement à 1. Il existe plusieurs paradigmes de résolution pour résoudre ce type de problème. Un premier est de trouver une solution exacte en utilisant un algorithme de programmation linéaire en nombres entiers. Un second est de trouver une solution approchée en utilisant une méta-heuristique.

Definition 6 (Classe NP). La classe NP est la classe des problèmes de décision qui peuvent être résolus par un algorithme non déterministe en temps polynomial.

Definition 7 (Problème NP-difficile). Un problème est NP-difficile s'il est au moins aussi difficile que les problèmes de la classe NP. En d'autres termes, un problème est NP-difficile s'il existe un algorithme de réduction polynomiale qui transforme un problème de la classe NP en une instance de ce problème.

Definition 8 (Problème NP-complet). Un problème est NP-complet s'il est à la fois NP et NP-difficile.

Le problème du voyageur de commerce est un problème NP-complet 8. Il peut être traité comme un problème d'optimisation linéaire en nombres entiers [20, 21]. Pour ce faire, nous

utilisons la formulation présentée à l'équation 1.

$$\begin{aligned}
& \text{minimiser} && \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \\
& \text{soumis à} && \sum_{i \in V} x_{ij} = 1 && \forall j \in V \\
& && \sum_{j \in V} x_{ij} = 1 && \forall i \in V \\
& && \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 && \forall S \subset V, 2 \leq |S| \leq |V| - 1 \\
& && x_{ij} \in \{0, 1\} && \forall i \in V, \forall j \in V
\end{aligned} \tag{1}$$

La fonction objective à minimiser de la formulation 1 est la somme des distances entre chaque paire de villes. Les deux premières contraintes assurent que chaque ville est visitée exactement une fois. La troisième contrainte assure que le cycle formé par les villes visitées est simple, c'est-à-dire, qu'il ne contient pas de sous-cycles. La dernière contrainte assure que les variables de décision x_{ij} sont binaires, avec $x_{ij} = 1$ si le robot se déplace de la ville i à la ville j et $x_{ij} = 0$ sinon.

Le problème du voyageur de commerce multiple avec multiples dépôts est un problème NP-difficile [22]. Celui-ci peut être résolu en utilisant une méta-heuristique comme un algorithme génétique [23, 24]

Dans les prochaines sections, nous détaillerons chaque stratégie de navigation, en exposant les algorithmes et les mécanismes spécifiques utilisés pour mettre en œuvre notre proposition de solution. Nous analyserons également les performances et les résultats obtenus à travers des expérimentations et des évaluations approfondies.

3.3 Étude théorique de propriétés de la solution proposée

Stratégie de navigation *ski nordique*

Proposition 1. *L'angle du signal émis et reçu par les robots, pour la stratégie de navigation ski nordique, varie entre $-\tan^{-1}(\frac{s}{d})$ et $\tan^{-1}(\frac{s}{d})$.*

Nous pouvons observer le résultat de la proposition 1, en nous appuyant sur les propriétés de trigonométrie et la définition de la fonction tangente. Nous explicitons sur la figure 6 la démarche entreprise pour trouver $\alpha = -\tan^{-1}(\frac{s}{d})$. Cette proposition nous permet de quantifier la portée de l'orientation du signal émis et reçu par les robots.

Stratégie de navigation *investigation polygonale*

Proposition 2. *La stratégie de navigation investigation polygonale, définie par un polygone, P permet de couvrir tous points à l'intérieur du polygone P .*

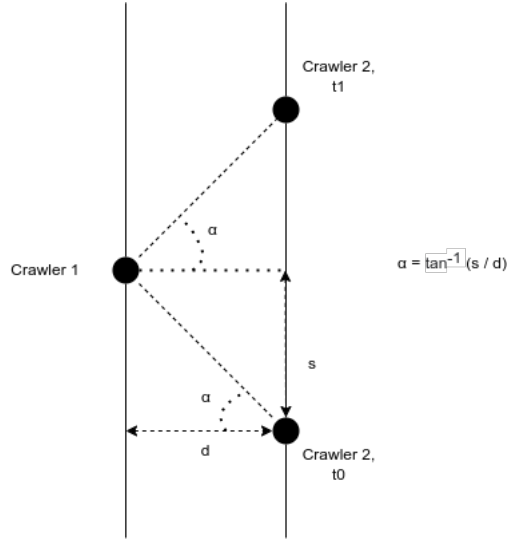


Figure 6: Orientation du signal émis et reçu pour la stratégie de navigation *ski nordique*.

Proof. Démontrons la proposition 2.

- Soit P un polygone convexe à p sommets utilisé pour la stratégie de navigation *investigation polygonale*.
- Par simplicité, nous considérons une stratégie à 2 robots, mais la preuve reste semblablement la même pour $n > 2$ robots.
- Considérons le robot r_1 au sommet s_1 du polygone P et le robot r_2 au sommet s_2 du polygone P .
- Soit un point p à l'intérieur du polygone P .
- Alors, il existe un point p' tel que p est sur le segment $[s_1, p']$ et p' est sur le bord du polygone P , par définition de la convexité de P .
- Par définition de la stratégie de navigation, *investigation polygonale*, le robot r_2 se déplace sur les contours du polygone P et donc en particulier sur le point p' .
- On a donc pour tout point p à l'intérieur du polygone P , il existe un couple de positions pour les robots r_1 et r_2 tel que p se trouve sur le segment formé par les points où se trouvent les robots r_1 et r_2 .
- Donc tous les points à l'intérieur du polygone P sont couverts par la stratégie de navigation *investigation polygonale*.

□

Proposition 3. *L'approximation de l'enveloppe convexe des zones de corrosion à l'issue de l'investigation polygonale, avec un polygone à p sommets, $p \in \mathbb{N}$, est un polygone d'au plus $2p$ sommets.*

Proof. Donnons une intuition de la preuve de la proposition 3.

- On a, pour chaque sommet du polygone, il existe deux droites passant par ce point et touchant la zone de corrosion sans la traverser.
- On a donc, pour un sommet du polygone, au plus deux droites qui participent à la construction de l'approximation de l'enveloppe convexe et donc, dont un segment de ces droites est une arête de l'approximation de l'enveloppe convexe.
- On a donc, pour un sommet du polygone, au plus deux arêtes de l'approximation de l'enveloppe convexe.
- On a donc, pour un polygone à p sommets, au plus $2p$ arêtes de l'approximation de l'enveloppe convexe.
- On a donc, pour un polygone à p sommets, au plus $2p$ sommets de l'approximation de l'enveloppe convexe.

□

Conjecture 1. *Si le polygone utilisé lors de l'investigation polygonale est un cercle, alors l'approximation de l'enveloppe convexe de la zone de corrosion à l'issue de l'investigation polygonale est l'enveloppe convexe.*

4 Implémentations des algorithmes

Dans cette section, nous mettons en évidence certaines des différentes implémentations techniques que nous avons développées pour soutenir nos solutions de navigation et de contrôle multi-robots dans le contexte de l'inspection acoustique de structures métalliques. Nous commençons par décrire notre adaptation de l'algorithme de tracé de segment de Bresenham [18], largement utilisé pour déterminer quels sont les points d'un plan discret qui doivent être tracés afin de former une approximation de segment de droite entre deux points donnés. Ensuite, nous abordons l'implémentation de l'algorithme *peinture au rouleau*, qui permet aux robots de se déplacer de manière simultanée, en suivant des trajectoires parallèles. Nous poursuivons avec l'implémentation de l'algorithme *ski nordique*, qui permet aux robots de se déplacer de manière alternée, en suivant des trajectoires parallèles, modifiant ainsi l'orientation du vecteur représentant la direction de déplacement de l'onde émise et reçue par la paire de robot. De plus, nous examinons l'implémentation de l'algorithme *investigation polygonale*, qui permet aux robots d'examiner plus précisément des zones suspectes de corrosion. Enfin, nous présentons

l'algorithme de calcul du κ de Cohen [25], utilisé pour évaluer la qualité et la fiabilité des résultats de l'inspection acoustique. Nous discutons en détail de notre implémentation de cet algorithme, qui fournit des mesures quantitatives pour évaluer la performance des robots dans l'inspection des structures métalliques. Chacune de ces implémentations techniques contribue à l'efficacité et à la précision de notre approche de navigation et de contrôle multi-robots, et sera examinée en détail dans les sous-sections suivantes.

Algorithme de tracé de segment de Bresenham

Nous utilisons l'algorithme de tracé de segment de Bresenham pour déterminer les points du segment de droite entre les deux robots. L'algorithme est présenté à l'algorithme 1. La partie adaptée à notre problème se trouve entre les lignes 12 et 17 de ce dernier. À cet endroit, nous vérifions si la puissance du signal est suffisamment altérée et si le point du segment de droite entre les deux robots n'a pas déjà été perçu comme étant libre de corrosion. Si c'est le cas, alors le point considéré est marqué comme étant de la corrosion, modélisé par la valeur `OCCUPIED`. Si la puissance du signal n'est pas suffisamment altérée, alors le point considéré est marqué comme étant libre de corrosion, modélisé par la valeur `EMPTY`. Une fois que tous les points du segment ont été parcourus, la grille G est mise à jour avec les nouvelles informations. L'algorithme de tracé de segment de Bresenham contribue ainsi à la construction de la grille d'occupation qui permet de localiser les zones de corrosion détectées par les robots lors de l'inspection acoustique des structures métalliques.

Algorithme *peinture au rouleau* et *ski nordique*

Nous présentons dans cette sous-section les implémentations des algorithmes *peinture au rouleau* et *ski nordique*. Leur code source est disponible sur GitLab, ici¹ et ici².

Les implémentations de ces algorithmes, ont été réalisées en utilisant le langage de programmation Python et les bibliothèques ROS (Robot Operating System). Dans ces implémentations, nous utilisons le framework ROS Task Manager [26] pour gérer les tâches des robots inspecteurs. Tout d'abord, nous initialisons le nœud ROS et créons un client de tâches. Ensuite, nous récupérons les paramètres nécessaires tels que la vitesse des crawlers, l'identifiant du crawler, la distance entre les crawlers, le chevauchement ou encore les dimensions de la surface à inspecter.

Les algorithmes sont ensuite exécutés en suivant une séquence de mouvements précis. Pour chaque crawler, nous définissons des trajectoires verticales et horizontales en utilisant une boucle itérative et des calculs mathématiques. Les crawlers se déplacent le long des trajectoires définies,

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

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

Algorithm 1: Processus de mise à jour de la grille d'occupation à l'aide de l'algorithme de tracé de segment de Bresenham.

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.

```

1  $p_0 \leftarrow \text{from\_position\_to\_grid\_coordinate}(P_1)$ 
2  $p_1 \leftarrow \text{from\_position\_to\_grid\_coordinate}(P_2)$ 
3 if  $\text{is\_out\_of\_grid}(p_0)$  or  $\text{is\_out\_of\_grid}(p_1)$  then
4   | return
5 end
6  $dx \leftarrow p_1.x - p_0.x$ 
7  $dy \leftarrow p_1.y - p_0.y$ 
8  $sx \leftarrow \text{sign}(dx)$ 
9  $sy \leftarrow \text{sign}(dy)$ 
10  $err = dx - dy$ 
11 while  $p_0 \neq p_1$  do
12   | if  $pwd \leq threshold$  and  $G(p_0) = \text{UNKNOWN}$  then
13     |  $G(p_0) \leftarrow \text{OCCUPIED}$ 
14   | end
15   | else if  $pwd > threshold$  then
16     |  $G(p_0) \leftarrow \text{EMPTY}$ 
17   | end
18    $e2 \leftarrow 2 \times err$ 
19   if  $e2 > -dy$  then
20     |  $err \leftarrow err - dy$ 
21     |  $p_0.x \leftarrow p_0.x + sx$ 
22   end
23   if  $e2 < dx$  then
24     |  $err \leftarrow err + dx$ 
25     |  $p_0.y \leftarrow p_0.y + sy$ 
26   end
27 end

```

en utilisant les fonctions du client de tâches telles que `AlignWithTarget` et `FollowLine` pour maintenir une trajectoire précise.

Pendant l'exécution des algorithmes, les crawlers se synchronisent en utilisant les fonctions

`SetStatusSync` et `WaitForStatusSync` du client de tâches. Cela garantit que les crawlers exécutent les mouvements de manière coordonnée et se positionnent correctement pour couvrir toute la surface métallique. À la fin de chaque étape de mouvement, le statut est mis à jour et la synchronisation est effectuée avec le partenaire correspondant.

L’implémentation des deux algorithmes *peinture au rouleau* et *ski nordique* permettent aux crawlers d’explorer la surface métallique de manière méthodique et complète. En utilisant des trajectoires verticales et horizontales, les crawlers parcourent la surface en chevauchant les zones précédemment inspectées pour s’assurer d’une couverture optimale. Ici, nous avons utilisé un chevauchement de 10 cm entre les différentes trajectoires verticales et horizontales.

Une fois les algorithmes terminés, le temps d’exécution est enregistré, fournissant une indication de la durée nécessaire pour inspecter la surface métallique. La grille d’occupation est également enregistrée afin de calculer le score de l’inspection. Cette implémentation constitue une étape essentielle dans notre proposition de solution pour l’inspection acoustique de structures métalliques et permet de garantir une couverture complète et efficace de la surface à inspecter.

Algorithme *investigation polygonale*

Nous présentons dans cette sous-section l’implémentation de l’algorithme *investigation polygonale*. Le code source correspondant est disponible sur GitLab ³.

L’implémentation de cet algorithme, a également été réalisée en utilisant le langage de programmation Python et les bibliothèques ROS (Robot Operating System). Dans cette implémentation, nous utilisons toujours le framework ROS Task Manager [26] pour gérer les tâches des robots inspecteurs. Premièrement, nous initialisons le nœud ROS et récupérons les différents paramètres et plus particulièrement la carte des zones potentielles de corrosion, sur laquelle nous nous basons pour l’inspection, qui sont issues d’une des deux stratégies grossières. Ensuite, nous extrayons les composantes fortement connexes de la carte en utilisant la fonction `connectedComponentsWithStats` de la bibliothèque *OpenCV*. Cette fonction utilise l’algorithme spaghetti de Bolleli [27] pour extraire les composantes fortement connexes d’une image. Pour chacune de ces composantes, nous récupérons son centre et ses dimensions. Ensuite, nous construisons un polygone à $p \in \mathbb{N}$ côtés autour de chaque centre d’une composante. Pour ce faire, nous plaçons p points sur une ellipse centrée sur le centre de la composante et dont les axes sont les dimensions de la composante. Nous avons donc pour chaque zone potentielle de corrosion un polygone à p côtés qui l’entoure. Il ne reste plus qu’à trouver le plus court chemin qui passe par tous les polygones. Pour cela, nous utilisons la bibliothèque *Gurobi* pour résoudre un simple TSP dans le cas où le nombre d’équipes de robots $k = 1$. Lorsque $k > 1$, nous utilisons l’algorithme génétique proposé par Elad Kivelevitch [28] pour résoudre le problème du mTSP avec plusieurs dépôts.

