

INSA Lyon - Département Informatique

---

**Rapport**  
de  
**PROJET DE FIN D'ÉTUDES**

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

---

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

---

**Brandon ALVES**

*Soutenance le 27 juin 2023*

Projet réalisé du **9 janvier 2023** au **30 juin 2023**

dans la structure d'accueil

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

Référent	:	Cédric PRADALIER, Professeur	GT Europe
Référent	:	Olivier SIMONIN, Professeur	INSA Lyon
Tuteur	:	Mathieu MARANZANA, Maître de conférences	INSA Lyon



## Remerciements

Je tiens à exprimer ma sincère gratitude à toutes les personnes qui ont contribué à la réalisation de ce projet et à l'élaboration de ce rapport. Je tiens tout d'abord à remercier mes superviseurs, Cédric Pradalier et Olivier Simonin, pour leurs conseils précieux et leur expertise dans le domaine de la robotique et de la navigation multi-robots. Leurs orientations ont été essentielles pour la réussite de ce projet. Je souhaite également remercier les équipes de recherche des laboratoires INSA Lyon CITI-INRIA et CNRS IRL2958 GT-CNRS pour leur collaboration et leur partage de connaissances. Je tiens à exprimer ma reconnaissance envers mes proches pour leur soutien inconditionnel et leur compréhension tout au long de cette période intense. Leur présence et leurs encouragements ont été une source de motivation essentielle. Enfin, je tiens à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce projet et à la rédaction de ce rapport. Leur soutien et leur engagement ont été déterminants pour le succès de cette étude.



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Étude bibliographique</b>	<b>2</b>
<b>3</b>	<b>Propositions scientifiques et techniques</b>	<b>4</b>
3.1	Définitions préliminaires . . . . .	4
3.2	Proposition de solution . . . . .	6
3.3	Étude théorique de propriétés de la solution proposée . . . . .	15
<b>4</b>	<b>Implémentations des algorithmes</b>	<b>16</b>
<b>5</b>	<b>Expérimentations, validations et évaluations</b>	<b>22</b>
<b>6</b>	<b>Bilan personnel</b>	<b>37</b>
<b>7</b>	<b>Conclusion et perspectives</b>	<b>38</b>
	References bibliographiques	41

## *Annexes*

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



# 1 Introduction

Ce projet de fin d'étude s'inscrit dans le contexte plus large du projet européen BugWright2, qui vise à résoudre la problématique de l'inspection autonome et la maintenance de grandes structures métalliques avec des flottes hétérogènes de robots mobiles. Dans ce projet, nous nous concentrons sur le développement de stratégies de navigation pour un ensemble de robots mobiles utilisant des ondes ultrasoniques guidées (UGW), ou ondes de Lamb, pour réaliser l'inspection des plaques métalliques. En effet, les ondes guidées ont la particularité de se propager le long d'une plaque en interagissant avec la matière qui la compose, et en étant affectées par des changements de géométrie liés, en particulier, à la corrosion.

Le problème principal est donc de définir des stratégies de navigation multi-robots pour optimiser l'acquisition des données permettant de réaliser une tomographie des surfaces métalliques. Pour atteindre cet objectif, nous allons dans un premier temps effectuer une recherche bibliographique, puis mettre en place des méthodes de navigation dans un environnement de simulation. Enfin, nous envisagerons un déploiement sur différents robots en fonction des résultats obtenus. Ce projet sera réalisé sous la supervision d'Olivier Simonin (INSA Lyon CITI lab) et de Cédric Pradalier (CNRS IRL2958 GT).

Les contributions attendues de ce projet sont les suivantes :

- Développement de stratégies de navigation multi-robots pour l'inspection acoustique de structures métalliques.
- Optimisation de l'acquisition de données pour la réalisation de la tomographie.
- Résolution des problèmes de coordination et de synchronisation entre les robots.
- Implémentation des méthodes de navigation dans un environnement de simulation.

Ce rapport présente le travail effectué dans le cadre de notre projet de fin d'étude sur la navigation et le contrôle multi-robots pour l'inspection acoustique de structures métalliques. Dans la première section, nous introduisons le sujet du rapport et présentons les objectifs de notre projet. La deuxième section est consacrée à l'étude bibliographique, où nous résumons les recherches et les publications existantes sur le sujet. Dans la troisième section, nous proposons une solution pour la navigation et le contrôle multi-robots pour l'inspection acoustique de structures métalliques. Cette section est divisée en trois sous-sections : définitions préliminaires, proposition de solution et étude théorique de propriétés de la solution proposée. La quatrième section décrit les détails de l'implémentation technique de notre solution proposée. La cinquième section présente les résultats de nos expérimentations, validations et évaluations. Dans la sixième section, nous faisons un bilan personnel de notre expérience de travail sur ce projet. Enfin, dans la septième section, nous concluons notre rapport en résumant les résultats obtenus, les limites du projet et les perspectives pour des recherches futures.

## 2 Étude bibliographique

Cette section présente une étude approfondie de la littérature scientifique pertinente dans le domaine de l’inspection multi-robots. L’objectif de cette étude bibliographique était de recueillir des informations clés, d’analyser les travaux antérieurs et de situer notre projet dans le contexte de recherche existant. Les références et les sources citées dans cette section fournissent une base solide de connaissances et d’expertise sur le sujet.

Dans un premier temps, nous nous sommes intéressés aux propriétés des ondes guidées ultrasoniques et à leurs applications dans le domaine de la tomographie [1,2], de la cartographie des robots et des structures métalliques [3–6], des robots et des capteurs utilisés dans notre projet [7], de l’exploration multi-robots [8,9] ainsi que des stratégies de placement pour la détection d’objet [10–12].

Le papier [1], les auteurs proposent une méthode pour inférer la géométrie des plaques métalliques en utilisant des ondes de Lamb. Ils utilisent le beamforming [13] pour estimer les limites de la plaque en se basant sur des mesures acoustiques. Les résultats expérimentaux montrent une inférence précise de la géométrie de la plaque. Cependant, les auteurs se contentent de cartographier les contours des structures, sans proposer de méthode pour cartographier les défauts, ce qui est l’objet de notre problématique.

L’article [3] présente une approche basée sur FastSLAM [14] pour l’inspection robotique des structures métalliques en utilisant des ultrasons. Les auteurs proposent une méthode d’allocation des fronts pionniers pour l’exploration multi-robots, permettant une inspection rapide et précise des structures. Notre travail considère la localisation et la cartographie des robots comme étant connues, et se concentre sur la planification de trajectoire pour l’inspection des structures métalliques. L’approche utilisée dans cet article peut donc être complémentaire à notre travail.

Le papier [4], les auteurs proposent une méthode de cartographie des structures métalliques en utilisant des ondes UGW. Ils combinent une grille cartésienne avec des caractéristiques spécifiques pour la détection des défauts. Les résultats expérimentaux montrent une cartographie précise des structures métalliques. Cependant, les auteurs se contentent encore une fois de cartographier les contours des structures, sans proposer de méthode pour cartographier les défauts.

L’article [5] se concentre sur la localisation d’impacts dans des structures composites en utilisant une méthode d’imagerie développée. Les auteurs utilisent des capteurs piézoélectriques [15] pour détecter et localiser les impacts, et une méthode de transformée en ondelettes [16] pour analyser les signaux acoustiques. Les résultats expérimentaux montrent une détection et une localisation précises des impacts. Les capteurs utilisés sont similaires à ceux utilisés dans notre projet, mais ces capteurs sont positionnés de manière fixe sur la structure, alors que nous souhaitons une stratégie mobile.

Le papier [2], les auteurs proposent une méthode de tomographie ultrasonore à haute résolution pour la quantification de l’épaisseur des parois. Ils exploitent la nature dispersante des



ondes de Lamb pour convertir les variations d'épaisseur en variations de vitesse d'onde, permettant ainsi une reconstruction précise de l'épaisseur des parois. Les résultats expérimentaux montrent des reconstructions précises des défauts de corrosion. Cet article fut intéressant pour comprendre les propriétés des ondes ultrasoniques guidées et leurs applications dans le domaine de la tomographie.

L'article [7], issu des travaux du projet BugWright2, présente un système de robot magnétique pour l'inspection et la maintenance autonome de structures de grande taille. Les auteurs proposent un cadre de localisation basé sur une grille créée à partir d'un nuage de points, couplée à des capteurs ultra wideband (UWB) et à une unité de mesure inertielle (IMU). Ils intègrent également un capteur piézoélectrique pour la détection d'ondes UGW pour une localisation précise du robot et une cartographie des caractéristiques structurelles. Ce sont typiquement ces robots et ces capteurs qui sont utilisés dans notre travail.

Le papier [8] présente un algorithme de planification pour l'exploration multi-robots. Cet algorithme, appelé *MinPos*, est conçu pour allouer de manière efficace les frontières aux robots afin de minimiser les déplacements et le temps nécessaire pour explorer l'environnement. Il utilise des techniques avancées d'optimisation pour résoudre ce problème de manière efficace. Cependant, notre travail se concentre sur de l'inspection de structures pour la détection de défauts. Nous souhaitons une inspection fine des zones de corrosion et non une exploration globale de l'environnement.

L'article [10] présente des stratégies pour le placement optimal de caméras de surveillance dans les galeries d'art. Les auteurs proposent des méthodes pour maximiser la couverture de surveillance tout en minimisant le nombre de caméras nécessaires. Cependant, les capteurs utilisés dans notre projet sont des capteurs qui fournissent une information sur un segment uniquement, entre un émetteur et un récepteur, et non une information globale comme une caméra. Les capteurs utilisés dans notre projet sont présentés dans la section 3.1.

Dans l'article [9], les auteurs proposent une méthode de localisation et de dimensionnement automatiques des défauts dans les structures en utilisant l'imagerie par ondes guidées. Ils utilisent un réseau de neurones convolutifs [17] pour analyser les signaux d'ondes guidées et estimer la taille des défauts. Les résultats expérimentaux montrent l'efficacité de l'approche proposée pour inverser à la fois les données synthétiques et expérimentales. Cette approche nécessite des capteurs fixes sur la structure. Nous souhaitons une approche mobile, ne nécessitant pas le déploiement de capteurs sur la structure.

L'article [6] présente une exploration autonome sur plaque pour un robot d'inspection utilisant des ondes UGW. Les auteurs proposent une méthode de localisation basée sur un maillage créé à partir d'un nuage de points et utilisent des mesures de capteurs IMU et UWB. Ils intègrent également un capteur piézoélectrique dans le système pour une localisation précise du robot et une cartographie des caractéristiques structurelles. Dans notre approche, la localisation est supposée connue. Ce travail peut être utilisé pour la localisation du robot, bien que ce ne soit pas le sujet de notre projet. Néanmoins, le type de robot et de capteurs utilisés sont similaires à ceux utilisés dans notre projet.

L'article [11] présente des stratégies efficaces de planification de mesures pour la détection à distance des gaz avec des robots mobiles. L'objectif de l'étude est d'optimiser la planification des mesures de manière à maximiser la précision de détection des gaz tout en minimisant le temps et les ressources nécessaires. Les auteurs proposent différentes approches pour la planification des mesures, notamment l'utilisation de techniques d'exploration basées sur les frontières des zones de détection, la sélection de trajectoires efficaces pour couvrir l'environnement et la réduction du nombre de mesures nécessaires en utilisant des modèles probabilistes. Le type de capteur utilisé possède des caractéristiques similaires à celles des capteurs utilisés dans notre projet. Cependant, notre problématique impose des déplacements de paires de robots. La manière de scinder l'investigation en deux phases, une phase d'inspection grossière et une phase d'affinement, est également similaire à notre approche. Cependant, cette première phase est réalisée par des capteurs fixes, ce qui n'est pas souhaitable dans notre approche. Nous utiliserons également un TSP (Travelling Salesman Problem) pour optimiser les déplacements des robots entre les zones d'intérêt.

Dans le papier [12], les auteurs proposent une méthode efficace de planification des mesures pour la détection de gaz à distance avec des robots mobiles. Leur approche consiste à optimiser la planification des mesures afin de minimiser le temps et les ressources nécessaires. Pour ce faire, ils utilisent une technique de relaxation convexe afin de résoudre le problème d'optimisation qui permet de minimiser le nombre de mesures nécessaires, tout en garantissant une couverture complète de l'environnement. Cette étude est intéressante pour notre problématique et pourrait inspirer des améliorations de notre approche dans l'optimisation du TSP 4 utilisé.

En résumé, les travaux présentés dans cette section sont intéressants pour notre problématique, car ils nous ont permis d'approfondir notre connaissance des problématiques liées à la tomographie par ondes guidées. Les articles [11, 12] sont ceux qui se rapprochent le plus de notre problématique. Cependant, ces articles se concentrent sur un recouvrement de l'environnement sans se soucier de la qualité de la cartographie des zones d'intérêt. De plus, ces articles utilisent des capteurs fixes sur la structure pour la première phase d'inspection grossière, ce qui n'est pas souhaitable dans notre approche. C'est pourquoi nous proposons une approche de navigation multi-robots pour l'inspection acoustique de structures métalliques afin d'optimiser l'acquisition de données qui permettront de réaliser la tomographie des surfaces métalliques.

### 3 Propositions scientifiques et techniques

Dans cette section, nous présentons les définitions préliminaires, la proposition de solution et l'étude théorique de propriétés de la solution proposée.

