

CONCEPTION ET OPTIMISATION D'ALGORITHMES DE COORDINATION  
MULTI-ROBOT POUR L'EXPLORATION AUTONOME DE RÉSEAUX DE GALERIE.

# I Introduction

Definition d'un robot, Spécification du type étudié dans ce rapport, les défis de la planification de trajectoire, l'évitement d'obstacle (petit point sur l'optimisation de trajectoire en milieu ouvert pour de la recherche de victime par exemple) Les contraintes extérieures et celle du robot lui même.

L'un des ouvrages de référence utilisés dans ce travail est le livre de S. M. LaValle, *Planning Algorithms* [1], que je remercie particulièrement pour son engagement à rendre ses travaux accessibles au public (<https://lavalle.pl/>).

## II Etat de l'art

### II.1 État de l'Art : Exploration Multi-Robot de Cavités

L'exploration des cavités est essentielle pour la cartographie souterraine, la recherche scientifique et les interventions d'urgence, tout en posant des défis uniques pour la robotique autonome.[2] Leur exploration multi-robot est un domaine de recherche en plein essor dans le domaine de la robotique, avec des applications dans des environnements variés tels que l'exploration spatiale, l'exploration de grottes, les missions de sauvetage dans des environnements urbains sinistrés, ou encore l'extraction minière.[3, 4] Ces environnements, souvent complexes et dynamiques, présentent de nombreux défis pour la planification de mission, notamment des obstacles imprévisibles, des zones inaccessibles pour les robots mobiles, et l'absence de signal GPS. De ce fait, l'exploration multi-robot permet de tirer parti de la coopération entre robots pour surmonter ces difficultés.

#### II.1.1 Défis dans l'exploration multi-robot de cavités

L'exploration de cavités avec plusieurs robots implique plusieurs défis techniques. En particulier, la gestion de la coopération entre robots et la gestion de l'information dans des environnements complexes sont deux aspects fondamentaux de ce type de mission. Les défis peuvent être classés en plusieurs catégories :

### II.2 État de l'Art des Techniques de SLAM

Le Simultaneous Localization and Mapping (SLAM) est un problème clé en robotique qui permet à un robot de localiser sa position tout en construisant une carte de son environnement. Plusieurs techniques ont été développées pour résoudre ce problème, telles que le EKF-SLAM, qui utilise un filtre de Kalman pour estimer la localisation et la carte, mais souffre de limitations en termes de scalabilité dans de grands environnements. Le FastSLAM, basé sur un filtre de particules, améliore cette scalabilité en traitant les caractéristiques du monde à l'aide de multiples hypothèses. Les approches basées sur les graphes, comme le Graph-SLAM, sont efficaces pour résoudre des problèmes d'optimisation à grande échelle, bien qu'elles nécessitent une gestion complexe des données. Le SLAM visuel (V-SLAM) repose sur des caméras pour estimer la localisation et construire une carte, tandis que le SLAM LiDAR utilise des capteurs laser pour des mesures de profondeur précises, particulièrement efficaces pour des environnements extérieurs. Des variantes comme le SLAM dynamique gèrent les environnements avec des objets mobiles en excluant ces derniers des mises à jour de la carte. Ces méthodes varient en fonction des environnements, des ressources du robot et des contraintes en temps réel, avec pour objectif de maintenir une localisation et une cartographie précises dans des conditions dynamiques et complexes.

### II.3 État de l'Art des Méthodes de Communication Inter-Robot dans un Environnement de Cavité

La communication inter-robot (CIR) dans un environnement de cavité est un défi majeur pour les systèmes multi-robots, en raison des conditions particulières que ces environnements présentent, telles que des espaces confinés, des obstacles physiques et des perturbations qui peuvent affecter les signaux de communication. Les méthodes de CIR peuvent être classées en différentes catégories selon la technologie utilisée et la stratégie de communication adoptée. Les approches classiques reposent principalement sur des réseaux sans fil, tels que la communication par radiofréquence (RF), qui est couramment utilisée pour des applications de communication à longue portée mais

qui peut souffrir de limitations dans les cavités où les signaux sont atténués par des parois solides. Dans ce contexte, des méthodes de communication ad hoc sont souvent employées, où les robots créent un réseau dynamique de relais pour échanger des informations. Une approche courante est l'utilisation de la *communication par maillage*, dans laquelle chaque robot agit comme un relais, permettant ainsi une couverture étendue et une transmission de données entre robots même lorsque les obstacles interfèrent avec les signaux directs. La *communication acoustique* ou *ultrasonique*, qui repose sur des ondes sonores, est une alternative viable dans les environnements de cavité, où elle peut transmettre des informations de manière robuste, notamment pour des distances plus courtes ou dans des environnements très confinés. De plus, la communication optique (par exemple, la *communication par lumière visible ou infrarouge*) devient de plus en plus populaire dans des applications de haute précision dans des cavités étroites, car elle est peu sensible aux interférences électromagnétiques et permet des transmissions rapides et sécurisées. Une autre stratégie consiste à utiliser des protocoles de communication coopérative, où les robots collaborent pour optimiser le flux d'informations. Des techniques de *routage adaptatif* et de *partitionnement dynamique de réseau* sont souvent utilisées pour gérer les obstacles et les interférences dans les réseaux multi-robots. Enfin, les méthodes de *communication opportuniste*, basées sur des échanges de données ponctuels lorsque la ligne de visée est dégagée, sont particulièrement adaptées aux environnements où la connectivité est intermittente et où les robots doivent s'organiser pour minimiser les interruptions dans le flux de données. L'un des principaux défis reste de maintenir une communication fiable et efficace, notamment en optimisant les stratégies de répartition de la bande passante, d'allocation des ressources et de gestion des interférences dans des espaces très contraints. Des approches récentes cherchent à intégrer des systèmes hybrides combinant ces technologies pour assurer une meilleure résilience face aux conditions environnementales changeantes et aux mouvements des robots.