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

Une fois les algorithmes terminés, le temps d'exécution est enregistré, fournissant une indication de la durée nécessaire pour inspecter les différentes zones potentielles de corrosion. La grille d'occupation est également enregistrée afin de calculer le score de l'inspection. Cette implémentation constitue une étape essentielle dans notre proposition de solution pour l'inspection acoustique de structures métalliques et permet d'investiguer les zones potentielles de corrosion de manière efficace.

Algorithme de calcul du κ de Cohen

L'évaluation de la qualité et de la fiabilité des résultats de l'inspection acoustique est essentielle pour garantir des mesures précises de l'état des structures métalliques. Dans cette sous-section, nous présentons l'algorithme du calcul du κ de Cohen [25], présenté à l'algorithme 2, une mesure statistique couramment utilisée pour évaluer l'accord entre les résultats obtenus par les robots et une référence humaine.

κ	Interprétation
< 0	Désaccord
$0.00 - 0.20$	Accord très faible
$0.21 - 0.40$	Accord faible
$0.41 - 0.60$	Accord modéré
$0.61 - 0.80$	Accord fort
$0.81 - 1.00$	Accord presque parfait

Table 1: Interprétation du κ de Cohen selon Landis et Koch.

L'algorithme de calcul du κ de Cohen se base sur la notion de concordance et de discordance entre les résultats des inspections réalisées par les robots et celles réalisées par des inspecteurs humains (vérité de terrain). Il prend en compte les résultats positifs, négatifs, faux positifs et faux négatifs obtenus lors de l'inspection acoustique. Ces informations sont utilisées pour calculer la valeur du coefficient de Cohen, noté κ , avec $\kappa = \frac{p_o - p_e}{1 - p_e}$, où p_o est le taux d'accord observé et p_e le taux d'accord attendu.

L'algorithme se déroule en plusieurs étapes. Tout d'abord, les résultats des inspections réalisées par les robots et les réelles répartitions des zones de corrosion sont comparés pour chaque zone inspectée. Pour ce faire, nous comparons les valeurs de chaque cellule de la grille d'occupation, obtenue à l'issue de l'inspection par les robots, avec celles de la vérité terrain. Ayant modélisé les différents environnements de tests, nous connaissons la véritable répartition des zones de corrosion. Ensuite, les résultats sont regroupés en quatre catégories : concordance positive, concordance négative, discordance positive (faux positifs) et discordance négative (faux négatifs). Ces catégories sont utilisées pour calculer les taux d'observation et d'accord observés entre les robots et la véritable répartition des zones de corrosion. Le κ de Cohen est ensuite calculé à partir des taux d'observation et d'accord observés, prenant en compte la possibilité de

Algorithm 2: Algorithme du κ de Cohen.

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: $\kappa \in [0, 1]$

```
1  $TP \leftarrow 0$ 
2  $TN \leftarrow 0$ 
3  $FP \leftarrow 0$ 
4  $FN \leftarrow 0$ 
5 for  $i \leftarrow 1$  to  $l$  do
6   for  $j \leftarrow 1$  to  $w$  do
7     if  $is\_label\_1(I_0(i, j))$  then
8       if  $is\_label\_1(I(i, j))$  then
9          $TP \leftarrow TP + 1$ 
10      end
11      else
12         $FN \leftarrow FN + 1$ 
13      end
14    end
15    else
16      if  $is\_label\_1(I(i, j))$  then
17         $FP \leftarrow FP + 1$ 
18      end
19      else
20         $TN \leftarrow TN + 1$ 
21      end
22    end
23  end
24 end
25  $f_c \leftarrow \frac{(TN+FN)(TN+FP)+(FP+TP)(FN+TP)}{TP+TN+FN+FP}$ 
26  $\kappa \leftarrow \frac{TP+TN-f_c}{TP+TN+FN+FP-f_c}$ 
```

concordance due au hasard. Plus le κ de Cohen se rapproche de 1, plus il y a un accord élevé entre les résultats des robots et ceux de la vérité terrain. En revanche, un κ proche de 0 indique un faible niveau d'accord, tandis qu'un κ négatif suggère une discordance entre les résultats. Une Interprétation du κ de Cohen selon Landis et Koch est présentée dans le tableau 1.

Nous avons implémenté cet algorithme dans le cadre de notre projet, en utilisant les résultats des inspections acoustiques effectuées par les robots et des cartes composées de zones de corrosion comme base de comparaison. Cette implémentation nous permet d'obtenir des mesures quantitatives pour évaluer la performance de notre approche de navigation et de con-

trôle multi-robots dans l’inspection des structures métalliques. Dans les prochaines sections, nous détaillerons les résultats obtenus grâce à l’application de cet algorithme du calcul du κ de Cohen.

5 Expérimentations, validations et évaluations

Dans cette section, nous présentons les expérimentations que nous avons menées pour valider et évaluer nos différentes stratégies de navigation et de contrôle multi-robots dans le contexte de l’inspection acoustique de structures métalliques. Ces expérimentations visent à démontrer l’efficacité, la précision et la fiabilité de notre système dans la détection et la localisation des zones de corrosion.

Pour mener à bien ces étapes, nous avons choisi d’effectuer nos expériences en utilisant *Gazebo*, un environnement de simulation bien établi dans le domaine de la robotique. Nous avons commencé par construire plusieurs cartes de tests. Ces cartes modélisent une surface plane sur laquelle sont placées des formes géométriques simples, des rectangles et des cercles, et des formes plus complexes, des polygones entre 3 et 8 sommets. Ces différentes formes géométriques représentent les zones de corrosion que nous souhaitons détecter et localiser. Nous présentons en annexe A, sur la figure 17 les cartes que nous avons construites pour nos expérimentations. Chacune de ces cartes est de taille 6 mètres par 6 mètres. Le nombre de zones de corrosion varie entre 5, 8, 11, 15, 20 et 30 zones. La taille et l’emplacement des zones de corrosion sont générés aléatoirement. Pour les cartes de 5, 8, 11 et 15 zones, nous avons généré 5 cartes différentes afin d’avoir des résultats plus représentatifs. Nous ne nous sommes pas permis de générer plusieurs cartes pour les cartes de 20 et 30 zones, le temps d’investigation polygonale étant trop conséquent.

Nous avons également simulé le capteur UGW en exploitant la simulation d’un capteur UWB. Ce capteur UWB permet d’émettre une impulsion et de la recevoir. En mesurant la puissance du signal, nous sommes en mesure de savoir si le signal a traversé un objet ou non. Le comportement de ce capteur UWB est donc similaire à celui du capteur UGW, à savoir qu’il permet de détecter la présence d’un objet entre deux points, mais pas de le localiser.

Nous avons évalué les performances des trois stratégies de navigation en termes de κ de Cohen et de temps d’inspection. Pour la stratégie *peinture au rouleau* et *ski nordique*, nous avons uniquement utilisé deux robots. Pour ces deux stratégies, nous avons fait varier la distance d entre les robots. Pour la stratégie *ski nordique*, nous avons également fait varier le pas s entre les robots. Pour la stratégie *investigation polygonale*, nous faisons varier le nombre de robots n , le nombre d’équipes k et le nombre de côtés p des polygones utilisés. Nous utilisons le résultat de la stratégie de navigation *peinture au rouleau* comme point de départ. Nous justifions ce choix par le fait que cette stratégie est la plus rapide et la moins précise et donc, la plus susceptible de bénéficier d’une amélioration de la part de la stratégie *investigation polygonale*, sans atteindre des temps d’inspection trop longs. Nous faisons donc varier le paramètre d de cette stratégie. Nous résumons les paramètres expérimentaux utilisés pour chaque stratégie dans le tableau 2.

Stratégie	Paramètre	Valeurs
<i>peinture au rouleau</i>	n	2
	d	1, 2, 3, 6 (mètres)
<i>ski nordique</i>	n	2
	d	1, 2, 3, 6 (mètres)
	s	1, 2, 3, 6 (mètres)
<i>investigation polygonale</i>	stratégie initiale	<i>peinture au rouleau</i>
	d	1, 2, 3, 6 (mètres)
	n	2, 4
	k	1, 2
	p	4, 6

Table 2: Paramètres expérimentaux utilisés pour chaque stratégie de navigation.

Au cours de ces simulations, nous nous attendons à avoir certains résultats. Parmi eux, nous nous attendons à ce que la stratégie *peinture au rouleau* soit la plus rapide, mais également la moins précise. À l'inverse, nous nous attendons à ce que la stratégie *investigation polygonale* soit la plus précise, mais également la plus lente. Nous nous attendons également à ce que le paramètre d ait un impact sur la précision et le temps d'inspection des stratégies *peinture au rouleau* et *ski nordique*. Une distance d faible devrait permettre d'obtenir une meilleure précision, mais devrait également augmenter le temps d'inspection. De plus, nous nous attendons à ce que le paramètre s ait également un impact sur la précision et le temps d'inspection de la stratégie *ski nordique*. Une distance s faible devrait permettre d'obtenir une meilleure précision, mais devrait également augmenter le temps d'inspection. Nous nous attendons également à ce que le paramètre p ait un impact sur la précision et le temps d'inspection de la stratégie *investigation polygonale*. Un nombre de côtés p faible devrait permettre d'obtenir une meilleure précision, mais devrait également augmenter le temps d'inspection. Ensuite, nous nous attendons à ce que les paramètres k et n aient un impact sur le temps d'inspection de la stratégie *investigation polygonale*. Un nombre d'équipes k ou un nombre de robots n élevé devrait permettre d'obtenir un temps d'inspection plus faible. Enfin, nous nous attendons à ce que le nombre de zones de corrosion ait un impact sur le temps d'inspection de la stratégie *investigation polygonale*, mais pas sur les stratégies *peinture au rouleau* et *ski nordique*. Plus le nombre de zones de corrosion est élevé, plus le temps d'inspection devrait être élevé pour la stratégie *investigation polygonale*. Finalement, nous nous attendons à ce que le nombre de zones de corrosion ait un impact sur la précision des différentes stratégies. Plus le nombre de zones de corrosion est élevé, moins la précision devrait être élevée. En effet, plus le nombre de zones de corrosion est élevé, plus la probabilité d'apparition de zones fantômes devrait être élevée pour les stratégies *peinture au rouleau* et *ski nordique*. Pour la stratégie *investigation polygonale*, plus le nombre de zones de corrosion est élevé, et plus la probabilité que deux zones distinctes de corrosion aient été confondues en une seule lors des stratégies *peinture au rouleau*

ou *ski nordique*.

Les différents résultats issus des différentes simulations effectuées sont disponibles en annexe C. Sur ces images, il est possible de voir en noir les zones réelles de corrosion et en bleu les zones détectées comme comportant de la corrosion par les différents algorithmes de navigation.

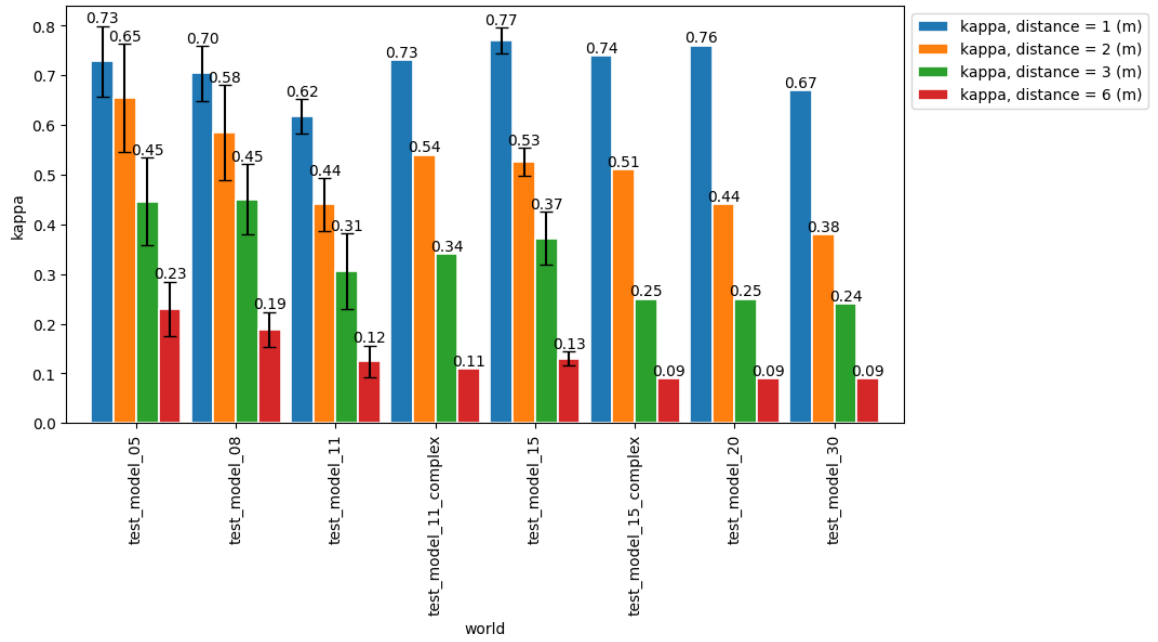
Stratégie de navigation *peinture au rouleau*

Nous résumons sur la figure 7a l'évolution du score de Cohen en fonction de la densité du monde pour chaque valeur de d . Nous résumons également sur la figure 7b l'évolution du temps d'inspection en fonction de la densité du monde pour chaque valeur de d .