#### 3.1 Définitions préliminaires

Ici, nous allons expliciter les hypothèses et les définitions préliminaires qui seront utilisées dans la suite de ce rapport. Premièrement, nous considérons un environnement plan, borné et

de taille connue. Nous ne nous intéressons pas à la localisation des robots dans l'environnement, mais nous supposons que chaque robot est capable de connaître sa position dans l'environnement. Nous supposons également que les obstacles sont localisés dans l'environnement. Seules les zones de corrosion ne sont pas localisées.

Nous utilisons des robots de type *crawlers*. Ces robots sont équipés de deux roues motrices et d'une roue folle. Un exemple de crawler est présenté sur la figure 1. La pose du robot est définie par un triplet  $(x, y, \theta)$  où  $x$  et  $y$  sont les coordonnées du robot dans l'environnement et  $\theta$  est l'orientation du robot dans l'environnement. Nous supposons que la pose du robot est connue. Nous supposons également que les robots sont capables de se synchroniser afin de pouvoir se déplacer de manière simultanée ou bien de manière alternée. On note  $cr$  le coût unitaire de rotation du robot et  $ct$  le coût unitaire de translation du robot.

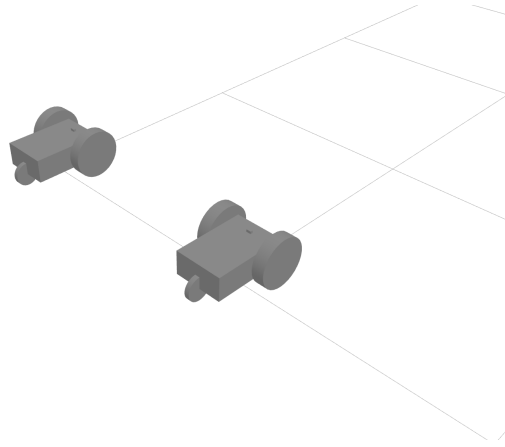


FIGURE 1 – Modèle de crawler utilisé pour l'inspection acoustique de structures métalliques.

Chaque robot est soit émetteur, soit récepteur, soit les deux. Les crawlers sont équipés de différents capteurs. Parmi eux :

- un capteur IMU (Inertial Measurement Unit)
- un capteur UGW (Ultrasonic Guided Waves)
- un capteur LIDAR (Light Detection And Ranging)

Le capteur IMU permet de connaître l'orientation du robot dans l'environnement. Le capteur UGW permet de détecter la présence de corrosion sur la surface métallique en émettant et en recevant des ondes ultrasoniques. Le capteur LIDAR permet de détecter les obstacles dans l'environnement. Les obstacles considérés ici, sont principalement les différents robots inspectant la surface métallique. La détection des zones de corrosion est réalisée par l'émission d'ondes ultrasoniques par un robot et la réception de ces ondes par un autre robot. Dans la mesure où l'onde reçue par un des crawlers est altérée, alors il existe un point de corrosion entre le robot émetteur et le robot récepteur. La détection de ces zones de corrosion est réalisée en temps réel. La portée maximale des ondes ultrasoniques est notée  $d_{max}$ . Nous approximations le temps de propagation des ondes ultrasoniques dans la surface métallique par un temps nul.

Nous utilisons une grille d'occupation pour modéliser l'environnement dans lequel les robots évoluent lors de l'inspection acoustique des structures métalliques. Cette grille nous permet de représenter et de catégoriser les différents états des zones de la surface métallique. La grille d'occupation est composée de cellules, où chaque cellule correspond à une petite région de l'environnement. En particulier, nous avons utilisé une résolution de 0.05 mètre par cellule. Nous utilisons trois états pour caractériser ces cellules : inconnu, vide et occupé. L'état inconnu désigne les zones dont l'état n'a pas encore été déterminé ou détecté. L'état vide indique les zones où il n'y a pas de corrosion détectée, c'est-à-dire que la surface métallique est saine. Enfin, l'état occupé représente les zones de corrosion identifiées, où la présence de défauts ou de détérioration est détectée.

En utilisant cette grille d'occupation, nous pouvons suivre et mettre à jour en temps réel l'état des différentes zones de la surface métallique pendant l'inspection. Cela nous permet de planifier les mouvements des robots, d'optimiser leur trajectoire et d'assurer une couverture complète de la surface à inspecter. De plus, cette représentation nous offre une vision claire de l'état de corrosion de la structure métallique, facilitant ainsi l'analyse et l'évaluation des résultats de l'inspection.

Dans la suite de notre proposition, nous détaillerons les algorithmes et les méthodes utilisées pour mettre à jour la grille d'occupation en fonction des informations recueillies par les capteurs des robots. Nous discuterons également des stratégies de navigation multi-robots qui exploitent cette modélisation pour optimiser l'acquisition de données et améliorer l'efficacité de l'inspection acoustique.

### 3.2 Proposition de solution

Nous présentons notre proposition de solution pour l'inspection acoustique de structures métalliques en utilisant des stratégies de navigation multi-robots. Nous avons développé trois stratégies spécifiques pour optimiser l'acquisition de données et permettre la réalisation de la tomographie des surfaces métalliques. Ces trois stratégies sont les suivantes :

1. Stratégie de navigation *peinture au rouleau*
2. Stratégie de navigation *ski nordique*
3. Stratégie de navigation *investigation polygonale*

Parmi ces stratégies, les deux premières sont non réactives et peuvent être considérées comme des stratégies d'exploration grossières, le but étant de rapidement obtenir une couverture globale de la surface à inspecter. La troisième stratégie est réactive et permet d'optimiser l'acquisition de données pour la réalisation de la tomographie. Ces trois stratégies ont pour objectif de cartographier la surface métallique et de détecter les zones de corrosion. Nous définissons ces trois stratégies de navigation dans les sous-sections suivantes. Nous explicitons également comment la structure de données utilisée pour la cartographie des zones de corrosion, une grille d'occupation, est mise à jour en fonction des informations recueillies par les capteurs UGW des robots.

## Mise à jour de la grille d'occupation pour la cartographie

Lors du balayement de la surface à inspecter par une paire de robots émetteur et récepteur, le robot émetteur émet une onde acoustique dans la structure métallique, qui est ensuite reçue par le robot récepteur. La détection étant considérée comme parfaite, le robot récepteur reçoit l'onde émise par le robot émetteur, sans quasi-altération de la puissance du signal, si et seulement si le segment de droite entre les deux robots ne traverse pas une zone de corrosion. Il est ainsi possible de déterminer si une zone de corrosion est présente entre les deux robots en vérifiant si le signal reçu par le robot récepteur est suffisamment puissant. Dans la mesure où il n'y a pas de détection de corrosion entre l'émetteur et le récepteur, alors le segment de droite entre les deux robots est considéré comme étant libre de corrosion. Dans le cas contraire, alors les points du segment de droite entre les deux robots sont considérés comme étant de la corrosion, à l'exception des points préalablement perçus comme étant libre de corrosion. On surestime donc la présence de corrosion sur le segment. Les stratégies de déplacement viseront à réaliser plusieurs mesures, pour réduire cette surestimation, et approcher la forme réelle de la corrosion.

Il nous suffit maintenant de déterminer quelles cellules de la grille d'occupation sont traversées par le segment de droite entre les deux robots. Pour cela, nous utilisons l'algorithme de tracé de segment de Bresenham [18] qui est couramment utilisé pour déterminer les points d'un plan discret qui doivent être tracés afin de former une approximation de segment de droite entre deux points donnés. Nous détaillons notre implémentation de cet algorithme dans la section 4.

Lors de l'exploration de la surface métallique, la grille d'occupation est mise à jour en fonction des informations recueillies par les robots. Plus précisément, les cellules de la grille d'occupation qui identifient des éléments de corrosion sont mises à jour avec l'état occupé, tandis que les cellules qui identifient des zones saines sont mises à jour avec l'état vide. Nous aboutissons ainsi à une grille d'occupation qui représente l'état de corrosion de la surface métallique, avec, pour chaque zone de corrosion, une approximation de l'enveloppe convexe de la zone de corrosion.

## Stratégie de navigation *peinture au rouleau*

La première stratégie de navigation que nous proposons est la stratégie de navigation *peinture au rouleau*. Nous avons choisi ce nom pour cette stratégie, car le mouvement des robots durant cette stratégie est similaire à celui d'un rouleau de peinture lors de la peinture d'un mur. Cette stratégie repose sur une exploration grossière de la surface à inspecter, où les robots se déplacent en ligne droite sur des trajectoires parallèles, garantissant une couverture globale de la zone d'inspection. Il s'agit donc ici d'effectuer un quadrillage de la surface à inspecter.

Nous présentons sur la figure 2 un schéma décrivant la stratégie de navigation *peinture au rouleau*. Cette stratégie consiste en deux phases, une phase de déplacement vertical et une phase de déplacement horizontal. La figure 2 présente la première phase de déplacement vertical. Afin de réaliser cette stratégie, un minimum de,  $n \geq 2$  robots, alignés horizontalement et séparés par une distance  $d < d_{max}$ , est utilisé. Ces robots se déplacent verticalement, en

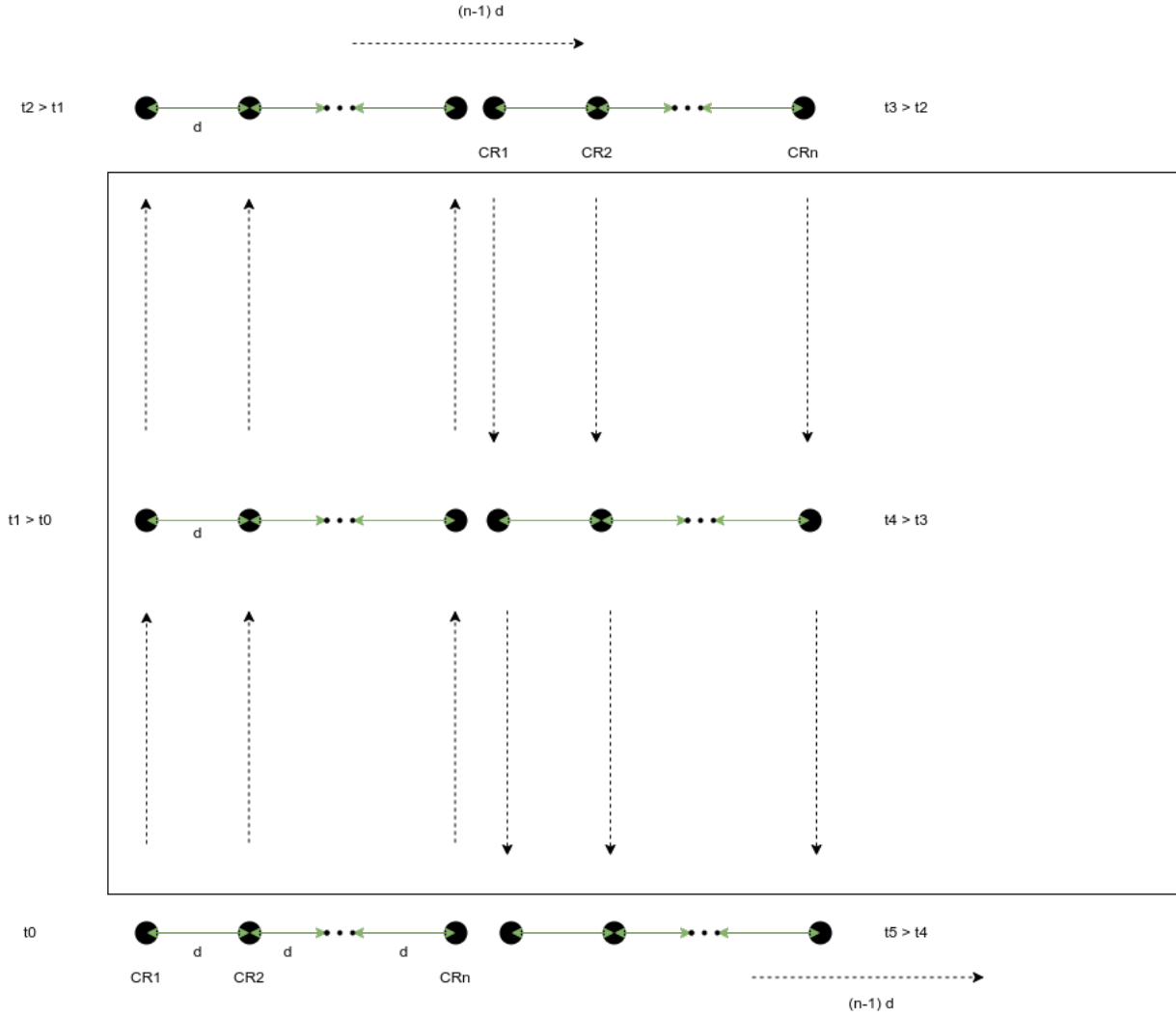


FIGURE 2 – Stratégie de navigation *peinture au rouleau*.

simultané, en suivant une trajectoire parallèle. Une fois l'extrémité de la surface à inspecter atteinte, les robots effectuent une rotation de 90 degrés et se translatent horizontalement, en simultané, d'une distance  $(n - 1) \cdot d$ . Ils effectuent ensuite une nouvelle rotation de 90 degrés et se déplacent de nouveau verticalement, en ligne droite, en simultané, en suivant une trajectoire parallèle entre eux, jusqu'à atteindre l'autre extrémité de la surface à inspecter. Ce procédé est répété jusqu'à ce que la surface métallique soit entièrement inspectée. Le même procédé est ensuite répété, mais cette fois-ci, horizontalement.

Durant cette stratégie, chaque robot est à la fois un émetteur et un récepteur d'ondes UGW. Si la distance qui sépare un robot  $n_a$  d'un robot  $n_b$ ,  $(n_a, n_b) \in \{1, 2, \dots, n\}^2$ , est inférieure à la distance maximale de propagation des ondes UGW,  $d_{max}$ , alors le robot  $n_a$  est capable de

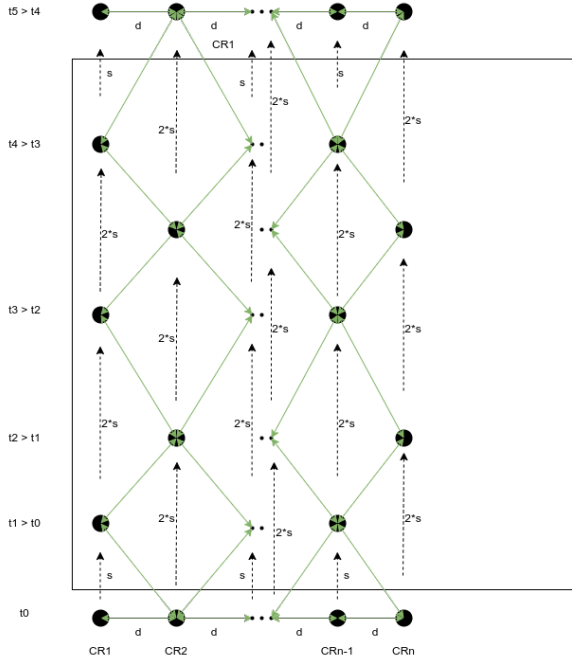
recevoir le signal émis par le robot  $n_b$  et réciproquement. Il n'est cependant pas nécessaire pour un robot  $n_k$ ,  $n_k \in \{1, 2, \dots, n\}$ , de traiter les signaux reçus de tous les autres robots. En effet, les robots étant alignés, les signaux reçus des robots  $n_{k-1}$  et  $n_{k+1}$ , sont suffisants pour la reconstruction de l'état de la surface métallique. Les ondes émises par les robots  $n_1, n_2, \dots, n_{k-2}$  et  $n_{k+2}, \dots, n_n$  ne sont pas utiles pour le robot  $n_k$ . Le robot  $n_k$  peut donc ignorer ces signaux et se concentrer uniquement sur les signaux reçus des robots  $n_{k-1}$  et  $n_{k+1}$ . Dans la mesure où les premiers signaux perçus par le robot  $n_k$  sont ceux émis par les robots  $n_{k-1}$  et  $n_{k+1}$ , du fait de leur proximité, il est possible pour le robot  $n_k$  de filtrer les signaux reçus des autres robots. Ceci constitue une optimisation en termes de traitement pour chaque robot.

Le fait que les robots se déplacent en suivant une trajectoire parallèle et en simultanément, implique que les rayons du signal émis par le robot émetteur et reçu par le robot récepteur, ont toujours une orientation de 0 radian pour la phase verticale et une orientation de  $\frac{\pi}{2}$  radians pour la phase horizontale. Il n'y a donc pas une grande variation de l'orientation du signal émis et reçu. Ainsi, cette stratégie ne pourra approcher les enveloppes convexes des zones de corrosion que par des rectangles. Des exemples de grilles d'occupations résultantes de la stratégie de navigation *peinture au rouleau*, représentées sous formes d'images, où les cellules de la grille correspondent aux pixels des images, sont représentés à l'annexe C, figure 19.

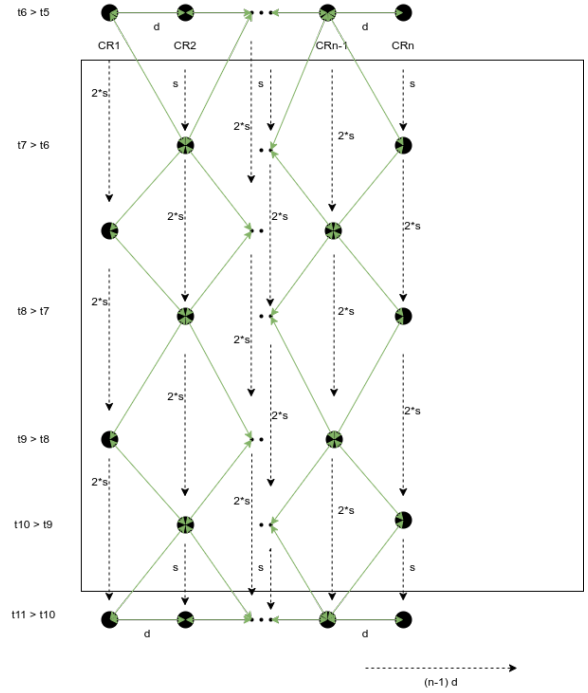
### Stratégie de navigation *ski nordique*

La deuxième stratégie que nous proposons est la stratégie de navigation *ski nordique*. Nous avons choisi ce nom pour cette stratégie, car le mouvement des robots durant cette stratégie est similaire au mouvement des skis d'un skieur. Cette stratégie consiste toujours en des déplacements en ligne droite et en suivant des trajectoires parallèles, mais cette fois-ci, les robots se déplacent de manière séquentielle et non plus de manière simultanée. Dans cette stratégie, nous avons voulu accroître la diversité d'orientation des rayons du signal émis et reçu, afin d'approcher plus précisément les enveloppes convexes des zones de corrosion.

La figure 3 présente un schéma décrivant la stratégie de navigation *ski nordique*. Cette stratégie consiste également en deux phases, une phase de déplacement vertical et une phase de déplacement horizontal. La figure 3 présente la première phase de déplacement vertical. Afin de réaliser cette stratégie, un minimum de,  $n \geq 2$  robots, alignés horizontalement et séparés par une distance  $d < d_{max}$ , est utilisé. Ces robots se déplacent verticalement, en suivant une trajectoire parallèle, mais de manière séquentielle. Les robots impairs se déplacent en ligne droite d'une distance  $s$  et s'arrêtent. Les robots pairs se déplacent ensuite en ligne droite d'une distance  $2 \cdot s$  et s'arrêtent. Ce procédé est répété jusqu'à ce que l'extrémité de la surface soit atteinte (figure 3a). Ensuite, les robots répètent ce même procédé, dans le sens inverse et de manière que les points d'arrêt des robots ne soient pas les mêmes que ceux précédemment (figure 3b). C'est-à-dire que cette fois-ci, ce sont les robots pairs qui commencent par se déplacer en ligne droite d'une distance  $s$  puis s'arrêtent. Ce sont ensuite les robots impairs qui se déplacent en ligne droite d'une distance  $2 \cdot s$  puis s'arrêtent. Les robots se déplacent ensuite horizontalement d'une distance  $(n - 1) \cdot d$  et répètent le même procédé jusqu'à ce que la surface



(a) Stratégie de navigation *ski nordique* - première phase.



(b) Stratégie de navigation *ski nordique* - seconde phase.

FIGURE 3 – Stratégie de navigation *ski nordique*.

métallique soit entièrement inspectée. Le même procédé est ensuite répété, mais cette fois-ci, horizontalement. Afin que les différents robots receveurs puissent recevoir les signaux émis par les robots émetteurs, il faut également imposer que  $s$  soit inférieur strictement à  $\frac{d_{max}}{2}$ , soit  $s < \frac{d_{max}}{2}$ .

Durant cette stratégie, chaque robot est à la fois un émetteur et un récepteur d'ondes UGW. Ici, comme pour la stratégie de navigation *peinture au rouleau*, il n'est pas nécessaire pour un robot  $n_k$ ,  $n_k \in \{1, 2, \dots, n\}$ , de traiter les signaux reçu par les robots autres que  $n_{k-1}$  et  $n_{k+1}$ .

Le fait que les robots se déplacent en suivant une trajectoire parallèle, mais de manière séquentielle, implique que les rayons du signal émis par le robot émetteur et reçu par le robot récepteur, ont une orientation d'une plus grande variation. Ainsi, cette stratégie permet d'approcher les enveloppes convexes des zones de corrosion par des formes plus diverses et précises que des rectangles. Des exemples de grilles d'occupations résultantes de la stratégie de navigation *ski nordique*, représentées sous formes d'images, où les cellules de la grille correspondent aux pixels des images, sont représentés à l'annexe C, figure 20 et figure 21.



## Stratégie de navigation *investigation polygonale*

La troisième stratégie que nous proposons est la stratégie de navigation *investigation polygonale*. Nous avons vu, précédemment, qu'à l'issue de la réalisation de la stratégie de navigation *peinture au rouleau*, l'enveloppe convexe des zones de corrosion était approximée par un rectangle. Cette approximation est un peu plus précise pour la stratégie de navigation *ski nordique*. Il serait intéressant d'avoir un plus grand degré de précision autour des zones potentielles de corrosion. C'est ce que nous proposons avec la stratégie de navigation *investigation polygonale*. Cette stratégie consiste à investiguer autour des zones potentielles de corrosion, détectées au préalable par une des deux stratégies de navigation précédentes. Elle consiste à positionner les robots autour des zones de corrosion et à les faire se déplacer en suivant une trajectoire polygonale, de manière que les rayons du signal émis et reçu aient une orientation d'une plus grande variation autour même de ces zones.

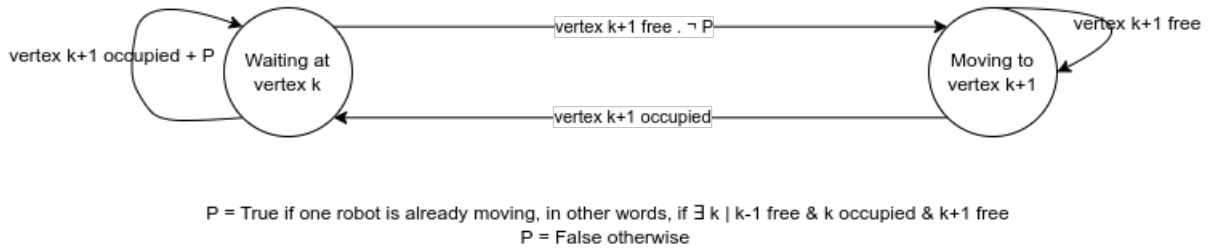
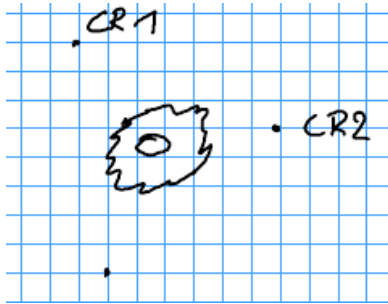


FIGURE 4 – Stratégie de navigation *investigation polygonale*.

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*.

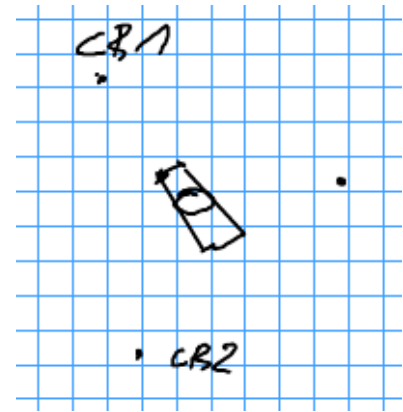
Nous présentons à la figure 5 un exemple des différentes phases de déplacement de la stratégie de navigation, *investigation polygonale* avec  $k = 1$  équipe,  $n = 2$  robots et un polygone à 3 sommets. Sur ces différentes figures, nous avons représenté une zone de corrosion par un cercle et une approximation 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



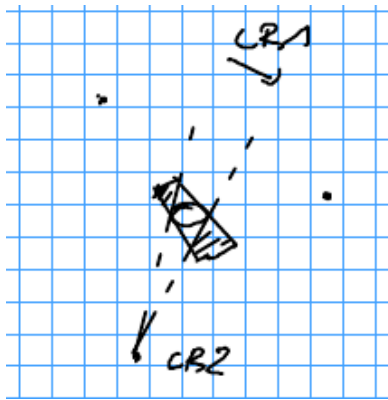
(a) Phase initiale.



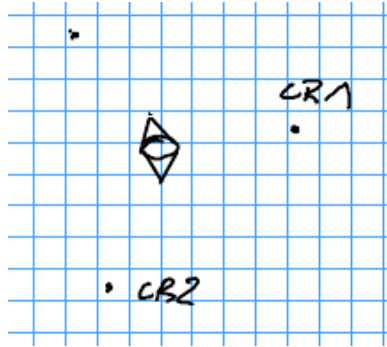
(b) Première phase de déplacement.



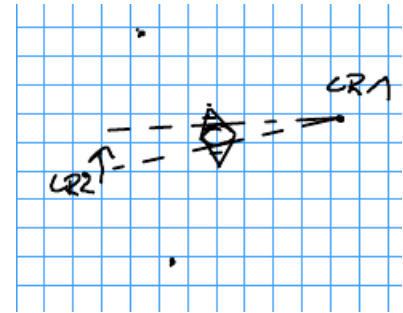
(c) Seconde phase.



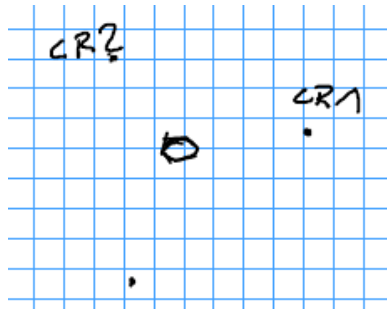
(d) Seconde phase de déplacement.