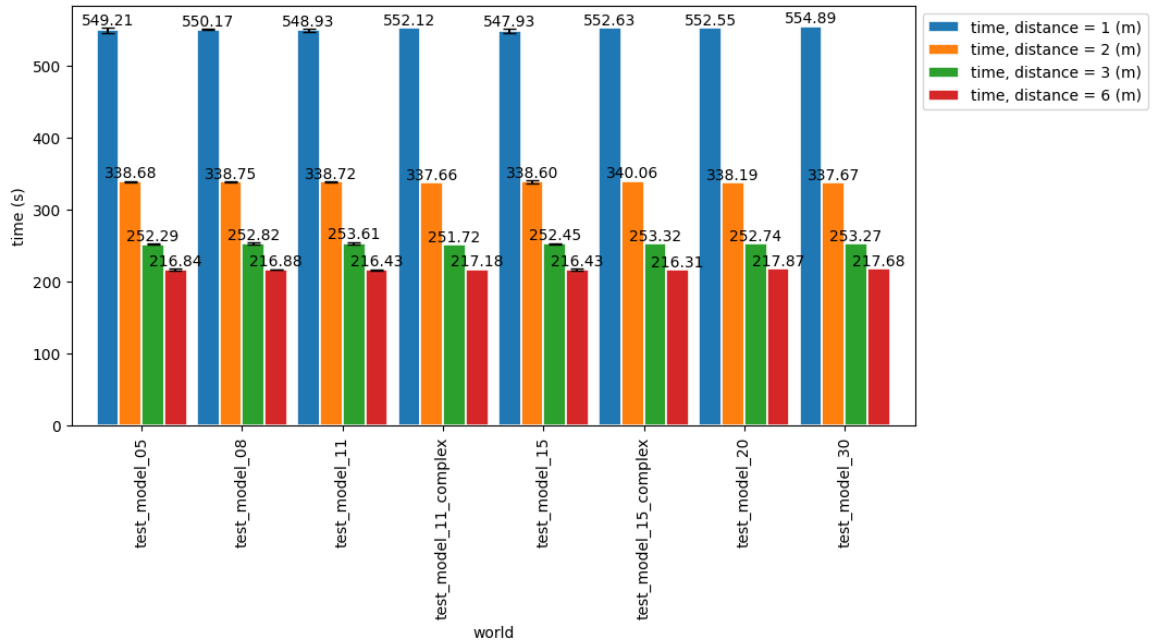
Premièrement, nous pouvons observer que le score de Cohen diminue, de manière générale, avec le nombre de zones de corrosion. Il existe des exceptions, notamment pour la carte composée de 15 zones de corrosion, où le score de Cohen est plus élevé que pour les cartes composées de 5, 8 et 11 zones de corrosion. Cela s'explique du fait que dans les cartes composées de 5, 8 et 11 zones de corrosion, nous avons introduit des zones de corrosion de formes allongées contrairement à la carte composée de 15 zones de corrosion où les zones de corrosion sont toutes des cercles. En effet, les zones de corrosion de formes allongées ont une probabilité plus grande de faire apparaître des zones fantômes, illustrées sur la figure 9, que les zones de corrosion de forme circulaire. Ces zones fantômes sont des zones libres de corrosion qui sont détectées par les crawlers. Il s'agit donc de faux positifs qui diminuent le score de Cohen. Ces zones fantômes ont également plus de chance d'apparaître lorsque la densité du monde est élevée et que, donc, les zones de corrosion sont plus proches les unes des autres, ou encore lorsque la distance d entre les deux crawlers est élevée. C'est bien ce que nous pouvons observer sur la figure 8a où le score de Cohen diminue lorsque la distance d entre les deux crawlers augmente. Nous observons qu'il semble exister une relation linéaire entre le score de Cohen et la distance d entre les deux crawlers.

Ensuite, nous observons que le temps d'exécution de l'algorithme *peinture au rouleau* est constant pour chaque valeur du nombre de zones de corrosion. Cela était attendu du fait que l'algorithme en question est un algorithme *a priori* et ne dépend donc pas du nombre de zones de corrosion. En revanche, le temps d'exécution dépend de la distance d entre les deux crawlers. Comme nous pouvons observer sur la figure 8b, le temps d'exécution augmente lorsque la distance d entre les deux crawlers diminue. Cela s'explique du fait que plus la distance d est grande, moins les crawlers doivent effectuer de déplacements pour couvrir la carte. Il ne semble pas exister de relation linéaire entre le temps d'exécution et la distance d entre les deux crawlers. Or, nous nous serions attendus à ce qu'il existe une relation linéaire entre le temps d'exécution et la distance d entre les deux crawlers. Il serait intéressant de vérifier s'il n'y a pas eu de biais introduit lors de l'implémentation de l'algorithme.

Nous avons également introduit deux cartes avec des formes plus complexes que les cartes de base. Celles-ci sont visibles en annexe A, sur les figures 17g et 17h. Malheureusement, nous n'avons pas pu, par souci de temps, faire varier la position des zones de corrosion, comme nous l'avons fait avec les cartes de faible densité. Néanmoins, il semble ne pas exister de différence

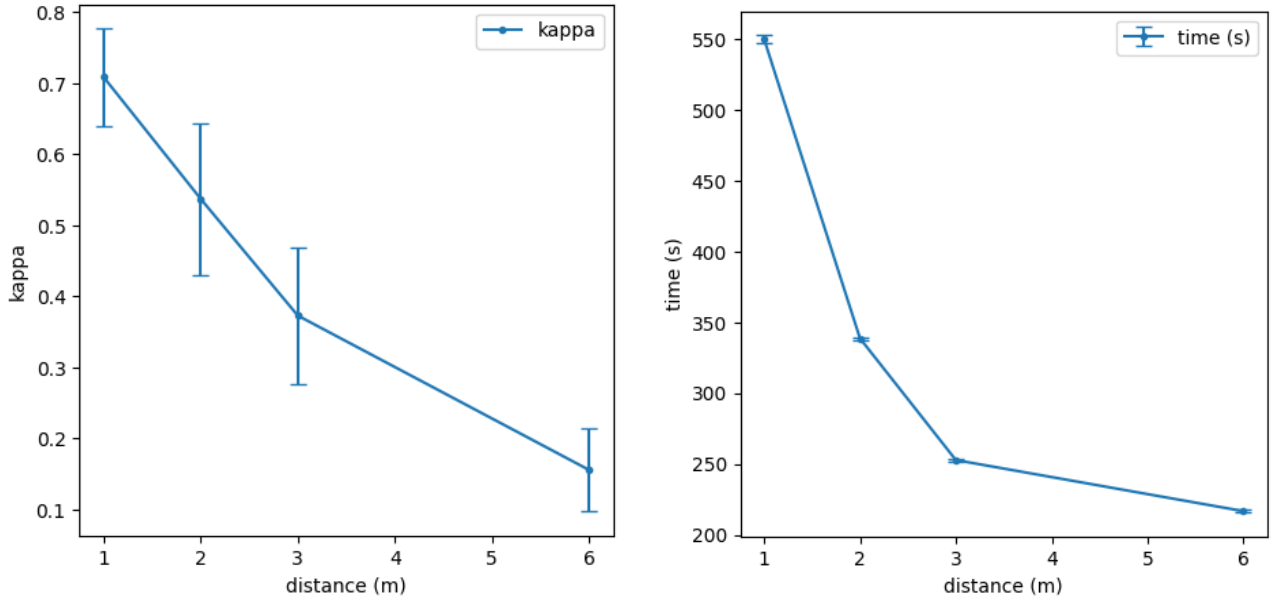


(a) κ en fonction de la densité du monde



(b) Temps d'exécution en fonction de la densité du monde

Figure 7: Évolution du κ de Cohen et du temps d'exécution de l'algorithme *peinture au rouleau* en fonction de la densité du monde et de la distance entre les robots.



(a) κ en fonction de la distance entre les deux crawlers (b) Temps d'exécution en fonction de la distance entre les deux crawlers

Figure 8: Évolution du κ de Cohen et du temps d'exécution de l'algorithme *peinture au rouleau* en fonction de la distance qui sépare les deux crawlers.

significative entre les cartes de formes complexes et les cartes de formes simples. Par exemple, pour les cartes avec 15 formes de corrosion et la carte avec 15 formes de corrosion complexes, le score de Cohen ne varie que de 0.02 en moyenne pour une distance $d = 1$ et de 0.04 en moyenne pour une distance $d = 6$.

Dans la suite de ce rapport, nous considérerons une distance $d = 3$ mètres entre les deux crawlers pour l'algorithme *peinture au rouleau*.

Stratégie de navigation *ski nordique*

Nous allons maintenant analyser les résultats obtenus pour l'algorithme *ski nordique*. Comme pour l'algorithme *peinture au rouleau*, nous avons fait varier la densité du monde et la distance d entre les deux crawlers, mais également le pas s utilisé entre les deux crawlers. La figure 10 présente l'évolution du score de Cohen et du temps d'exécution de l'algorithme *ski nordique* en fonction de la densité du monde pour différentes valeurs de la distance d entre les deux crawlers et un pas $s = 3$ mètres.

Nous avons des résultats très similaires à ceux obtenus pour l'algorithme *peinture au rouleau*. En effet, nous observons sur la figure 10a que le score de Cohen diminue de manière générale lorsque la densité du monde augmente. De plus, le temps d'exécution de l'algorithme *ski*

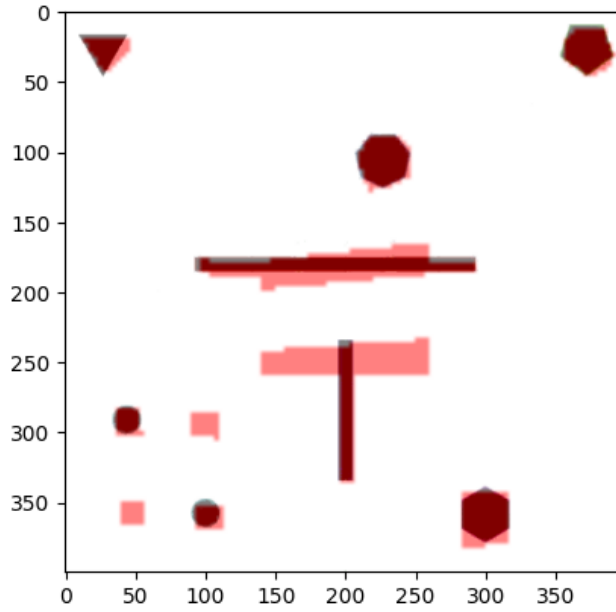
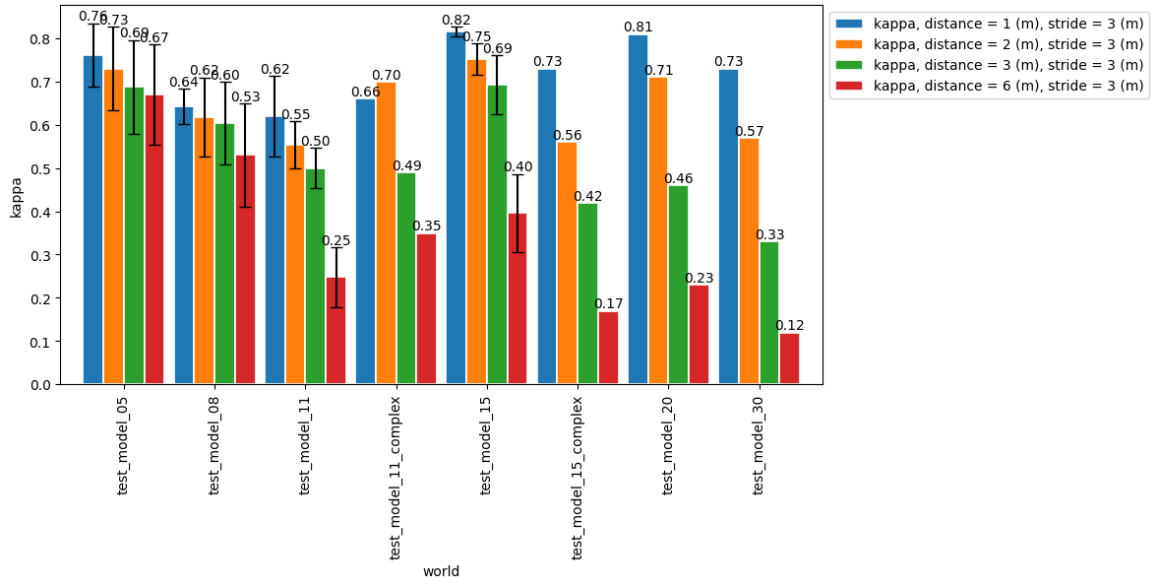


Figure 9: Exemple de zone fantôme situé en bas à gauche de la carte.

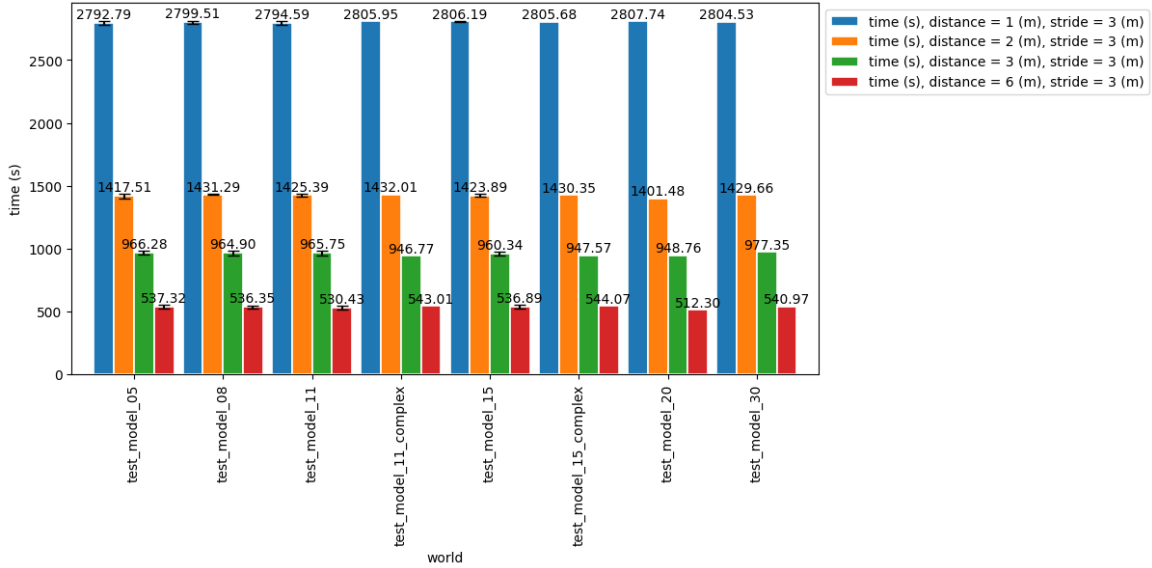
nordique, observé sur la figure 10b, est constant pour chaque valeur de la densité du monde.

Nous pouvons observer sur la figure 11 l'évolution du score de Cohen et du temps d'exécution de l'algorithme *ski nordique* en fonction de la densité du monde pour différentes valeurs du pas s entre les deux crawlers, et une distance $d = 3$ mètres entre les crawlers. sur la figure 11a, nous observons que le score de Cohen est le plus bas pour de grandes valeurs de densités et de grandes valeurs de d , comme pour $d = 6$ mètres et les cartes avec 30 et 20 zones de corrosion. Ceci s'explique par le fait que pour de grandes valeurs de densités et de d , la probabilité que les rayons du signal traversent des zones de corrosions est plus élevée. Il y a donc plus de chance que des zones fantômes soient créées, ce qui fait diminuer le score de Cohen. Les formes allongées des zones de corrosion sont également un facteur qui fait diminuer le score de Cohen comme étudié précédemment. C'est pourquoi nous observons que le score de Cohen est le plus haut pour la carte avec la plus petite densité et sans formes allongées de corrosion, c'est-à-dire la carte avec 15 zones de corrosion.

Sur la figure 11b, nous observons que le temps d'exécution de l'algorithme *ski nordique* est constant pour chaque valeur de la densité du monde. Ceci était attendu comme pour la stratégie *peinture au rouleau*. Cependant, nous observons que le temps d'exécution varie avec le pas s utilisé. Nous nous serions plutôt attendus à ce que le temps d'exécution reste constant avec le pas des crawlers. En effet, indépendamment de la valeur du pas, la distance verticale et horizontale à parcourir par les crawlers reste la même. Cette différence significative du temps d'exécution est due à la manière dont nous avons implémenté l'algorithme *ski nordique* qui n'est pas optimale. Nous n'avons pas fait arrêter les crawlers aux extrémités des plaques, mais



(a) κ en fonction de la densité du monde

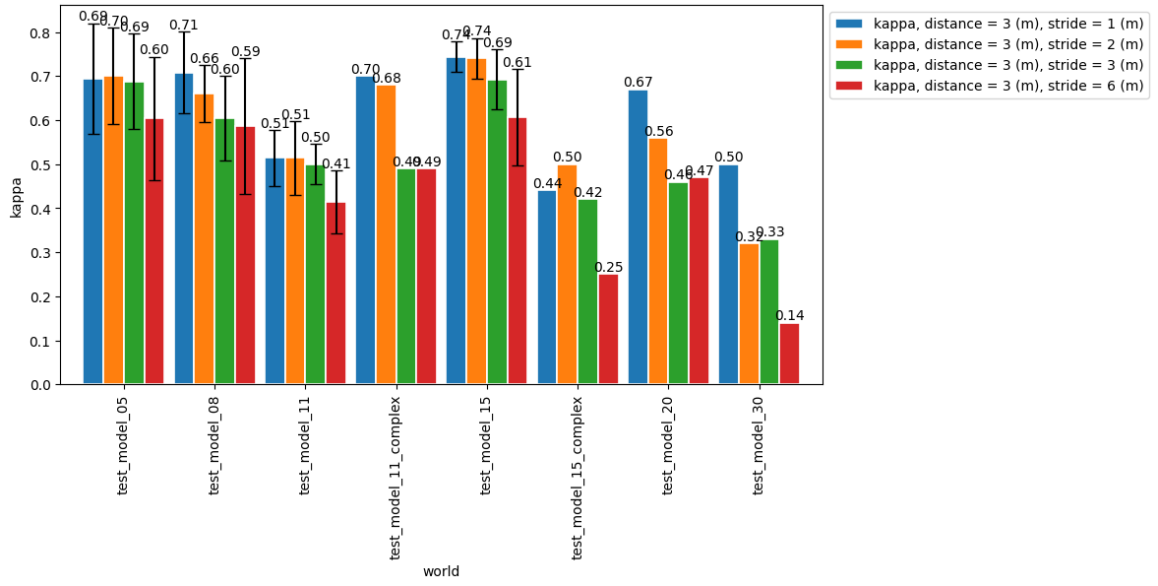


(b) Temps d'exécution en fonction de la densité du monde

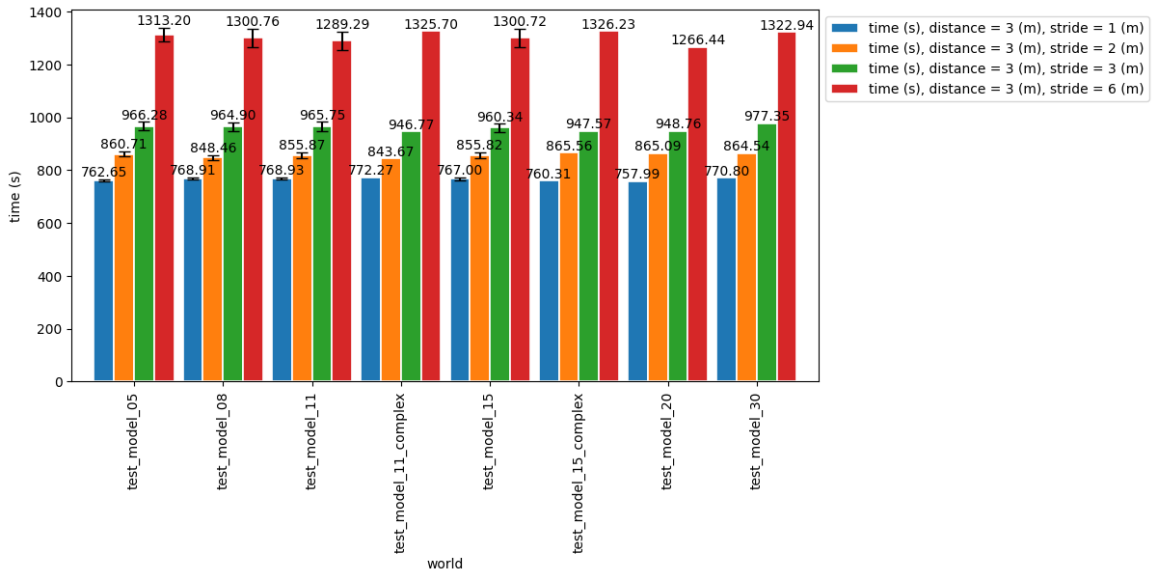
Figure 10: Évolution du κ de Cohen et du temps d'exécution de l'algorithme *ski nordique* en fonction de la densité du monde pour différentes valeurs de la distance entre les deux crawlers.

nous les avons fait continuer d'une valeur du pas s , en plus, par simplicité d'implémentation, sans penser que l'impact sur le temps d'exécution serait significatif.

Sur la figure 12, nous observons l'évolution du score de Cohen et du temps d'exécution de



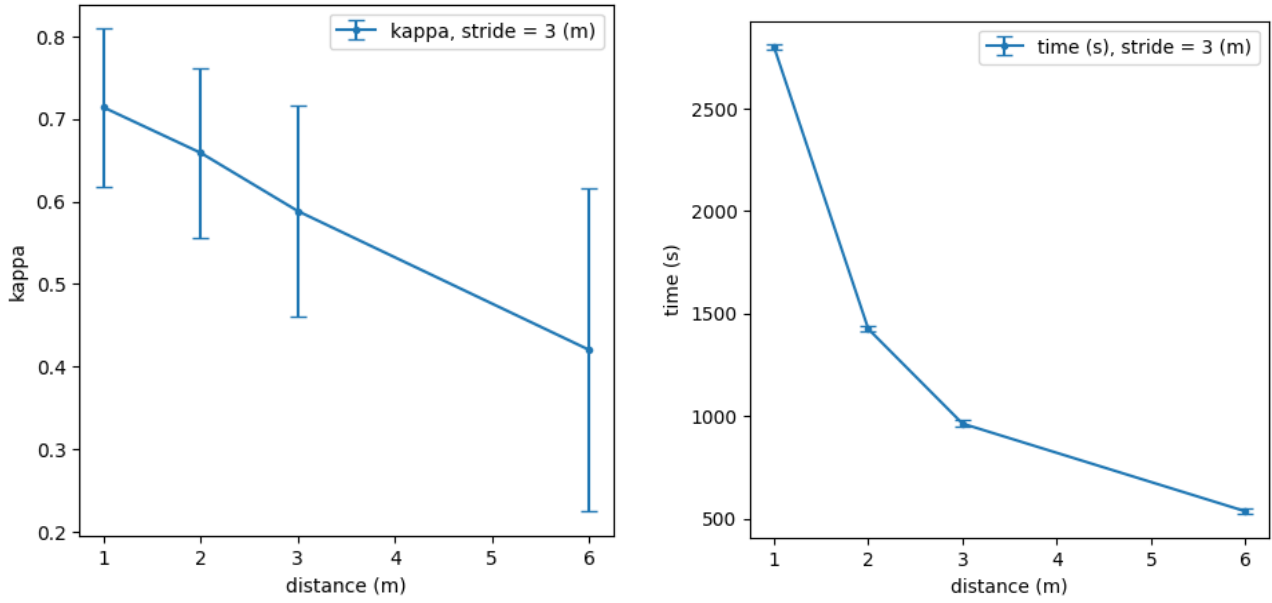
(a) κ en fonction de la densité du monde



(b) Temps d'exécution en fonction de la densité du monde

Figure 11: Évolution du κ de Cohen et du temps d'exécution de l'algorithme *ski nordique* en fonction de la densité du monde pour différentes valeurs du pas entre les deux crawlers.

l'algorithme *ski nordique* en fonction de la distance qui sépare les deux crawlers pour un pas de 3 mètres. Le score semble, comme pour la stratégie *peinture au rouleau*, suivre une relation linéaire avec la distance qui sépare les deux crawlers. Le temps d'exécution semble également



(a) κ en fonction de la distance entre les deux crawlers (b) Temps d'exécution en fonction de la distance entre les deux crawlers

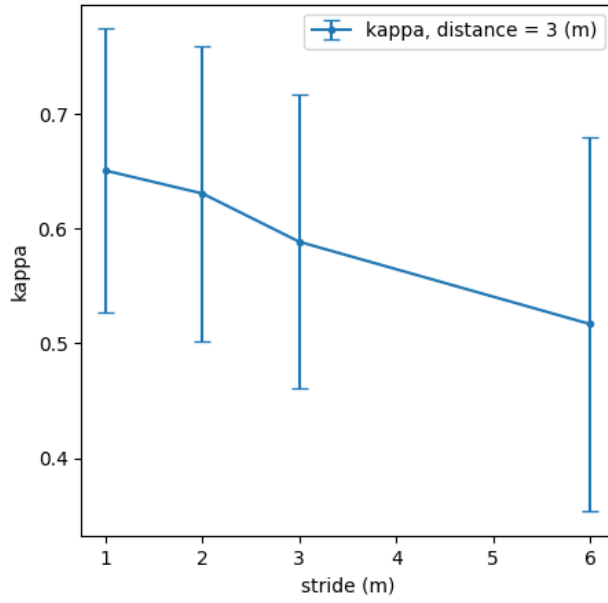
Figure 12: Évolution du κ de Cohen et du temps d'exécution de l'algorithme *ski nordique* en fonction de la distance qui sépare les deux crawlers.

suivre une relation linéaire avec la distance qui sépare les deux crawlers. Le fait que la courbe à figure 12b ne soit pas une droite est dû au fait que plus la distance entre les crawlers est petite et plus le nombre de rotations que les crawlers doivent effectuer est grand. Or le temps de rotation n'est pas négligeable dans les temps d'exécution des algorithmes.

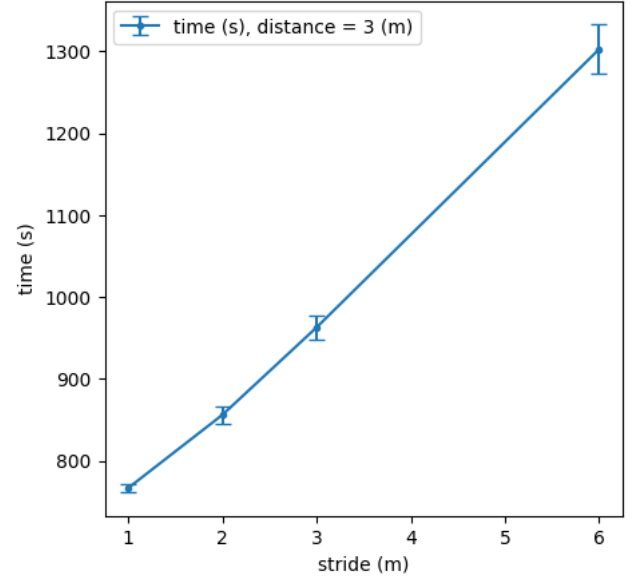
Sur la figure 13, nous observons l'évolution du score de Cohen et du temps d'exécution de l'algorithme *ski nordique* en fonction du pas s des crawlers pour une distance $d = 3$ mètres. Le score semble suivre une relation linéaire avec le pas des crawlers. Plus le pas est petit et plus le score est grand. Cela entre en concordance avec ce que nous expliquions précédemment. Plus le pas est grand et plus, il y a de chance de créer des zones fantômes et donc, de faire diminuer le score de Cohen. Il est à noter cependant qu'il existe une grande variation du score pour les différentes valeurs du pas. Il semble donc que l'impact de la valeur du pas sur le score soit plutôt faible contrairement à l'impact de la valeur de la distance sur le score. Le temps d'exécution, lui, semble suivre une relation linéaire avec le pas des crawlers. Comme expliqué précédemment, ce dernier aurait dû être constant, mais notre implémentation fait dépendre le temps d'exécution du pas des crawlers.

Ici encore, il semble que le score et le temps d'exécution ne soient pas affectés par le fait que les formes soient complexes ou non.

Dans la suite de ce rapport, nous considérerons une distance $d = 3$ entre les deux crawlers



(a) κ en fonction du pas des crawlers



(b) Temps d'exécution en fonction du pas des crawlers

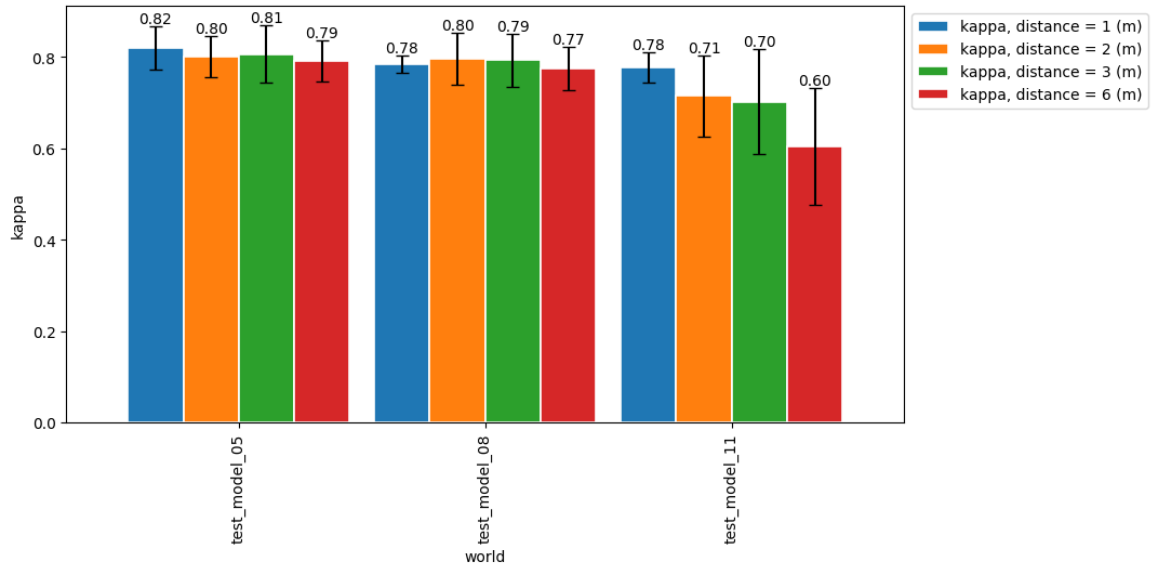
Figure 13: Évolution du κ de Cohen et du temps d'exécution de l'algorithme *ski nordique* en fonction du pas des crawlers.

et un pas $s = 3$ pour l'algorithme *ski nordique*.