(e) Troisième phase.



(f) Troisième phase de déplacement.



(g) Dernière phase.

FIGURE 5 – Phases de déplacement de la stratégie de navigation *investigation polygonale*.

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.

**Définition 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 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*.

**Définition 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  $u$ .

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

**Définition 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  $G$ .

**Définition 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.

**Définition 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.

**Définition 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.

**Définition 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 7 [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.

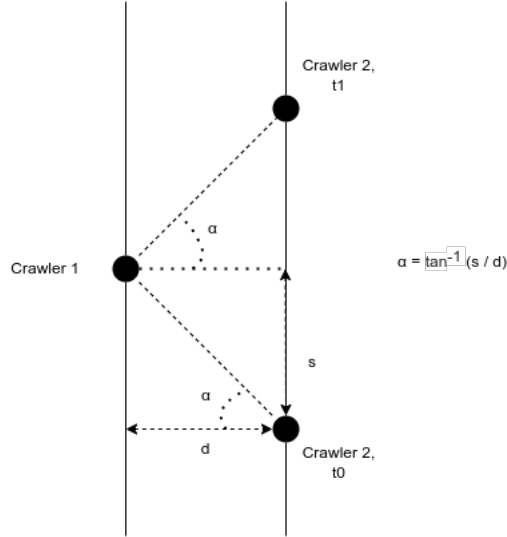


FIGURE 6 – Orientation du signal émis et reçu pour la stratégie de navigation *ski nordique*.

#### 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  $P$ .*

*Démonstration.* 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.*

*Démonstration.* 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  $\kappa$  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, ici <sup>1</sup> et ici <sup>2</sup>.

---

1. [https://gitlab.georgiatech-metz.fr/bugwright2/bugwright2-ws/blob/cr\\_nav\\_strat/bugwright\\_ws/src/floor\\_nav/missions/peinture\\_au\\_rouleau.py](https://gitlab.georgiatech-metz.fr/bugwright2/bugwright2-ws/blob/cr_nav_strat/bugwright_ws/src/floor_nav/missions/peinture_au_rouleau.py)

2. [https://gitlab.georgiatech-metz.fr/bugwright2/bugwright2-ws/blob/cr\\_nav\\_strat/bugwright\\_ws/src/floor\\_nav/missions/ski\\_nordique.py](https://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

```

---

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 inspec-



teurs. 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, 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<sup>3</sup>.

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

---

3. [https://gitlab.georgiatech-metz.fr/bugwright2/bugwright2-ws/blob/cr\\_nav\\_strat/bugwright\\_ws/src/floor\\_nav/missions/investigation\\_polygonale.py](https://gitlab.georgiatech-metz.fr/bugwright2/bugwright2-ws/blob/cr_nav_strat/bugwright_ws/src/floor_nav/missions/investigation_polygonale.py)

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.

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 $\kappa$ 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  $\kappa$  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.

$\kappa$	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  $\kappa$  de Cohen selon Landis et Koch.

L’algorithme de calcul du  $\kappa$  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  $\kappa$ , 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.

---

**Algorithm 2:** Algorithme du  $\kappa$  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  $\kappa \leftarrow \frac{TP+TN-f_c}{TP+TN+FN+FP-f_c}$ 
```

---

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  $\kappa$  de Cohen est ensuite calculé à partir des taux d’observation et d’accord observés, prenant en compte la possibilité de concordance due au hasard. Plus le  $\kappa$  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  $\kappa$  proche de 0 indique un faible niveau d’accord, tandis qu’un  $\kappa$  négatif suggère une discordance entre les résultats. Une Interprétation du  $\kappa$  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 contrôle 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  $\kappa$  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  $\kappa$  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*.

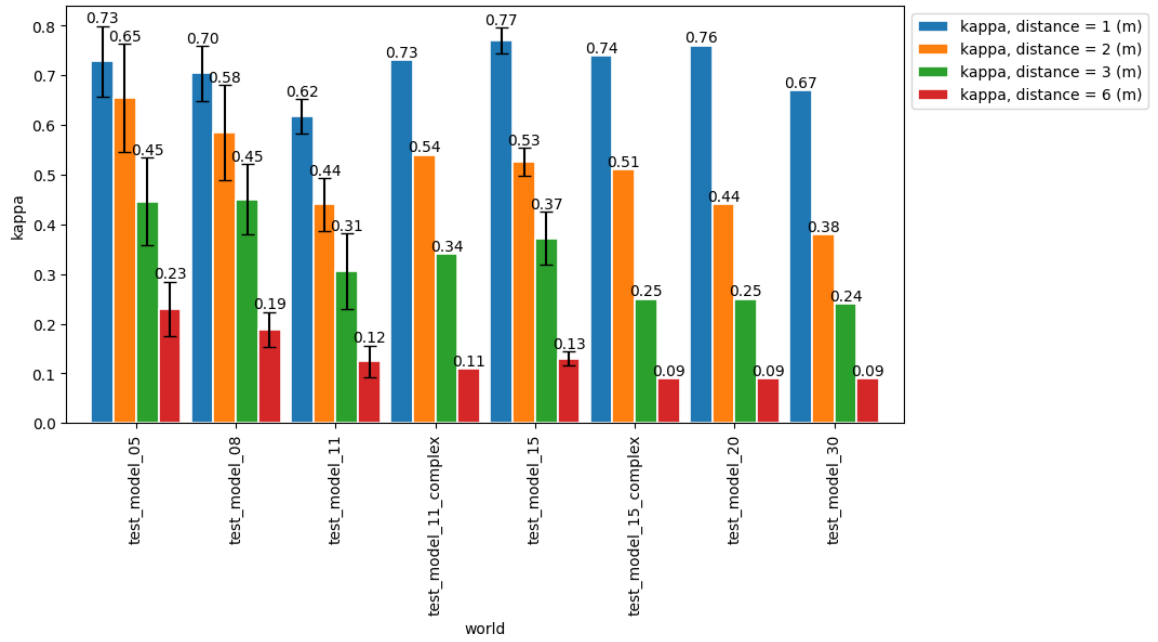
Les différents résultats issus des différentes simulations effectuées sont disponibles en annexe C. Sur ces images, il est possible de voir en noir les zones réelles de corrosion et en bleu les zones détectées comme comportant de la corrosion par les différents algorithmes de navigation.

## Stratégie de navigation *peinture au rouleau*

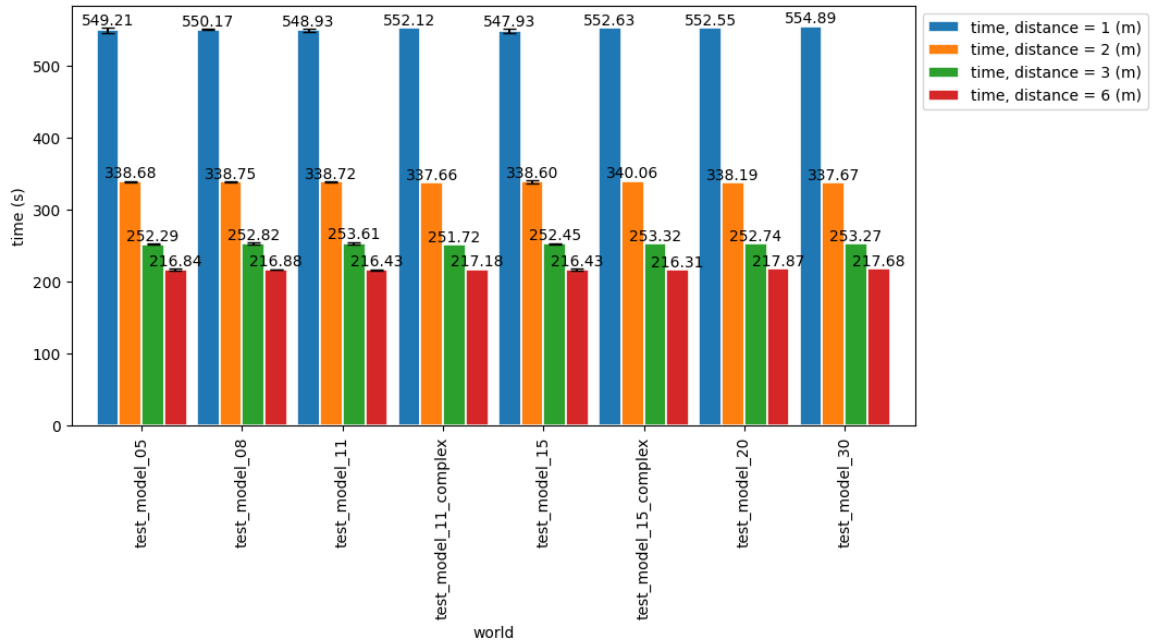
Nous résumons sur la figure 7a l'évolution du score de Cohen en fonction de la densité du monde pour chaque valeur de  $d$ . Nous résumons également sur la figure 7b l'évolution du temps d'inspection en fonction de la densité du monde pour chaque valeur de  $d$ .

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  $d$

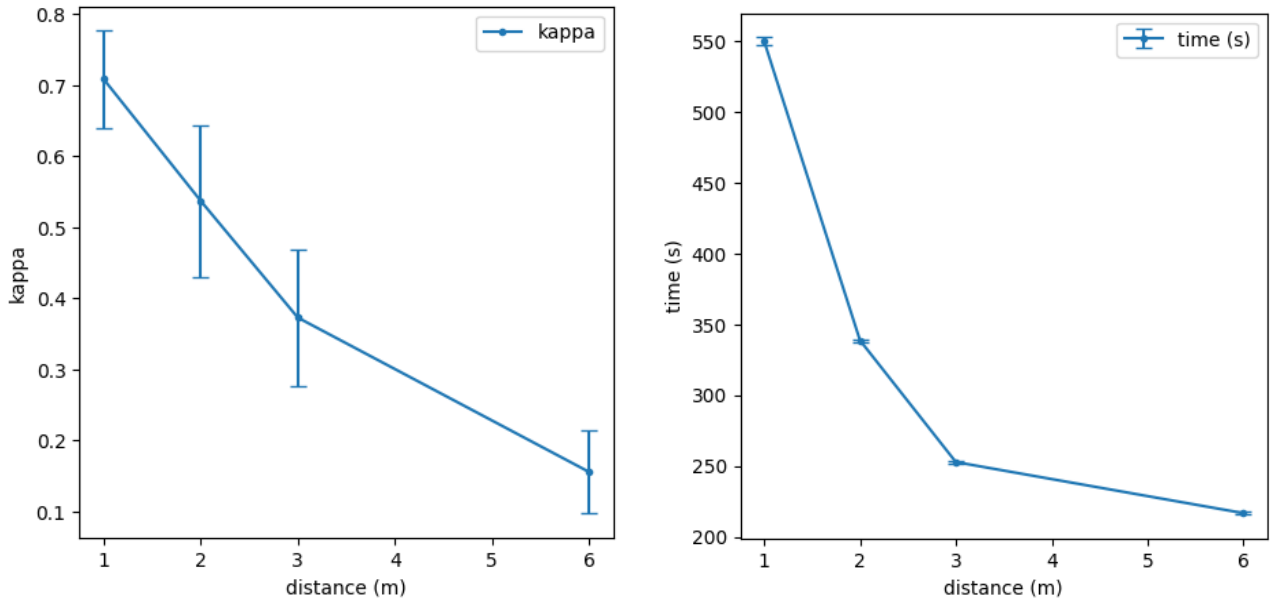


(a)  $\kappa$  en fonction de la densité du monde.



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

FIGURE 7 – Évolution du  $\kappa$  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)  $\kappa$  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  $\kappa$  de Cohen et du temps d'exécution de l'algorithme *peinture au rouleau* en fonction de la distance qui sépare les deux crawlers.

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 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*.



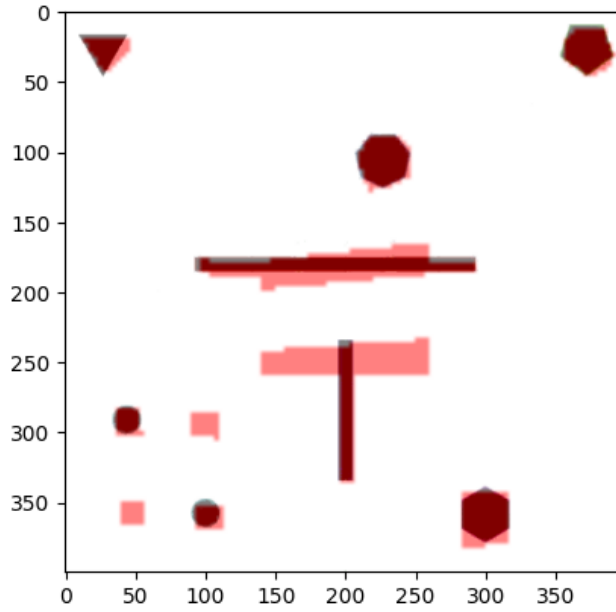


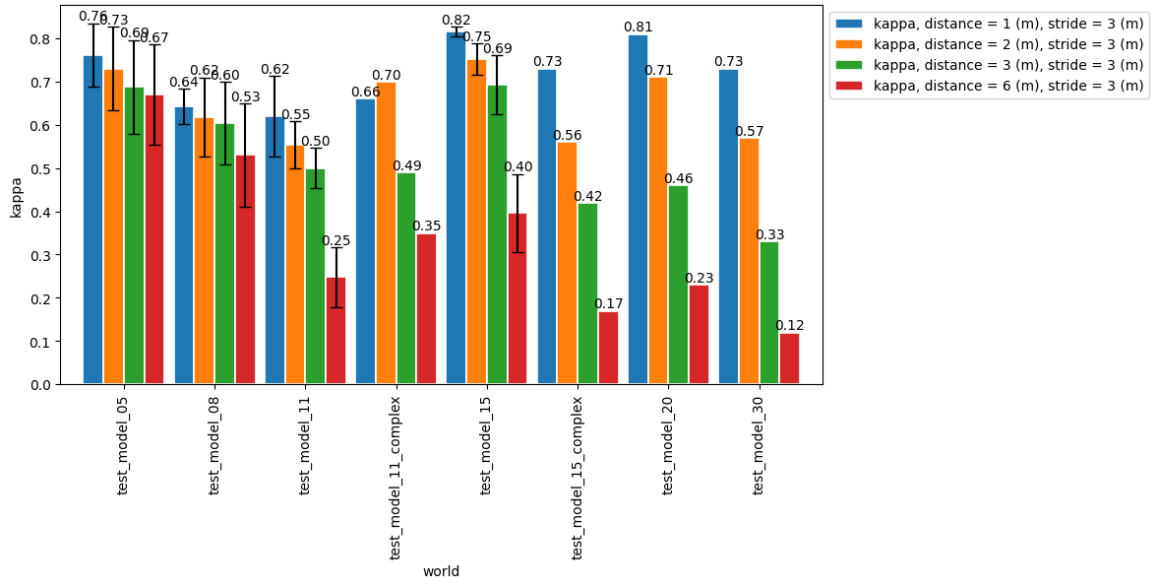
FIGURE 9 – Exemple de zone fantôme situé en bas à gauche de la carte.

## Stratégie de navigation *ski nordique*

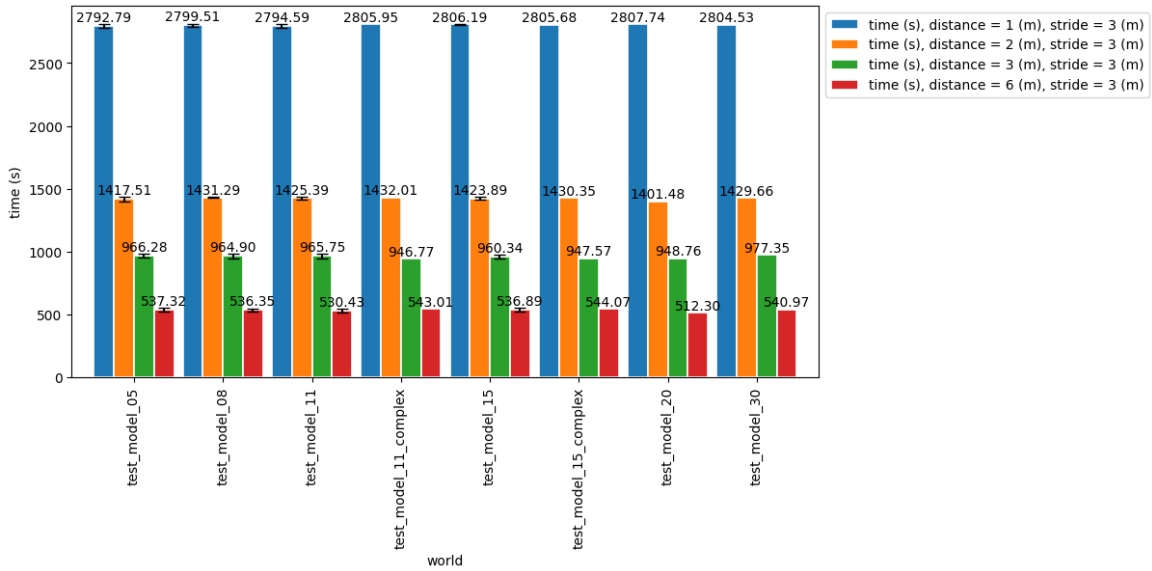
Nous allons maintenant analyser les résultats obtenus pour l'algorithme *ski nordique*. Comme pour l'algorithme *peinture au rouleau*, nous avons fait varier la densité du monde et la distance  $d$  entre les deux crawlers, mais également le pas  $s$  utilisé entre les deux crawlers. La figure 10 présente 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 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



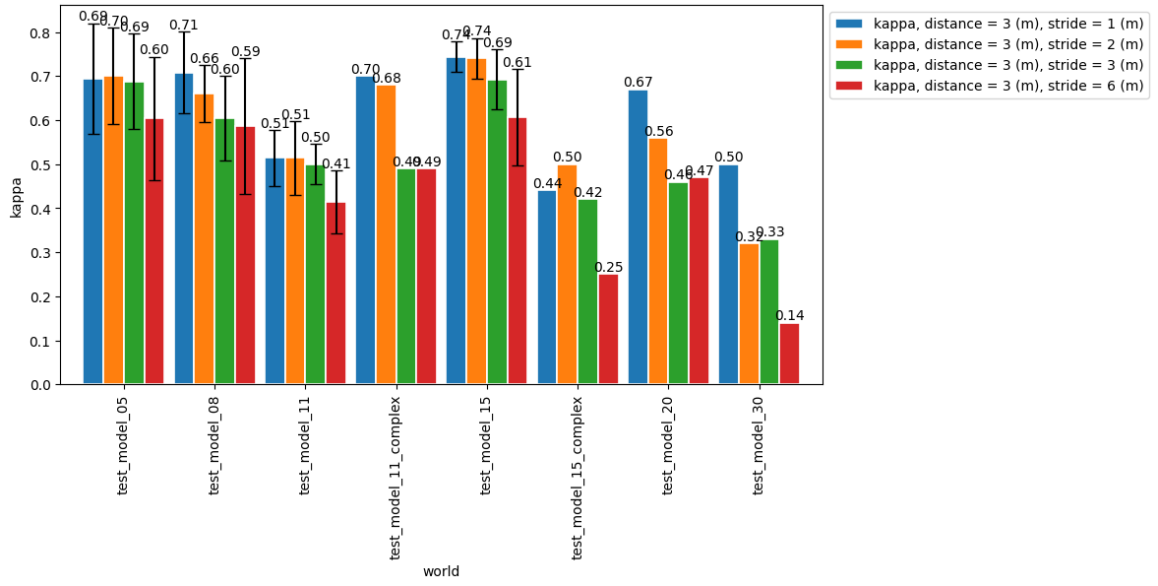
(a)  $\kappa$  en fonction de la densité du monde.



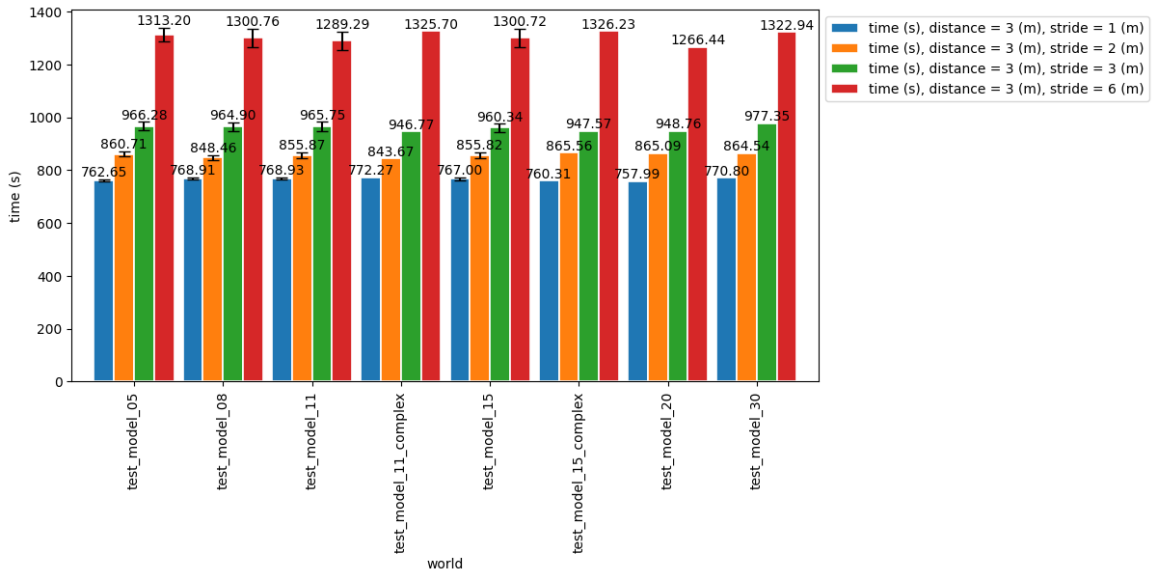
(b) Temps d'exécution en fonction de la densité du monde.

FIGURE 10 – Évolution du  $\kappa$  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.

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.



(a)  $\kappa$  en fonction de la densité du monde.

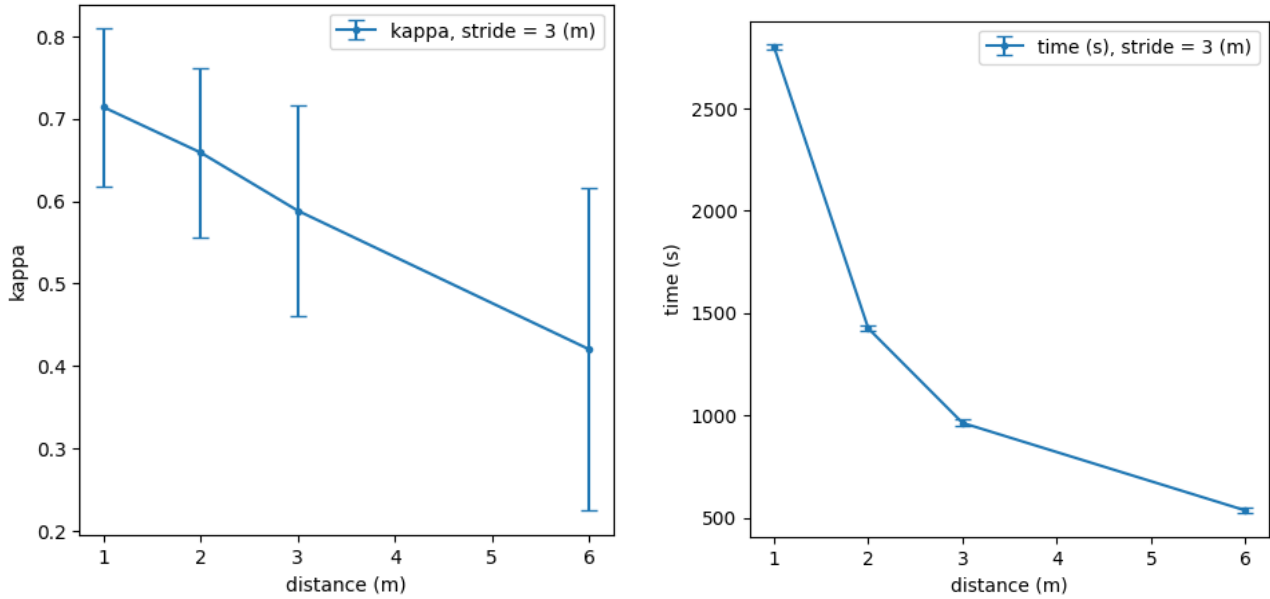


(b) Temps d'exécution en fonction de la densité du monde.

FIGURE 11 – Évolution du  $\kappa$  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.

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 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.



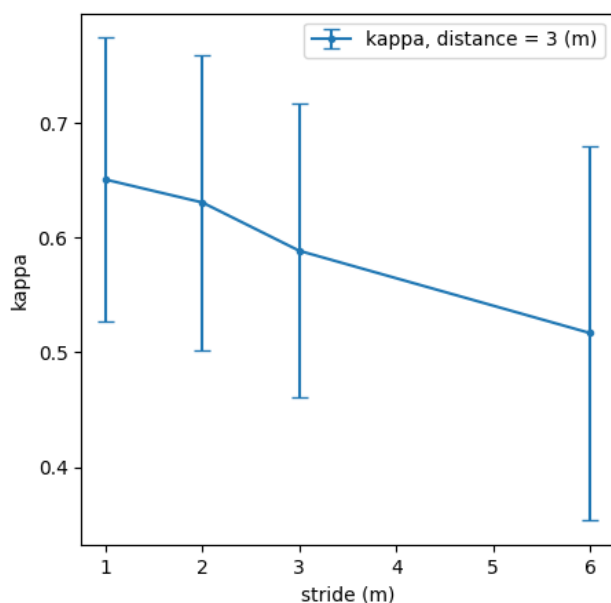
(a)  $\kappa$  en fonction de la distance entre les deux crawlers.