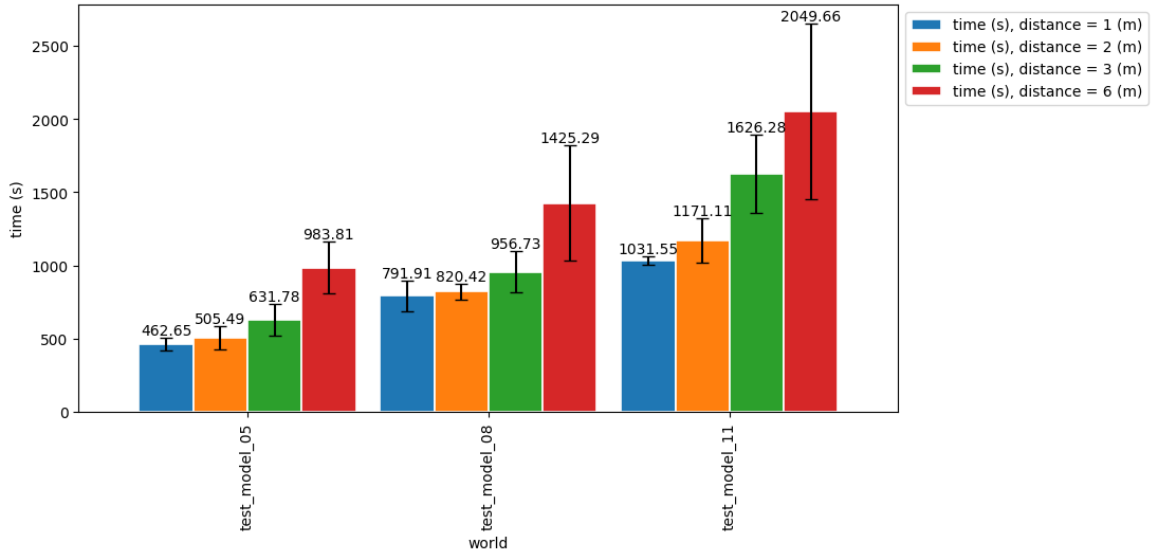
Stratégie de navigation *investigation polygonale*

Nous avons ensuite testé l'algorithme *investigation polygonale* sur des mondes composés de 5, 8 et 11 zones de corrosion. La stratégie d'inspection se base sur les résultats de la stratégie *peinture au rouleau*. Comme expliqué précédemment, nous justifions ce choix par le fait que la stratégie *peinture au rouleau* est la plus rapide des stratégies *a priori* que nous avons implémenté.

La figure 14a montre l'évolution du score de Cohen en fonction de la densité du monde pour chaque valeur de d utilisée dans la stratégie *peinture au rouleau*. Nous avons utilisé un polygone d'investigation à 4 côtés. En premier lieu, nous observons des scores de Cohen relativement indépendant de la distance entre les crawlers pour les cartes avec 5 et 8 zones de corrosion. Ceci est un résultat enthousiasmant, car cela signifie que nous pouvons utiliser la stratégie *investigation polygonale* en se basant sur les résultats de la stratégie *peinture au rouleau* en utilisant une grande distance entre les crawlers, et donc, une stratégie *peinture au rouleau* très rapide. Néanmoins, nous pouvons observer que pour les cartes avec 11 zones de corrosion, le score de Cohen est impacté par la distance entre les crawlers, lorsque cette dernière augmente.



(a) κ en fonction de la densité du monde



(b) Temps d'exécution en fonction de la densité du monde

Figure 14: Évolution du κ de Cohen et du temps d'exécution de l'algorithme *investigation polygonale* en fonction de la densité du monde pour différentes distances entre les crawlers avec un polygone à 4 côtés.

Nous imputons ce résultat au fait que cette carte comporte des zones de corrosion allongées très proches les unes des autres, ayant pour effet de bloquer certains rayons émis et reçus lors de l'inspection polygonale d'une zone. L'inspection polygonale est donc naturellement impactée

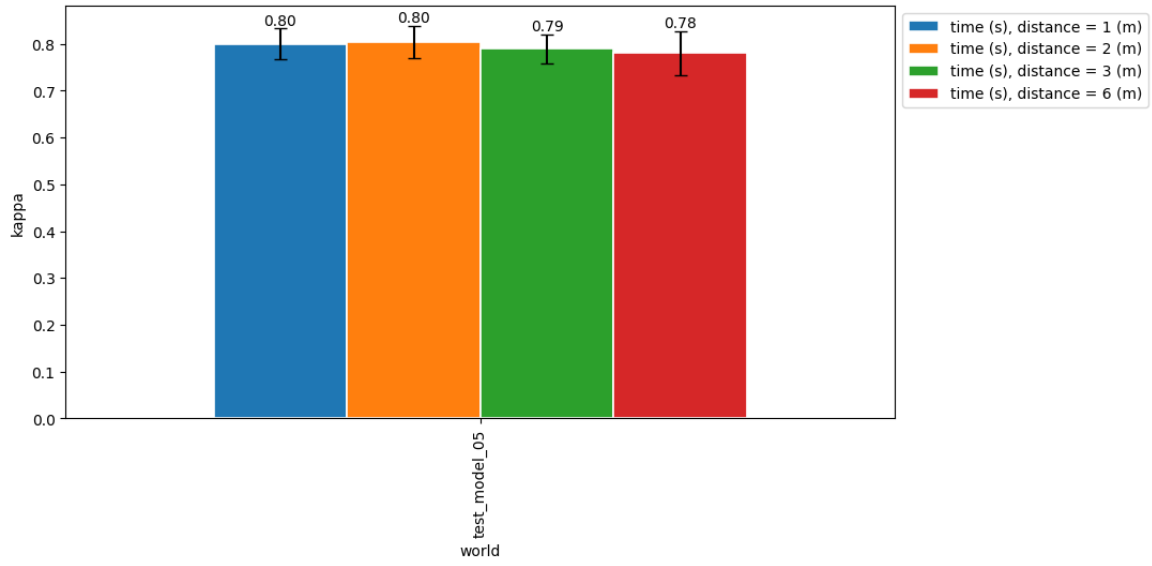
par la densité du monde. Cependant, nous pouvons imaginer, lors de la réparation des structures métalliques, qu'il soit plus convenable de fusionner des zones de corrosion proches les unes des autres en une seule zone de corrosion, bien que cela ne soit pas considéré dans notre problème.

La figure 14b montre l'évolution du temps d'exécution en fonction de la densité du monde pour chaque valeur de d utilisée dans la stratégie *peinture au rouleau*. Nous avons utilisé un polygone d'investigation à 4 côtés. Nous observons que le temps d'exécution augmente avec la densité du monde de manière linéaire. Ceci est un résultat attendu, car l'algorithme *investigation polygonale* est de complexité linéaire en fonction du nombre de zones de corrosion, ce dernier consistant à parcourir toutes les zones de corrosion potentielles et à les inspecter. Nous observons également que le temps d'exécution augmente avec la distance entre les crawlers. En effet, plus la distance entre les crawlers est grande, plus le nombre de zones fantômes est important à l'issue de la stratégie de navigation *peinture au rouleau*, et donc plus le nombre de zones de corrosion potentielles est important. Cependant, ces zones fantômes sont rapidement traitées par l'algorithme *investigation polygonale*. Par exemple, pour la carte 5 avec 11 zones de corrosion, nous obtenons 12 zones de corrosion potentielles avec une distance de 1 mètre entre les crawlers à l'issue de la stratégie *peinture au rouleau*, contre 20 zones de corrosion potentielles avec une distance de 6 mètres entre les crawlers. Or, nous observons un temps d'exécution de 1027 secondes pour la première configuration contre 1616 secondes pour la seconde configuration. Nous avons donc pour une augmentation de 67% du nombre de zones de corrosion potentielles, une augmentation de 57% du temps d'exécution. Le gain de performance n'est pas très important, mais est tout de même significatif.

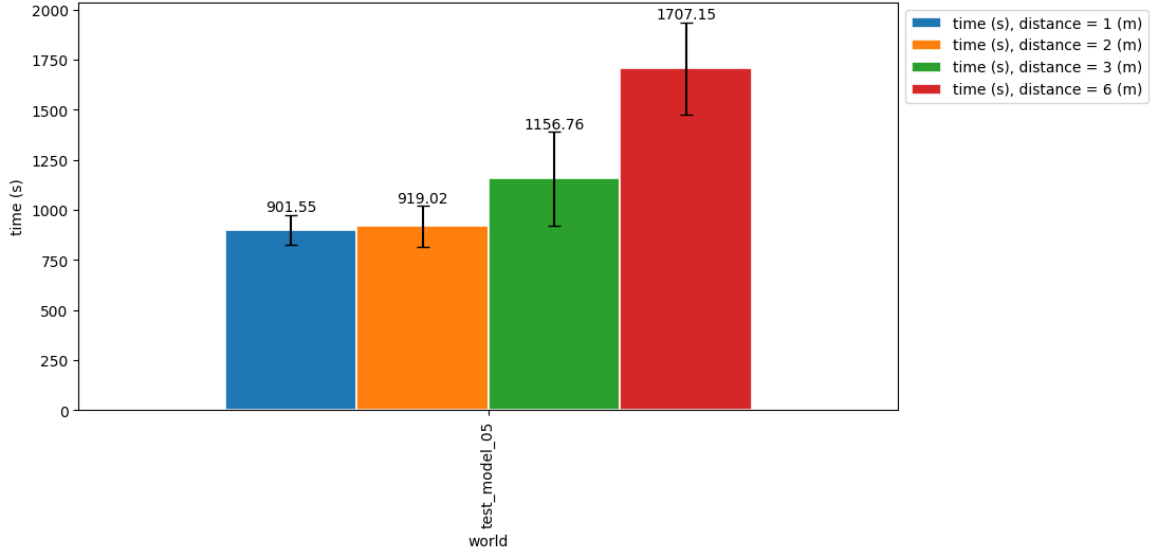
Nous avons également fait varier la taille du polygone d'investigation de la stratégie *investigation polygonale*. Nous présentons sur la figure 15a l'évolution du score de Cohen en fonction de la densité du monde pour la carte 5, pour un polygone à 6 sommets. Premièrement, nous n'observons pas une amélioration significative du score de Cohen lorsque la taille du polygone d'investigation augmente. Au contraire, nous observons une diminution en moyenne, bien que très faible, du score. En théorie, l'augmentation de la taille du polygone d'investigation devrait permettre de mieux approcher l'enveloppe convexe des zones de corrosion, et donc d'obtenir un meilleur score de Cohen. Cependant, nous sommes limités dans notre implémentation par la résolution utilisée pour la discrétisation de la carte. Cependant, pour une résolution plus précise, nous devrions observer une amélioration du score de Cohen.

Nous présentons sur la figure 15b l'évolution du temps d'exécution en fonction de la densité du monde pour la carte 5, pour un polygone à 6 sommets. Ici, nous observons naturellement une augmentation du temps d'exécution lorsque la taille du polygone d'investigation augmente.

Nous aurions également voulu faire varier le nombre de robots utilisés pour l'investigation polygonale ainsi que le nombre d'équipes de robots. Cependant, nous n'avons pas eu le temps de mettre en place une solution de gestion des collisions entre les robots. Il serait intéressant dans un travail futur de mettre en place une telle solution et d'analyser les performances de l'algorithme *investigation polygonale* avec ces différents paramètres. En effet, le temps d'exécution de l'algorithme *investigation polygonale* devrait diminuer lorsque k et n augmentent.



(a) κ en fonction de la densité du monde



(b) Temps d'exécution en fonction de la densité du monde

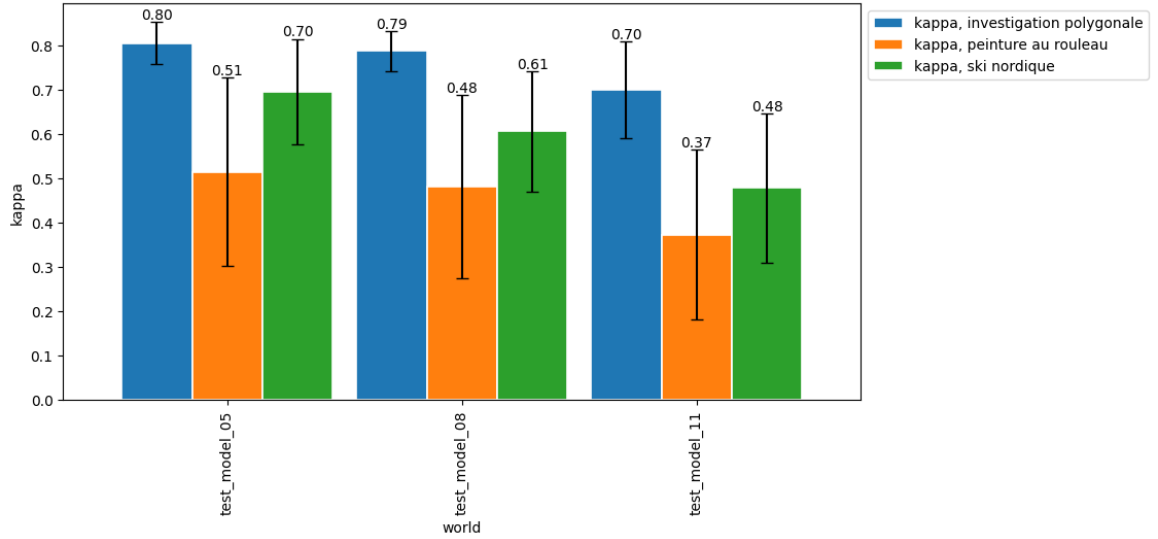
Figure 15: Évolution du κ de Cohen et du temps d'exécution de l'algorithme *investigation polygonale* en fonction de la densité du monde pour différentes distances entre les crawlers avec un polygone à 6 côtés.