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

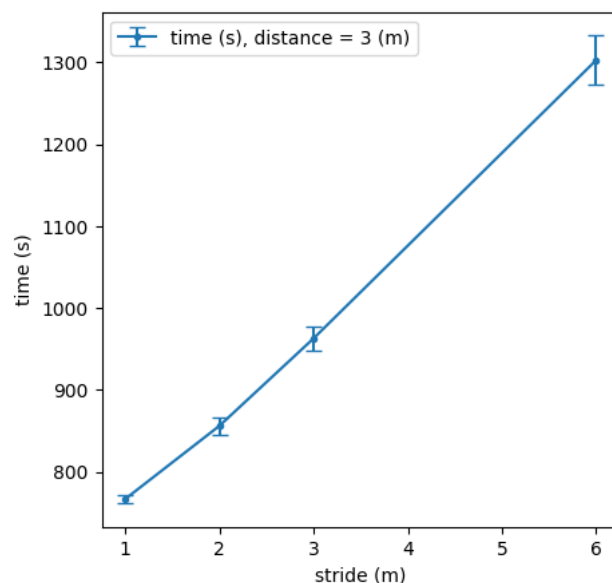
FIGURE 12 – Évolution du  $\kappa$  de Cohen et du temps d'exécution de l'algorithme *ski nordique* en fonction de la distance qui sépare les deux crawlers.

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



(a)  $\kappa$  en fonction du pas des crawlers.



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

FIGURE 13 – Évolution du  $\kappa$  de Cohen et du temps d'exécution de l'algorithme *ski nordique* en fonction du pas des crawlers.

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.

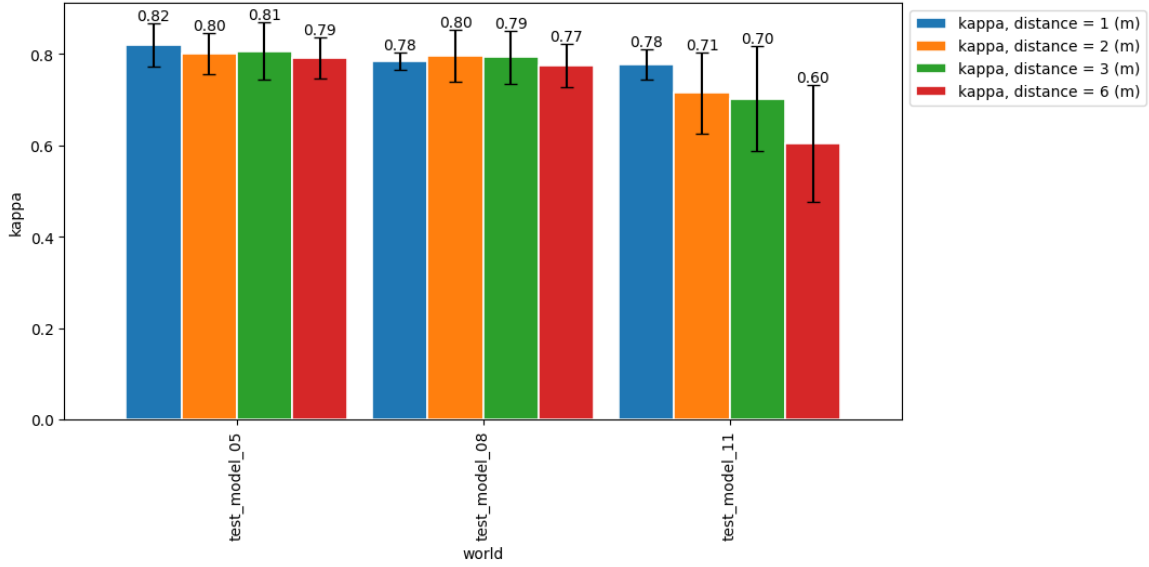
Ici encore, il semble que le score et le temps d'exécution ne soient pas affectés par le fait que les formes soient complexes ou non.

Dans la suite de ce rapport, nous considérerons une distance  $d = 3$  entre les deux crawlers et un pas  $s = 3$  pour l'algorithme *ski nordique*.

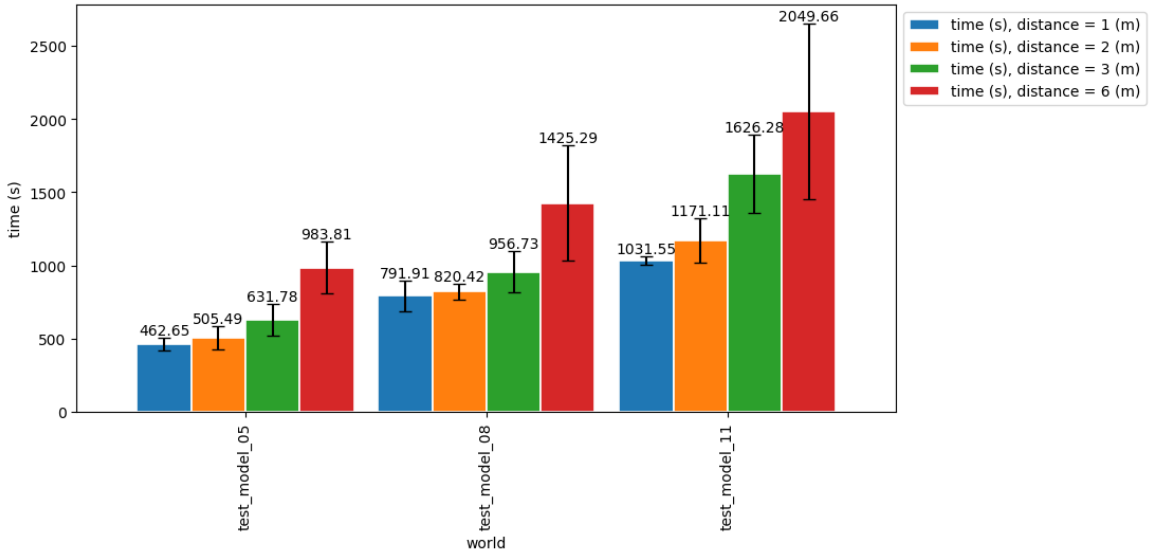
## Stratégie de navigation *investigation polygonale*

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.



(a)  $\kappa$  en fonction de la densité du monde.



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

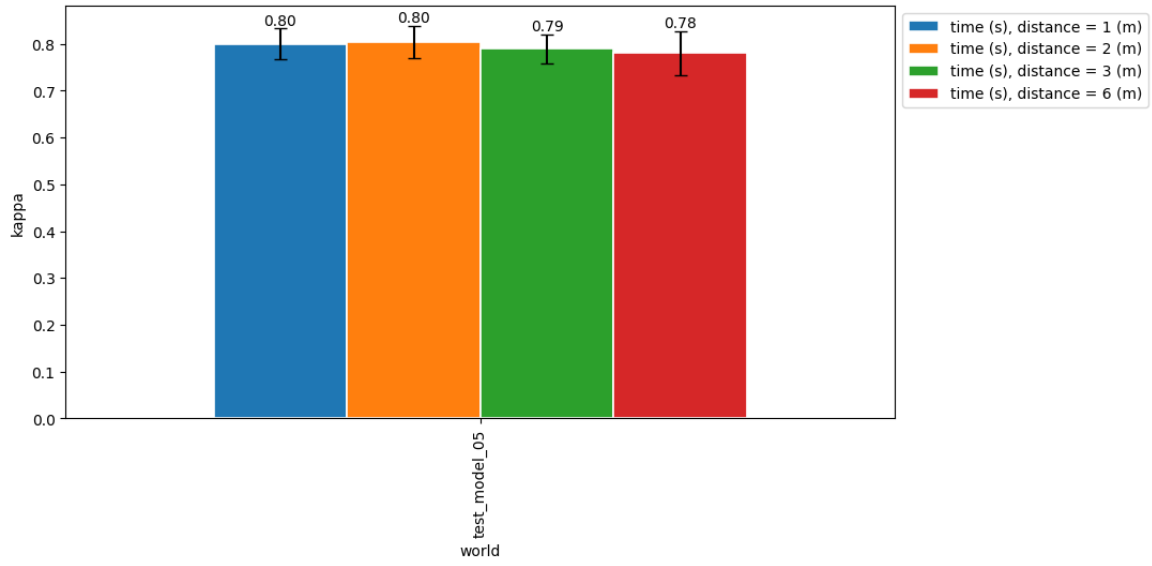
FIGURE 14 – Évolution du  $\kappa$  de Cohen et du temps d'exécution de l'algorithme *investigation polygonale* en fonction de la densité du monde pour différentes distances entre les crawlers avec un polygone à 4 côtés.

La figure 14a montre l'évolution du score de Cohen en fonction de la densité du monde pour

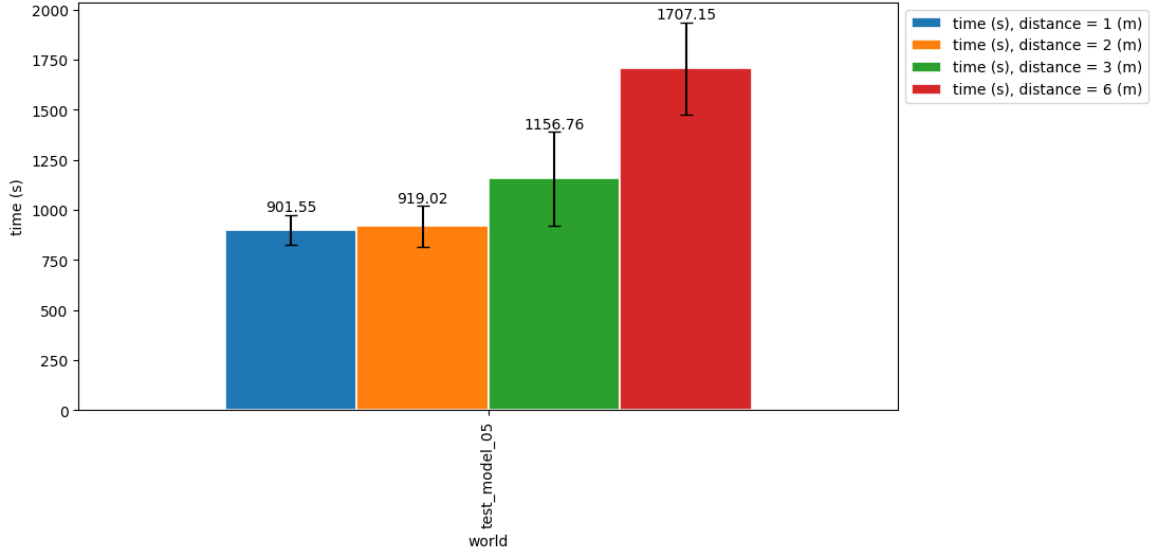
chaque valeur de  $d$  utilisée dans la stratégie *peinture au rouleau*. Nous avons utilisé un polygone d'investigation à 4 côtés. En premier lieu, nous observons des scores de Cohen relativement indépendant de la distance entre les crawlers pour les cartes avec 5 et 8 zones de corrosion. Ceci est un 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. 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



(a)  $\kappa$  en fonction de la densité du monde.



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

FIGURE 15 – Évolution du  $\kappa$  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.

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.

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.

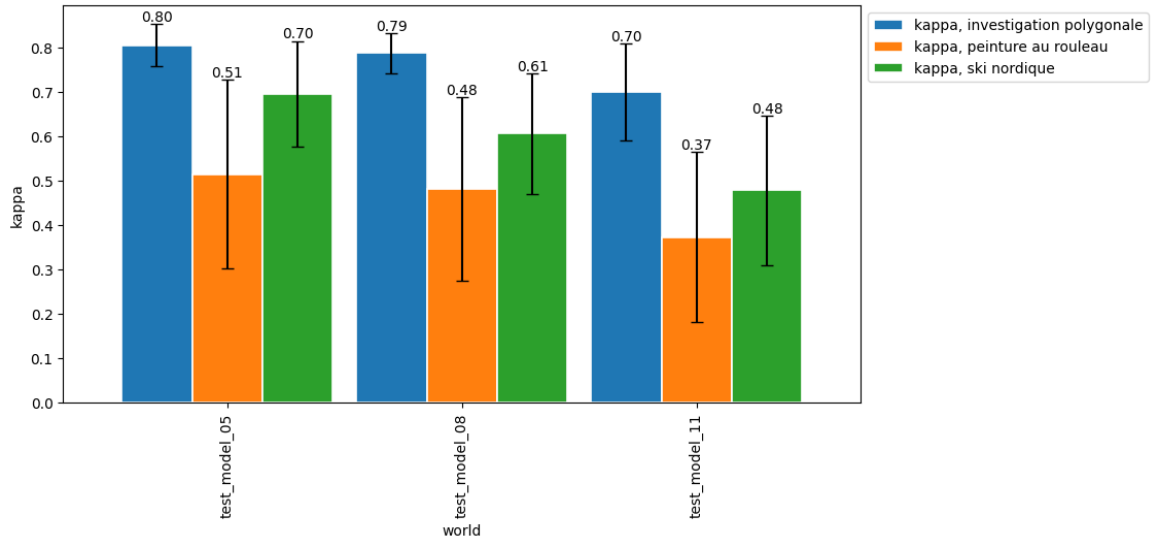
Nous présentons sur la figure 16 l'évolution du score de Cohen et du temps d'exécution en fonction de la densité du monde, pour chaque algorithme. Dans cette figure, nous avons représenté les scores et temps d'exécution obtenus, en moyenne, pour les différentes valeurs de  $d$  et de  $s$ , par soucis 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.

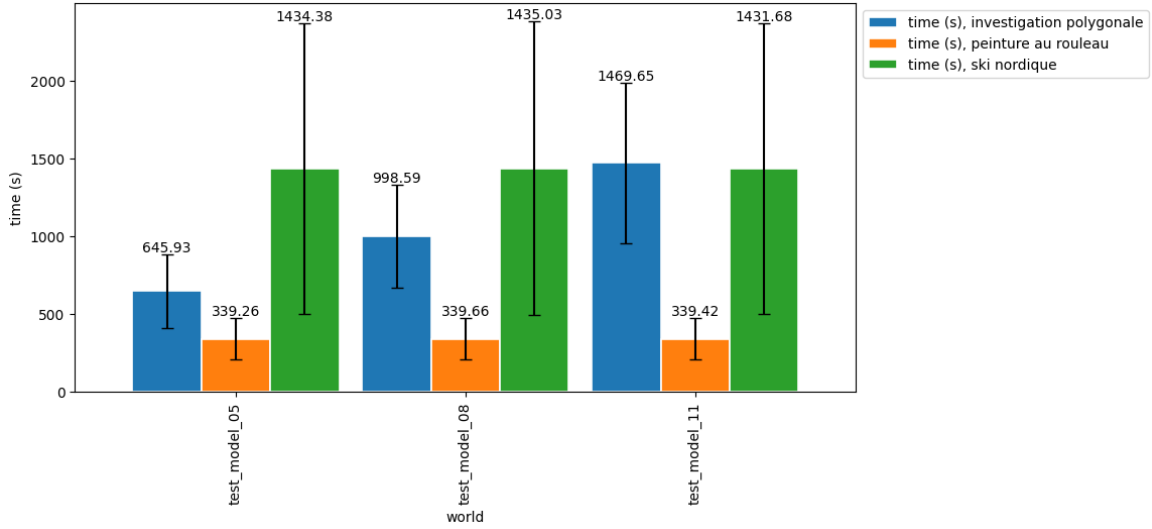
Nous pouvons observer sur la figure 16b que le temps d'exécution de l'algorithme *peinture au rouleau* est inférieur à ceux obtenus par les algorithmes *investigation polygonale* et *ski nordique*. Pour de faibles densités de cartes comme celles utilisées dans nos expérimentations, l'algorithme *ski 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	$\kappa$ 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*.



(a)  $\kappa$  en fonction de la densité du monde.



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

FIGURE 16 – Évolution du  $\kappa$  de Cohen et du temps d'exécution des différents algorithmes en fonction de la densité du monde pour différentes distances entre les crawlers.

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.

## Références

- [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 Charpillat. 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](http://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](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](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](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](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](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.

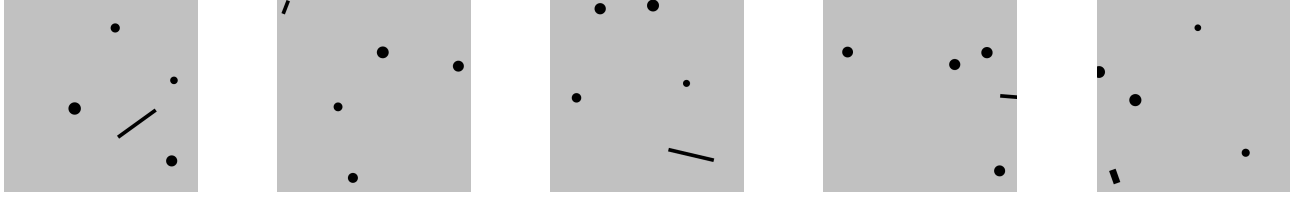
- [21] Gurobi. Traveling salesman problem. [https://colab.research.google.com/github/Gurobi/modeling-examples/blob/master/traveling\\_salesman/tsp\\_gcl.ipynb#scrollTo=-W\\_zXnoRaaao](https://colab.research.google.com/github/Gurobi/modeling-examples/blob/master/traveling_salesman/tsp_gcl.ipynb#scrollTo=-W_zXnoRaaao). [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](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](https://www.mathworks.com/matlabcentral/fileexchange/31814-mdmtspv_ga-multiple-depot-multiple-traveling-salesmen-problem-solved-by-genetic-algorithm). Dernier accès : 2023-06-02.



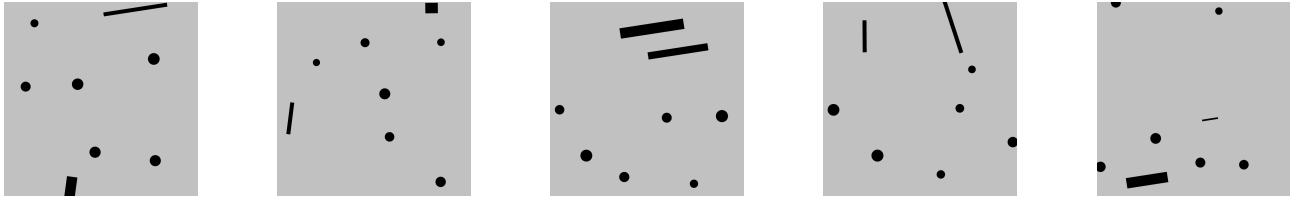


# Annexes

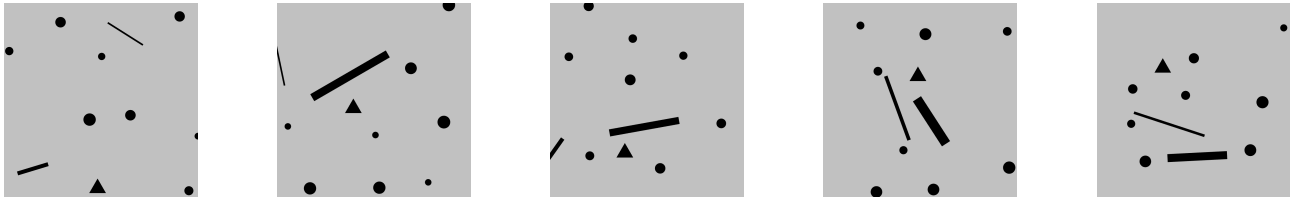
## Annexe A: Environnements de test



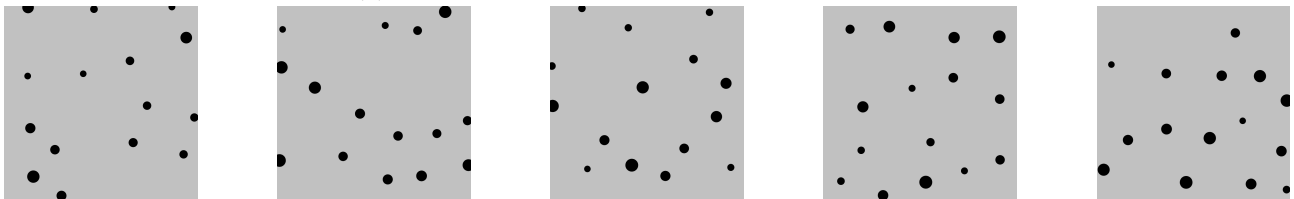
(a) Mondes de test avec 5 zones de corrosion.



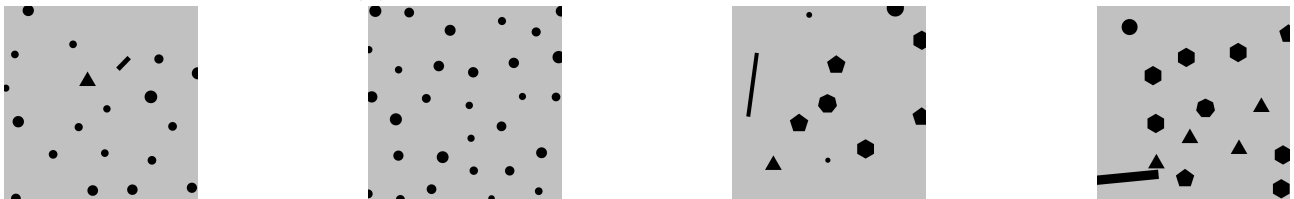
(b) Mondes de test avec 8 zones de corrosion.



(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.

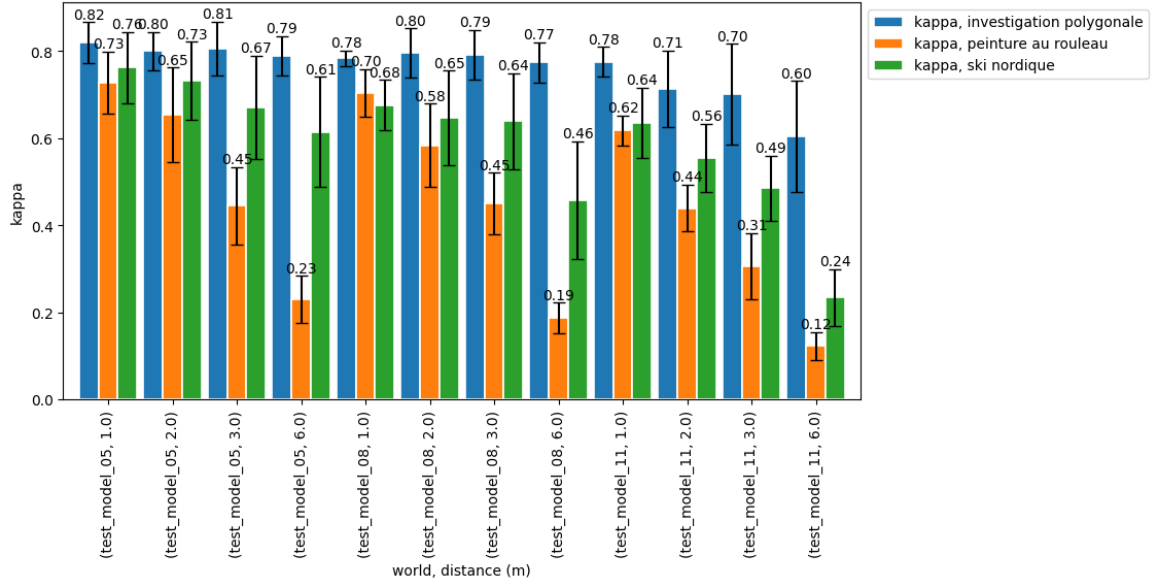
(f) Mondes de test avec 30 zones de corrosion.

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

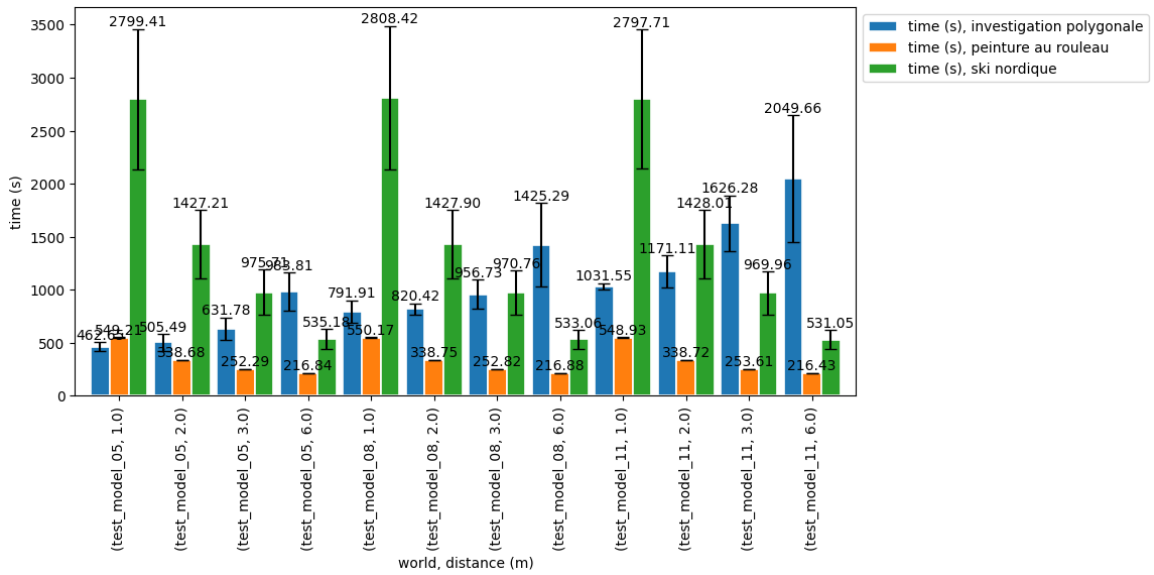
(h) Mondes de test avec 15 zones de corrosion complexes.

FIGURE 17 – Différents environnements de test.

## Annexe B: Comparaison des différentes stratégies de navigation



(a)  $\kappa$  en fonction de la densité du monde.



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

FIGURE 18 – Évolution du  $\kappa$  de Cohen et du temps d'exécution des différents algorithmes en fonction de la densité du monde pour différentes distances entre les crawlers.