Dans la sous-section suivante, nous comparerons les performances de l'algorithme *investigation polygonale* avec celles des algorithmes *peinture au rouleau* et *ski nordique*. Pour ce faire, nous considérerons un polygone d'investigation pour l'investigation polygonale à $p = 4$

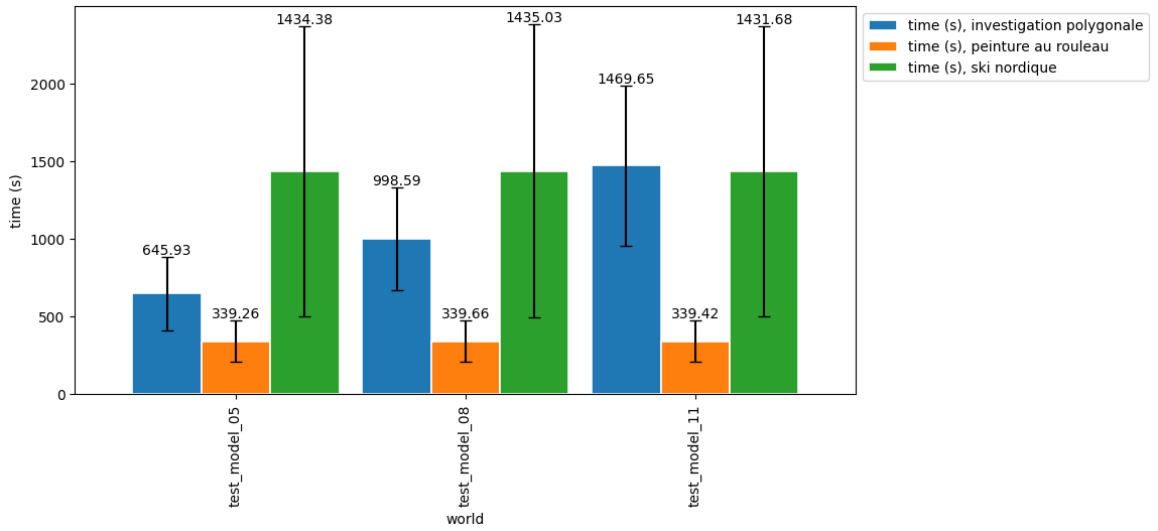
sommets.

Comparaisons et discussions

Dans cette sous-section, nous comparons les performances des différentes stratégies de navigation multi-robots.



(a) κ en fonction de la densité du monde



(b) Temps d'exécution en fonction de la densité du monde

Figure 16: Évolution du κ de Cohen et du temps d'exécution des différents algorithmes en fonction de la densité du monde pour différentes distances entre les crawlers.

Nous présentons sur la figure 16 l'évolution du score de Cohen et du temps d'exécution en fonction de la densité du monde, pour chaque algorithme. Dans cette figure, nous avons représenté les scores et temps d'exécution obtenus, en moyenne, pour les différentes valeurs de d et de s , par soucis de lisibilité. Une version plus détaillée, pour chaque valeur de d , est disponible en annexe B, sur la figure 18.

Nous pouvons observer sur la figure 16a que le score de Cohen obtenu par l'algorithme *investigation polygonale* est supérieur à ceux obtenus par les algorithmes *peinture au rouleau* et *ski nordique*. Seule la stratégie *investigation polygonale* a permis d'obtenir un score de Cohen considéré comme *accord presque parfait* (supérieur à 0.8) selon Landis et Koch.

Nous pouvons observer sur la figure 16b que le temps d'exécution de l'algorithme *peinture au rouleau* est inférieur à ceux obtenus par les algorithmes *investigation polygonale* et *ski nordique*. Pour de faibles densités de cartes comme celles utilisées dans nos expérimentations, l'algorithme *ski nordique* est le moins rapide. Ce résultat est à nuancer. En effet, pour des densités plus élevées, l'algorithme *investigation polygonale* devient plus lent que l'algorithme *ski nordique*, comme nous pouvons déjà presque le voir pour la carte comportant 11 zones de corrosion.

	Gain en performance <i>investigation polygonale</i>	
comparé à	κ de Cohen	Temps d'exécution
<i>peinture au rouleau</i>	+68.39%	+305.80%
<i>ski nordique</i>	+27.92%	-3.92%

Table 3: Gain en performance apporté par la stratégie *investigation polygonale* par rapport aux stratégies *peinture au rouleau* et *ski nordique*.

La table 3 présente le gain en performance apporté par la stratégie *investigation polygonale* par rapport aux stratégies *peinture au rouleau* et *ski nordique*. Ici, nous avons considéré le temps d'investigation de la stratégie *investigation polygonale* comme étant égal à la somme des temps d'investigation de la stratégie *investigation polygonale* et de la stratégie *peinture au rouleau* puisque la première stratégie se base sur la seconde pour explorer les zones de corrosion. Nous pouvons observer que l'algorithme *investigation polygonale* permet d'obtenir un score de Cohen supérieur de 68.39% à celui obtenu par l'algorithme *peinture au rouleau*, bien qu'il soit bien plus lent que ce dernier. En revanche, l'algorithme *investigation polygonale* permet d'obtenir un score de Cohen supérieur de 27.92% à celui obtenu par l'algorithme *ski nordique*, tout en étant plus rapide que ce dernier.

6 Bilan personnel

Au cours de cette période de travail, j'ai pu approfondir mes connaissances dans le domaine de la robotique et de l'inspection multi-robots. J'ai été confronté à divers défis que j'ai essayé de surmonter. Ce projet m'a permis d'acquérir de nouvelles compétences techniques et d'améliorer ma capacité à travailler au sein d'un projet de recherche.

Tout d’abord, j’ai pu approfondir mes connaissances en robotique et dans l’inspection multi-robots. J’ai effectué des recherches approfondies et j’ai consulté des sources fiables pour obtenir des informations pertinentes. Cela m’a permis de comprendre en profondeur les concepts clés liés à mon sujet d’étude.

Au cours de ce projet, j’ai pu grandement évoluer sur le plan personnel et professionnel. J’ai acquis de nouvelles compétences et enrichi mon bagage technique, ce qui a renforcé ma confiance dans mes capacités en tant qu’ingénieur.

Tout d’abord, j’ai considérablement développé mes compétences en programmation, en mettant l’accent sur Python et C++. Grâce à cette expérience, j’ai acquis une meilleure compréhension de ces langages et j’ai gagné en efficacité dans la résolution de problèmes.

Parallèlement, j’ai eu l’opportunité d’explorer de nouveaux outils qui se sont révélés essentiels pour la réalisation de mon projet. J’ai notamment travaillé avec Gazebo, une plateforme de simulation d’environnements virtuels, ce qui m’a permis de tester et de valider mes algorithmes dans des conditions proches de la réalité. J’ai également utilisé Blender pour créer des modèles 3D. De plus, l’apprentissage de ROS (Robot Operating System) m’a offert une infrastructure logicielle solide pour le développement et le contrôle de robots.

En outre, ma participation à ce projet m’a offert une première immersion dans le monde de la recherche. J’ai eu l’occasion d’explorer la littérature scientifique, de me familiariser avec les travaux antérieurs liés à mon domaine d’étude et de mettre en pratique des méthodologies de recherche. J’ai appris à collecter et à analyser des données, rédiger des rapports techniques, et lire des articles scientifiques. Cette expérience m’a permis de développer ma rigueur scientifique et ma capacité à mener des investigations approfondies.

Enfin, ce projet m’a également permis de renforcer ma gestion du temps et de prioriser efficacement les tâches. J’ai dû établir un planning détaillé pour respecter les échéances et m’assurer que chaque étape du projet était réalisée en temps voulu. J’ai appris à gérer les imprévus et à ajuster mon emploi du temps en conséquence pour maintenir un bon équilibre entre la qualité du travail et le respect des délais.

Dans l’ensemble, ce projet a été une expérience enrichissante qui m’a permis de développer mes compétences et de progresser dans mon domaine. J’ai acquis une compréhension approfondie de la robotique et de l’inspection multi-robots, renforcé mes compétences techniques et amélioré ma capacité à m’organiser. Je suis reconnaissant d’avoir eu l’opportunité de m’investir dans ce projet et je suis convaincu que les compétences que j’ai acquises me seront utiles dans mes futurs projets professionnels.

En conclusion, ce projet a été une étape essentielle dans mon parcours professionnel. J’ai acquis des compétences techniques avancées, j’ai exploré de nouveaux outils et j’ai pu me familiariser avec les exigences de la recherche. Ces acquis me seront précieux pour mes projets futurs, qu’il s’agisse de poursuivre dans le domaine de la recherche ou dans l’industrie.

7 Conclusion et perspectives

En conclusion, cette étude a permis de mettre en œuvre et d'évaluer trois stratégies pour la réalisation d'une exploration multi-robots autonome dans des environnements complexes. Les résultats obtenus démontrent l'efficacité de ces approches dans la résolution du problème d'inspection de surfaces métalliques et mettent en évidence leurs caractéristiques respectives.

La première stratégie, la stratégie *peinture au rouleau*, a permis d'obtenir des résultats satisfaisants en des temps d'exploration très courts. La distance entre les robots peut être ajustée pour optimiser les résultats dépendamment de la densité des zones de corrosion inférées. Cette stratégie est particulièrement adaptée à une utilisation préalable à la stratégie *investigation polygonale*, car elle permet de rapidement avoir une vue grossière des zones de corrosion potentielles.

La deuxième stratégie, la stratégie *ski nordique*, a permis d'obtenir des résultats supérieurs à ceux de la stratégie *peinture au rouleau*, mais avec des temps d'exploration plus longs. Cette approche est plus robuste que la stratégie de peinture au rouleau, car elle fait varier l'orientation des rayons UGW émis et reçus, permettant ainsi de plus finement approcher les zones de corrosion.

Enfin, la troisième stratégie, la stratégie *investigation polygonale*, a permis d'obtenir les meilleurs résultats, en termes de précision. Cette stratégie réactive permet d'affiner la localisation des zones de corrosion en se basant sur les résultats d'une des deux stratégies précédentes. Cependant, cette stratégie est plus sensible aux collisions entre les robots, ce qui peut entraîner des résultats moins satisfaisants dans certains cas.

En termes de perspectives, plusieurs axes de développement peuvent être envisagés. Tout d'abord, il serait intéressant d'approfondir l'exploration polygonale en recherchant des méthodes plus robustes pour la gestion des collisions avec les différents robots. Cela permettrait de rendre cette stratégie plus fiable et utilisable dans un large éventail d'environnements complexes. L'extension de cette étude à des expérimentations avec plusieurs équipes de robots constitue également une piste prometteuse pour accélérer davantage le processus d'exploration et d'investigation. Enfin, le déploiement de cette approche sur un système réel constituerait une étape importante pour valider les résultats obtenus en simulation et démontrer l'efficacité de cette approche dans un contexte industriel.

En conclusion, cette étude a permis de mettre en évidence les avantages et les limites de trois stratégies d'exploration autonome. Les résultats obtenus ouvrent la voie à de nombreuses perspectives de développement, notamment en ce qui concerne l'amélioration de l'efficacité et de la robustesse des approches existantes. Ces avancées pourraient avoir un impact significatif dans divers domaines, tels que la robotique de service, l'exploration spatiale ou encore la surveillance environnementale.

References

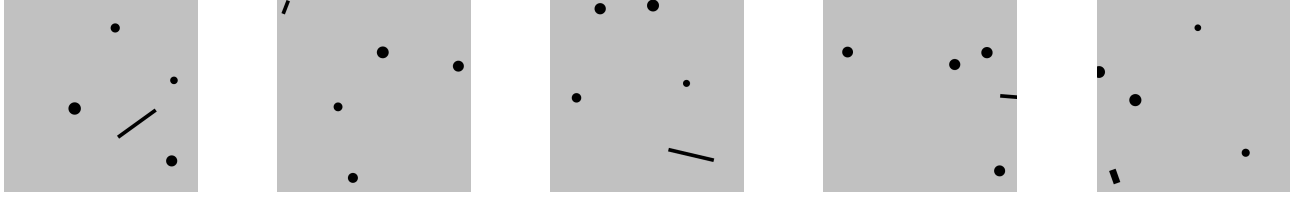
- [1] Othmane-Latif Ouabi, Pascal Pomarede, Nico F. Declercq, Neil Zeghidour, Matthieu Geist, and Cédric Pradalier. Learning the propagation properties of rectangular metal plates for lamb wave-based mapping. *Ultrasonics*, 123:106705, 2022.
- [2] Peter Huthwaite and Francesco Simonetti. High-resolution guided wave tomography. *Wave Motion*, 50(5):979–993, 2013.
- [3] Othmane-Latif Ouabi, Pascal Pomarede, Matthieu Geist, Nico F. Declercq, and Cédric Pradalier. A fastslam approach integrating beamforming maps for ultrasound-based robotic inspection of metal structures. *IEEE Robotics and Automation Letters*, 6(2):2908–2913, 2021.
- [4] Othmane-Latif Ouabi, Ayoub Ridani, Pascal Pomarede, Neil Zeghidour, Nico F. Declercq, Matthieu Geist, and Cédric Pradalier. Combined grid and feature-based mapping of metal structures with ultrasonic guided waves. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5056–5062, 2022.
- [5] Mohammad Faisal Haider, Asaad Migot, Md Yeasin Bhuiyan, and Victor Giurgiutiu. Experimental investigation of impact localization in composite plate using newly developed imaging method. *Inventions*, 3(3), 2018.
- [6] Ayoub Ridani, Othmane-Latif Ouabi, Nico F. Declercq, and Cédric Pradalier. On-plate autonomous exploration for an inspection robot using ultrasonic guided waves. In *2021 European Conference on Mobile Robots (ECMR)*, pages 1–6, 2021.
- [7] Georges Chahine, Pete Schroepfer, Othmane-Latif Ouabi, and Cédric Pradalier. A magnetic crawler system for autonomous long-range inspection and maintenance on large structures. *Sensors*, 22(9), 2022.
- [8] Antoine Bautin, Olivier Simonin, and François Charpillet. MinPos : A Novel Frontier Allocation Algorithm for Multi-robot Exploration. In Chun-Yi Su, Subhash Rakheja, and Honghai Liu, editors, *ICIRA - 5th International Conference on Intelligent Robotics and Applications - 2012*, volume 7507 of *Lecture Notes in Computer Science*, pages 496–508, Montréal, Canada, October 2012. Springer. The original publication is available at www.springerlink.com.
- [9] Roberto Miorelli, Clement FISHER, Andrii Kulakovskiy, Bastien CHAPUIS, Olivier Mesnil, and Oscar d’Almeida. Defect sizing in guided wave imaging structural health monitoring using convolutional neural networks. *NDT & E International*, 122:102480, 06 2021.
- [10] Marcelo Couto, Cid Souza, and Pedro De Rezende. Strategies for optimal placement of surveillance cameras in art galleries. *GraphiCon 2008 - International Conference on Computer Graphics and Vision, Proceedings*, 01 2008.

- [11] Muhammad Asif Arain, Erik Schaffernicht, Victor Hernandez Bennetts, and Achim 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)*, pages 4275–4281, 2016.
- [12] Muhammad Asif Arain, Marcello Cirillo, Victor Hernandez Bennetts, Erik Schaffernicht, Marco Trincavelli, and Achim J. Lilienthal. Efficient measurement planning for remote gas sensing with mobile robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3428–3434, 2015.
- [13] Wikipedia contributors. Beamforming — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Beamforming&oldid=1151960654>, 2023. [Online; accessed 15-June-2023].
- [14] Sebastian Thrun, Michael Montemerlo, Daphne Koller, Ben Wegbreit, Juan Nieto, and Eduardo Nebot. Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data. *Journal of Machine Learning Research*, 4, 05 2004.
- [15] Wikipedia contributors. Piezoelectric sensor — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Piezoelectric_sensor&oldid=1154129092, 2023. [Online; accessed 15-June-2023].
- [16] Wikipedia contributors. Wavelet transform — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Wavelet_transform&oldid=1147185762, 2023. [Online; accessed 15-June-2023].
- [17] Wikipedia contributors. Convolutional neural network — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=1159408824, 2023. [Online; accessed 15-June-2023].
- [18] Wikipedia contributors. Bresenham’s line algorithm — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Bresenham%27s_line_algorithm&oldid=1155124335, 2023. [Online; accessed 15-June-2023].
- [19] Wikipedia contributors. 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, 2023. [Online; accessed 15-June-2023].
- [20] Moustapha Diaby. The traveling salesman problem: A linear programming formulation of. *CoRR*, abs/cs/0609005, 01 2006.
- [21] Gurobi. Traveling salesman problem. https://colab.research.google.com/github/Gurobi/modeling-examples/blob/master/traveling_salesman/tsp_gcl.ipynb#scrollTo=-W_zXnoRaau. [Online; accessed 15-June-2023].

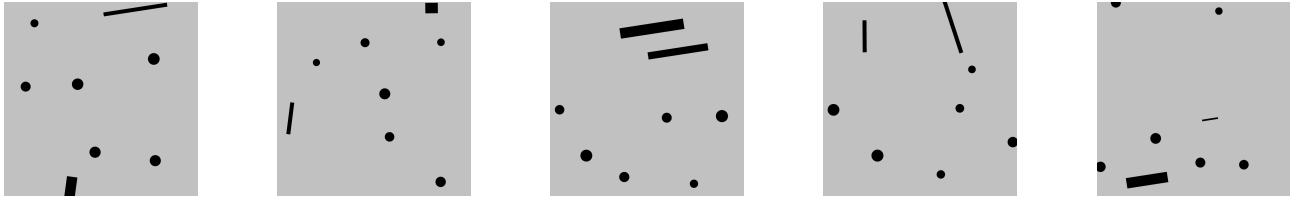
- [22] Kaarthik Sundar and Sivakumar Rathinam. Generalized multiple depot traveling salesmen problem - polyhedral study and exact algorithm. *Computers & Operations Research*, 70:39–55, 2016.
- [23] Dharm Singh, Manoj Kumar Singh, Tarkeshwar Singh, and Rajkishore Prasad. Genetic algorithm for solving multiple traveling salesmen problem using a new crossover and population generation. *Computación y Sistemas*, 22, 07 2018.
- [24] Andras Kiraly and Janos Abonyi. *Optimization of Multiple Traveling Salesmen Problem by a Novel Representation Based Genetic Algorithm*, pages 241–269. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [25] Wikipedia contributors. Cohen’s kappa — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Cohen%27s_kappa&oldid=1130024730, 2022. [Online; accessed 15-June-2023].
- [26] ROS Task Manager. <https://dream.georgiatech-metz.fr/research-projects/ros-task-manager/>. Dernier accès : 2023-05-31.
- [27] Federico Bolelli, Stefano Allegretti, Lorenzo Baraldi, and Costantino Grana. Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling. *IEEE Transactions on Image Processing*, PP:1–1, 10 2019.
- [28] Elad 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. Dernier accès : 2023-06-02.

Annexes

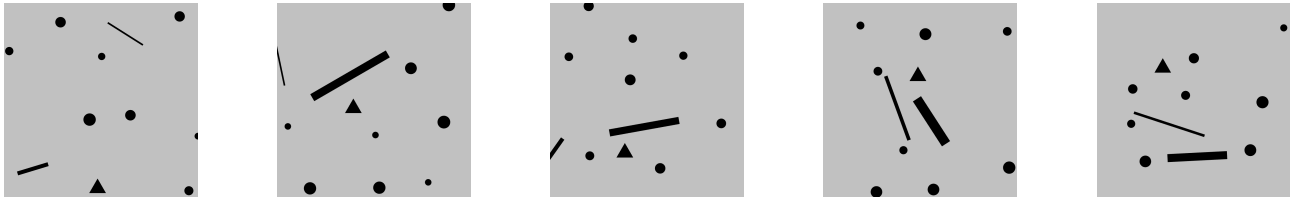
Annex A: Environnements de test



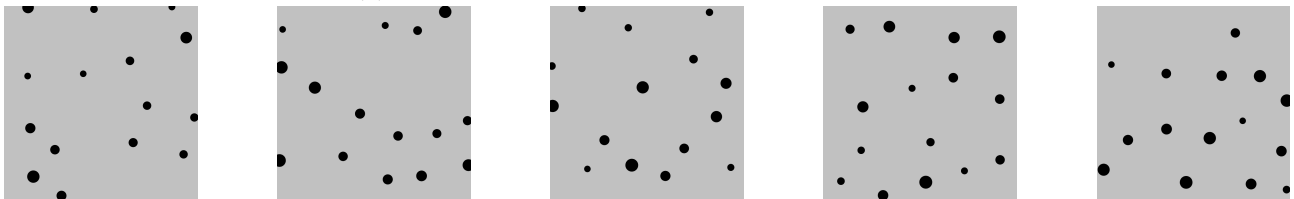
(a) Mondes de test avec 5 zones de corrosion.



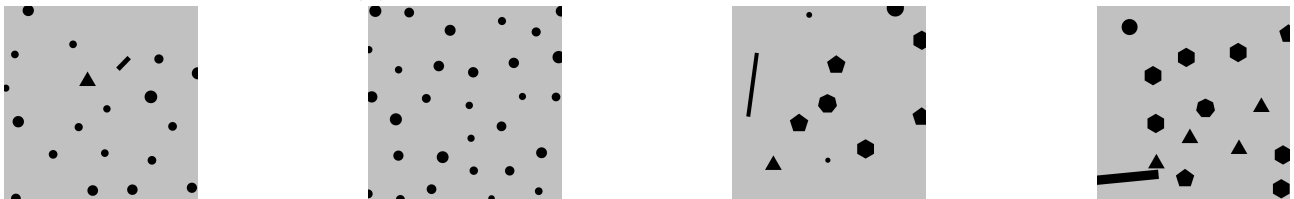
(b) Mondes de test avec 8 zones de corrosion.



(c) Mondes de test avec 11 zones de corrosion.



(d) Mondes de test avec 15 zones de corrosion.



(e) Mondes de test avec 20 zones de corrosion.

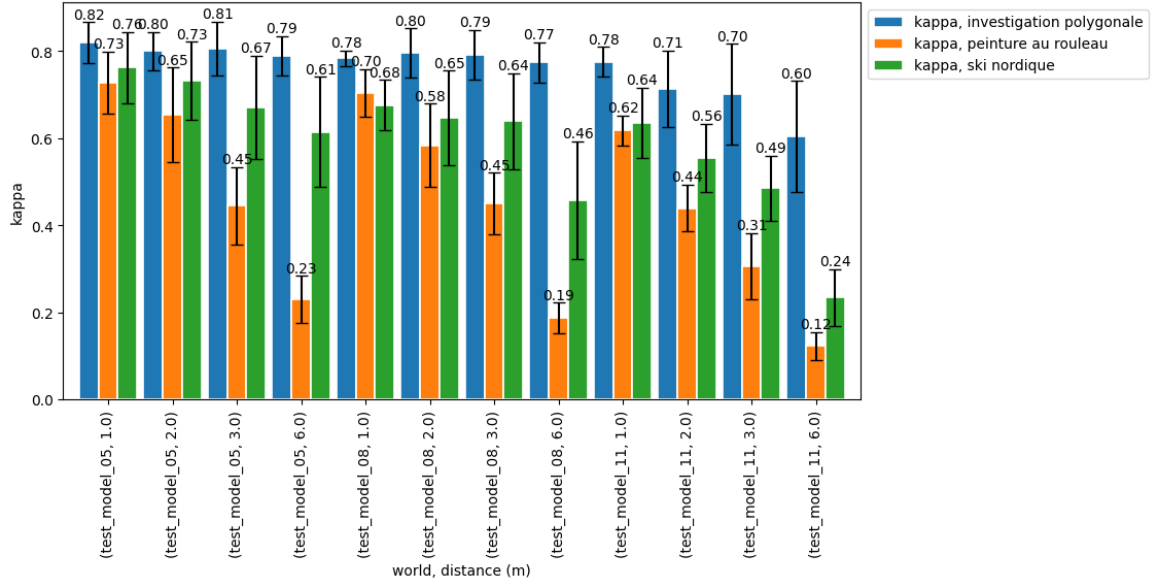
(f) Mondes de test avec 30 zones de corrosion.

(g) Mondes de test avec 11 zones de corrosion complexes.

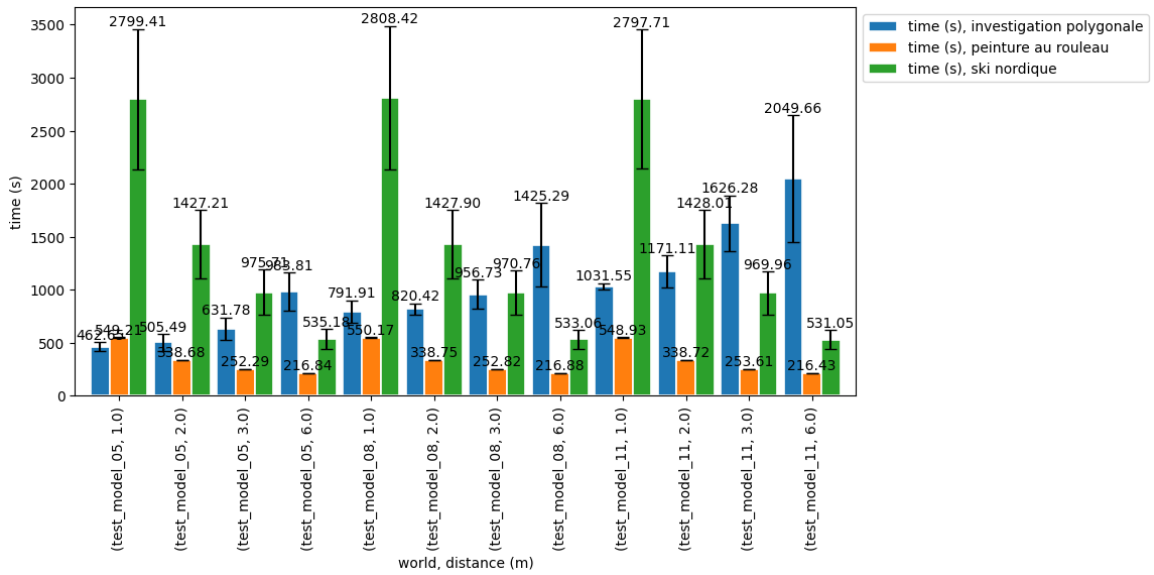
(h) Mondes de test avec 15 zones de corrosion complexes.

Figure 17: Différents environnements de test.

Annex B: Comparaison des différentes stratégies de navigation



(a) κ en fonction de la densité du monde



(b) Temps d'exécution en fonction de la densité du monde

Figure 18: Évolution du κ de Cohen et du temps d'exécution des différents algorithmes en fonction de la densité du monde pour différentes distances entre les crawlers.

Annex C: Résultat d'investigation

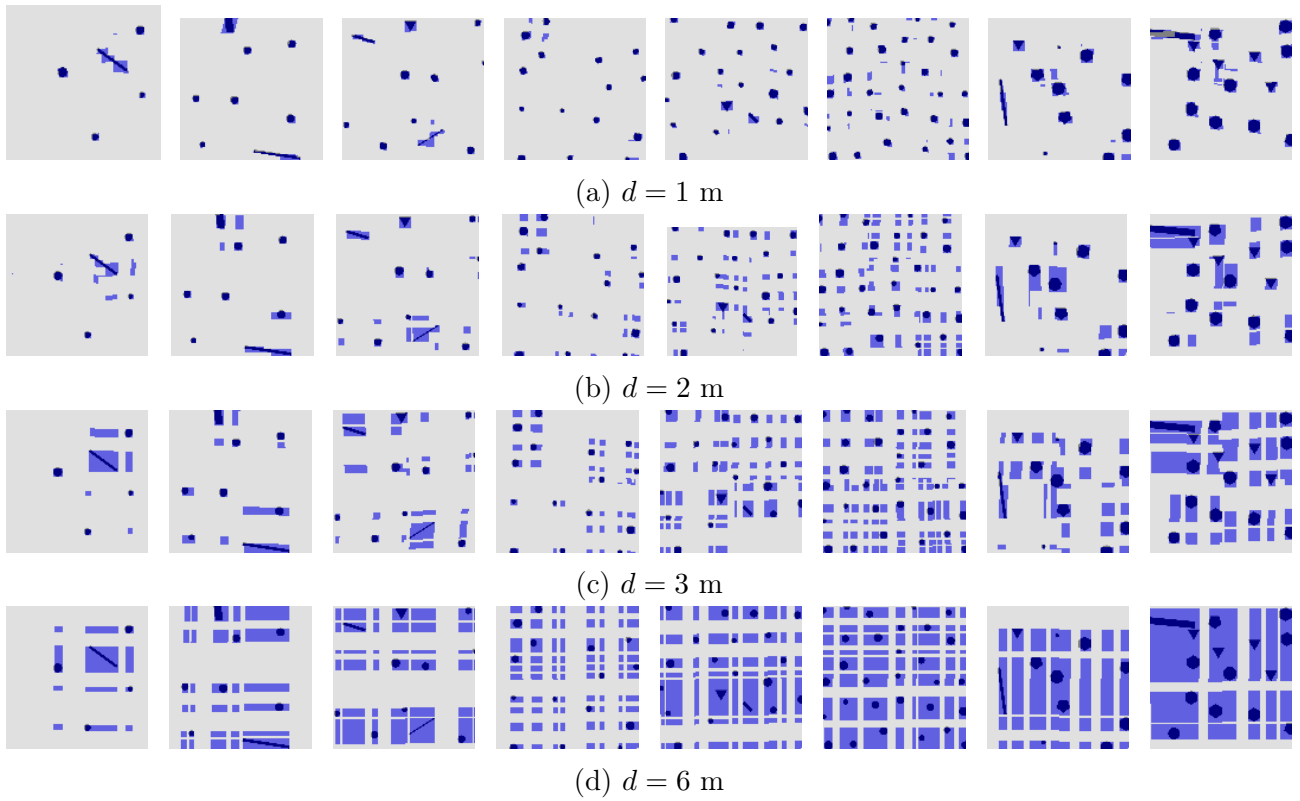


Figure 19: Superposition des cartes d'investigation avec la cartographie des zones de corrosion obtenue pour les différents mondes de test, pour la méthode *peinture au rouleau*.