## Annexe C: Résultat d'investigation

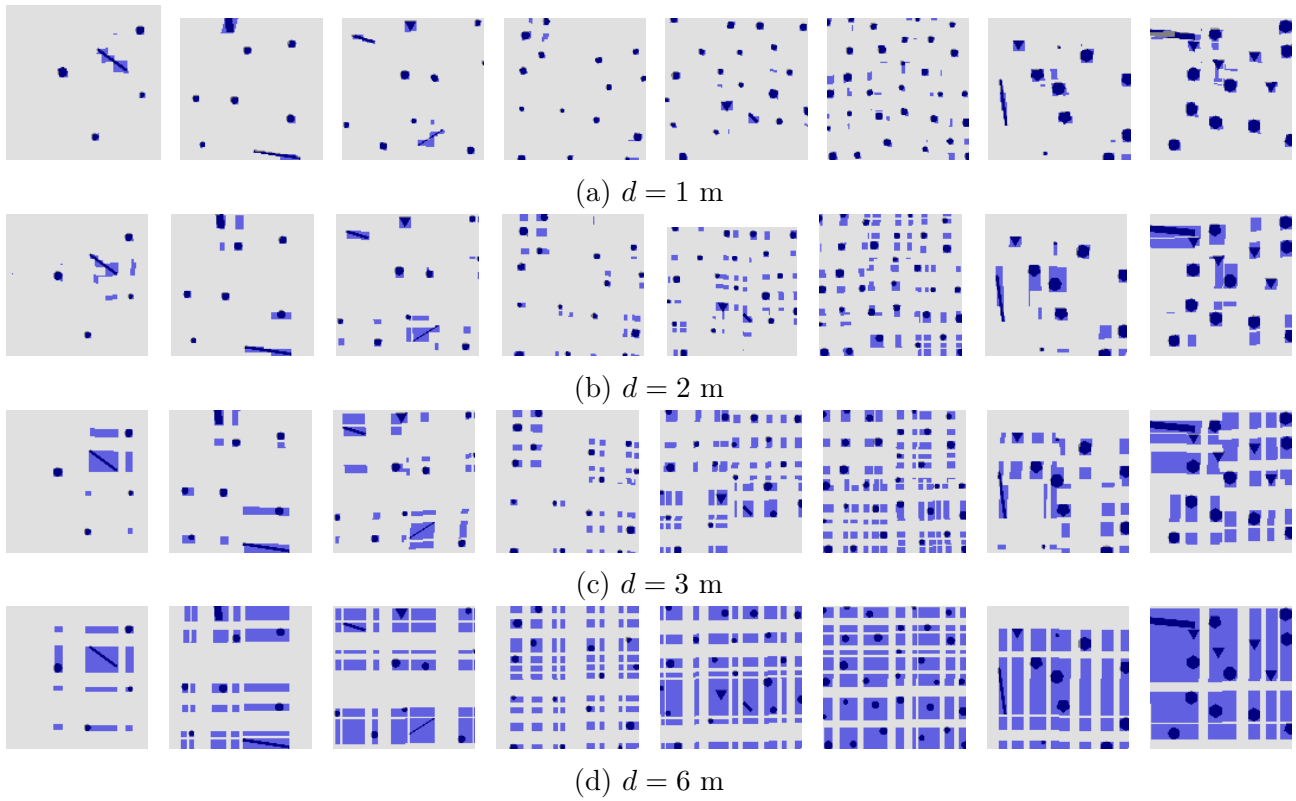


FIGURE 19 – Superposition des cartes d'investigation avec la cartographie des zones de corrosion obtenue pour les différents mondes de test, pour la méthode *peinture au rouleau*.

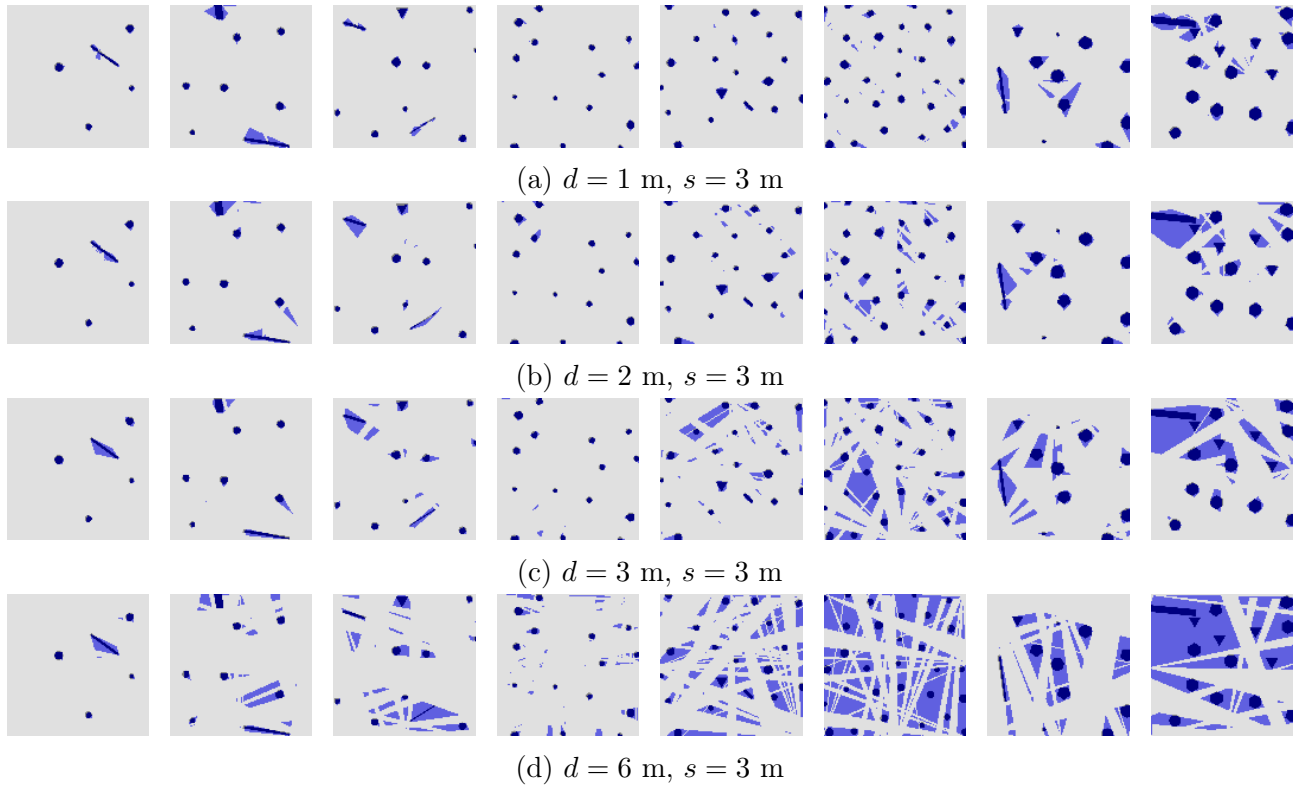


FIGURE 20 – Superposition des cartes d’investigation avec la cartographie des zones de corrosion obtenue pour les différents mondes de test, pour la méthode *ski nordique* - 1.

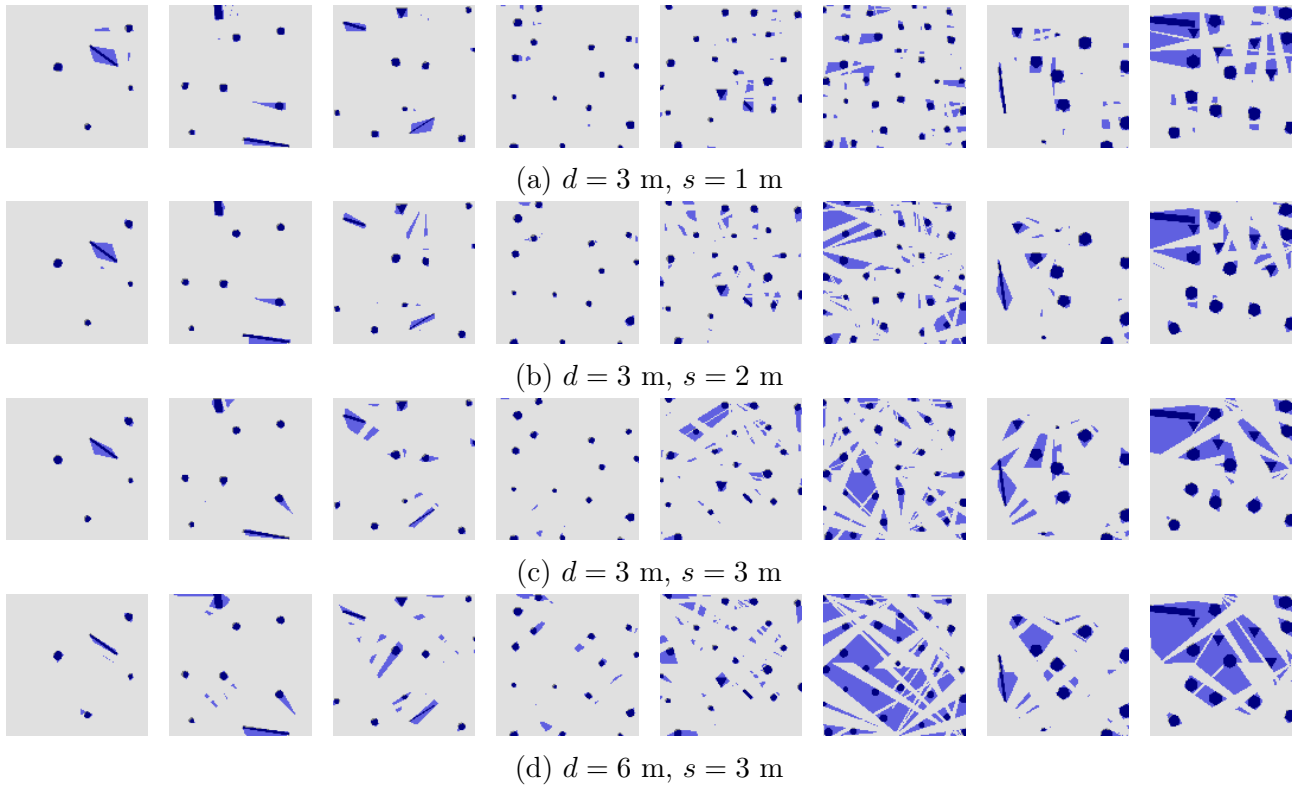
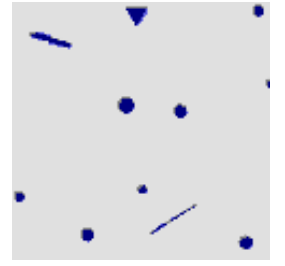
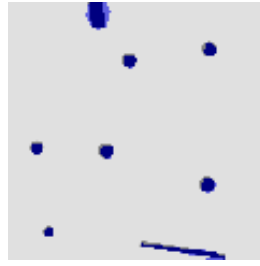
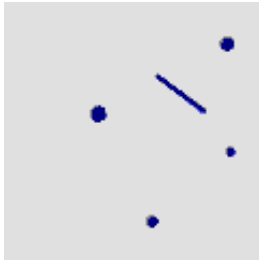
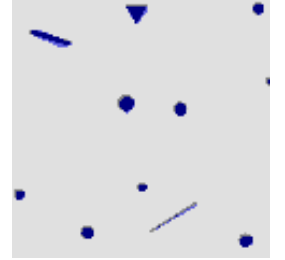
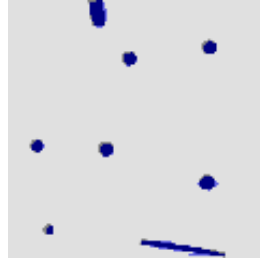


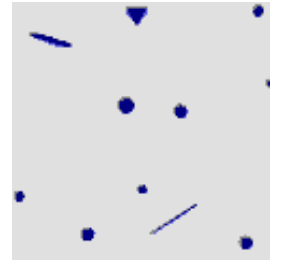
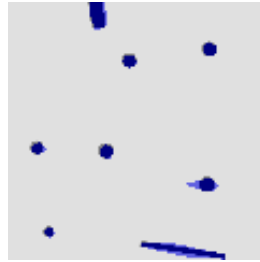
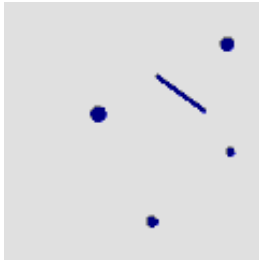
FIGURE 21 – Superposition des cartes d’investigation avec la cartographie des zones de corrosion obtenue pour les différents mondes de test, pour la méthode *ski nordique* - 2.



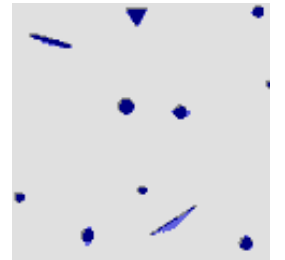
(a)  $k = 1, n = 2, p = 4, d = 1 \text{ m}$



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



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

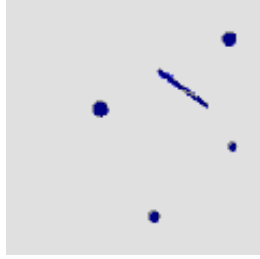


(d)  $k = 1, n = 2, p = 4, d = 6 \text{ m}$

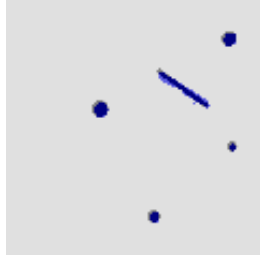
FIGURE 22 – Superposition des cartes d'investigation avec la cartographie des zones de corrosion obtenue pour les différents mondes de test, pour la méthode *investigation polygonale* - 1.



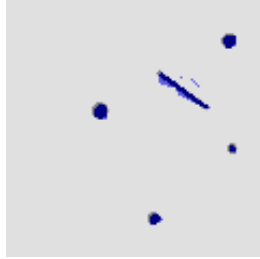
(a)  $k = 1, n = 2, p = 6, d = 1$  m



(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.





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

Brandon ALVES

## 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.

## 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.