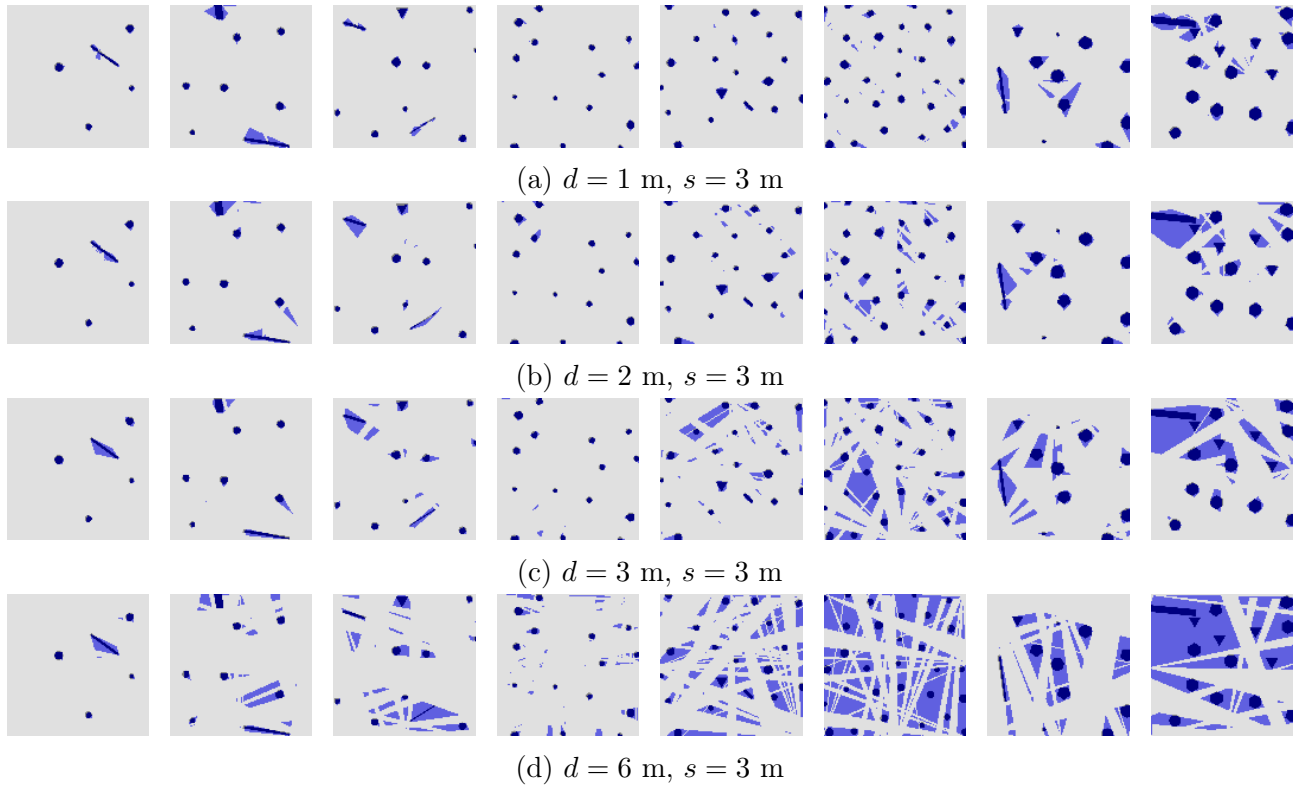


Figure 20: Superposition des cartes d'investigation avec la cartographie des zones de corrosion obtenue pour les différents mondes de test, pour la méthode *ski nordique* - 1.

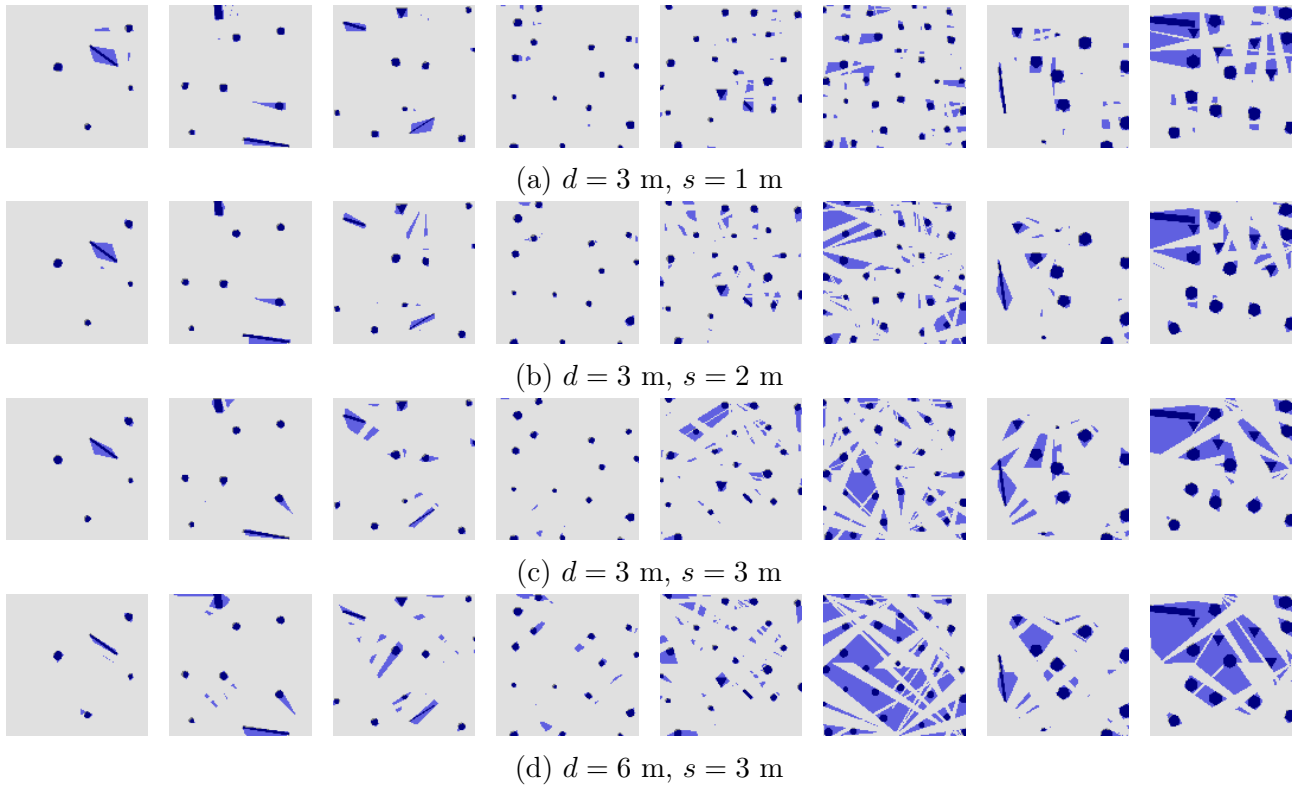
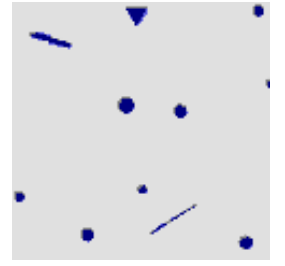
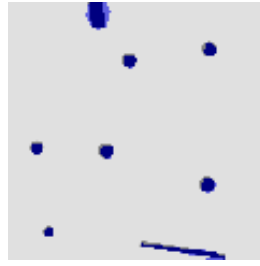
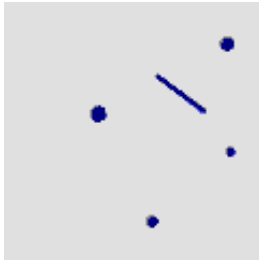
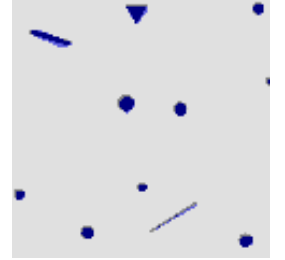
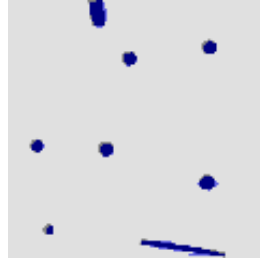


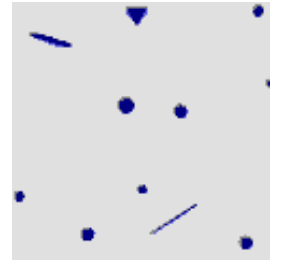
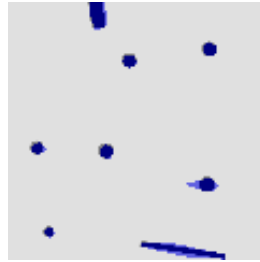
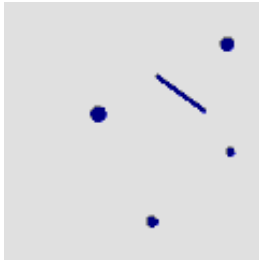
Figure 21: Superposition des cartes d'investigation avec la cartographie des zones de corrosion obtenue pour les différents mondes de test, pour la méthode *ski nordique* - 2.



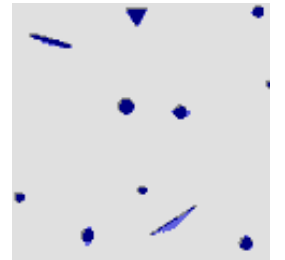
(a) $k = 1, n = 2, p = 4, d = 1 \text{ m}$



(b) $k = 1, n = 2, p = 4, d = 2 \text{ m}$



(c) $k = 1, n = 2, p = 4, d = 3 \text{ m}$

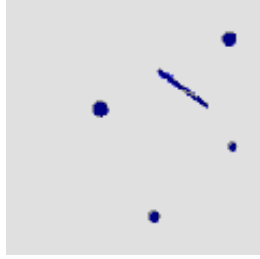


(d) $k = 1, n = 2, p = 4, d = 6 \text{ m}$

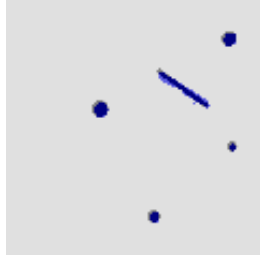
Figure 22: Superposition des cartes d'investigation avec la cartographie des zones de corrosion obtenue pour les différents mondes de test, pour la méthode *investigation polygonale* - 1.



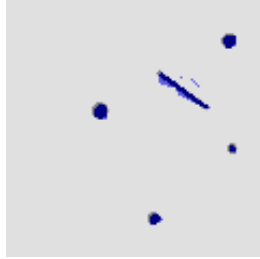
(a) $k = 1, n = 2, p = 6, d = 1$ m



(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 23: Superposition des cartes d'investigation avec la cartographie des zones de corrosion obtenue pour les différents mondes de test, pour la méthode *investigation polygonale* - 2.

Multi-robot navigation and control for acoustic inspection of metal plate structures [R&D]

Brandon ALVES

Abstract

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 robots to explore a metal structure, in a complete and efficient way, in search of corrosion points. The three strategies evaluated include *Roll Painting*, *Nordic Skiing* and *Polygonal Investigation*, all three based on occupancy grids. The *Roll Paint* strategy is a simple but robust approach, which exhaustively covers the search space with rectilinear trajectories and simultaneous movements of the robots. The *Nordic Skiing* strategy is a more complex approach, which introduces a phase shift in the movement of the different robots. The *Polygonal Investigation* strategy tries to improve the result of the previous strategies by investigating around the detected corrosion points. The experiments were carried out in simulation using Gazebo and a crawler model developed for the European project BugWright2. These robots are notably equipped with 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 *paint roller* 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 by allowing a less slow and more precise investigation than the previous one. 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.

Keywords: Navigation ; Multi-Robot ; Tomography ; Ultrasonic Guided Waves ; Inspection.

Résumé

Ce rapport présente une étude sur l'exploration autonome sur des structures métalliques, en se concentrant sur l'évaluation de trois stratégies d'exploration. L'objectif de cette étude est de développer des méthodes efficaces permettant à des robots autonomes d'explorer une structure métallique, de manière complète et efficace, à la recherche de points de corrosion. Les trois stratégies évaluées comprennent la *peinture au rouleau*, le *ski nordique* et l'*investigation polygonale*, toutes trois basées sur des grilles d'occupation. La stratégie *peinture au rouleau* est une approche simple, mais robuste, qui couvre de manière exhaustive l'espace de recherche avec des trajectoires rectilignes et des déplacements simultanés des robots. La stratégie *ski nordique* est une approche plus complexe, qui introduit un déphasage dans le déplacement des différents robots. La stratégie *investigation polygonale* tente d'améliorer le résultat des stratégies précédentes en investiguant autour des points de corrosion détectés. Les expérimentations ont été réalisées en simulation en utilisant Gazebo et un modèle de crawler développé pour le projet européen BugWright2. Ces robots sont notamment équipés de capteurs UGW, spécifiques à notre problématique. Les performances des différentes stratégies ont été évaluées en termes de temps d'investigation et de justesse de la cartographie obtenue. Les résultats obtenus ont démontré l'efficacité de chaque stratégie. La stratégie *peinture au rouleau* a permis une investigation rapide, mais peu précise. La stratégie *ski nordique* a permis une investigation lente, mais plutôt précise. Enfin, la stratégie *investigation polygonale* a permis de combiner les avantages des deux autres stratégies en permettant une investigation moins lente et plus précise que la précédente. Les perspectives futures incluent l'amélioration de la stratégie d'exploration polygonale en développant des méthodes plus robustes pour la gestion des collisions. De plus, l'extension de cette étude à des expérimentations avec plusieurs équipes de robots constitue une piste intéressante pour accélérer davantage le temps d'investigation. Cette étude contribue à la recherche en investigation autonome et fournit des indications pour le développement de systèmes d'investigation efficaces dans des environnements métalliques corrodés. Les résultats obtenus ont des implications importantes dans divers domaines, tels que la robotique de service, l'exploration spatiale et la surveillance environnementale.

Mots-clés : Navigation ; Multi-Robot ; Tomographie ; Ondes Guidées Ultrasoniques ; Inspection.