### 11.3.1 Planification de chemin, de trajectoire et évitement d'obstacle

La planification de chemin pour un système mécatronique constitue le fondement de tous les systèmes mobiles autonomes, qu'il s'agisse de drones ou de bras de manutention et d'assemblage. Le principe de la planification de chemin ou de trajectoire est de déterminer une solution - un chemin ou une trajectoire - reliant un point de départ à un point cible.

La distinction entre planification de chemin et planification de trajectoire réside dans le fait qu'un chemin planifié n'est pas nécessairement réalisable par un robot. En effet, la planification de chemin ne prend pas toujours en compte la faisabilité physique ou cinématique pour un robot mobile.

L'évitement d'obstacles est une contrainte essentielle dans ces deux approches. Les obstacles définissent les zones inaccessibles, et comme nous le verrons par la suite, leur nature - mobile ou immobile - détermine en grande partie la méthode à employer pour résoudre le problème de planification.

Parmi les méthodes déterministe, on trouve une large variété de méthodes principalement basées sur trois approches différentes. Les approches par graphes, celle de décomposition cellulaire et enfin les celles utilisant des champs potentiels.[5, 6]

Les méthodes des graphes consistent à construire une carte des chemins empruntable en partant des obstacles de la scène. Parmi ces méthodes utilisant des graphes, on peut distinguer quatre types différents : Les graphes de visibilité[7], les diagrammes de Voronoï [8] ou encore la méthode des Silhouettes[6].

Les méthodes associées au décomposition cellulaire consistent à diviser l'espace libre du robot en régions simples, appelées cellules, où il est facile de générer un chemin entre deux configurations. Un graphe représentant les relations d'adjacence entre les cellules est ensuite construit et exploré.[5,

9, 10, 11]

Une autre méthode repose sur une subdivision fine de l'espace afin de repérer les zones libres. La méthode des champs potentiels s'appuie sur cette idée en définissant des potentiels qui traduisent des forces d'attraction, dirigées vers les coordonnées cibles, et de répulsion, correspondant par exemple aux obstacles. Le chemin est ensuite déterminé en suivant l'opposé du gradient du potentiel total ainsi calculé.[5, 12]

Des approches alternatives ont été développées dans les années 1990 et 2000, notamment les méthodes stochastiques de planification de chemin RPP (*Random Path Planners*) et PRM (*Probabilistic Roadmap Planners*). Ces méthodes consistent à échantillonner l'espace de manière aléatoire afin de créer un graphe de chemins possibles (*roadmap*) dans cet espace.[13, 14, 15]

L'un des principaux atouts de cette méthode de planification de chemin est qu'elle ne dépend ni de la structure de l'espace, ni du nombre d'obstacles, ni de leur disposition. Une fois la *roadmap* créée, il suffit d'utiliser une méthode de recherche de chemin dans un graphe pour trouver le chemin menant au point objectif.[16]

### III Définitions mathématiques

Soit  $\Omega \in \mathbb{R}^2$ , soit  $\mathbf{X}(t)$  les coordonnées du robot à l'instant  $t$  dans cet espace.

Soit  $V(t)$  le champ de vision du robot à l'instant  $t$

Soit  $\mathbf{M}(t, \theta)$  le premier point d'intersection entre un segment et les murs,

$$\mathbf{M}(t, \theta) = \mathbf{X}(t) + R(t, \theta) \begin{pmatrix} \cos \theta & 0 \\ 0 & \sin \theta \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

On note  $\mathbf{M}_{max}(t, \theta)$  le point tel que  $R(t, \theta) = R_{max}$

Ici,

$$R(t, \theta) = \min(\text{distance}(\mathbf{X}(t), L(\theta) \cap W))$$

$$L(\theta) = \{ (1-l)\mathbf{X}(t) + l\mathbf{M}_{max}(t, \theta) \mid l \in [0, 1] \}$$

$$W = \{ \text{Segment}(\Omega) \}$$

On défini le champ de vision du robot tel que:

$$V(t) = \{ (1-l)\mathbf{X}(t) + l\mathbf{M}(t, \theta) \mid l \in [0, 1], \theta \in [0, 2\pi[ \} \quad (1)$$

Autrement dis,  $V(t)$  est l'ensemble des points de  $\Omega$  présents dans un disque de rayon  $R_{max}$  et situé entre le robot et la plus proche intersection à un mur

On construit maintenant la fonctionnelle que l'on cherchera plus tard à optimiser, celle-ci est relative au déplacement du robot.

On défini  $KM$  (*Known Map*) l'espace de la map connu par le robot et  $EM$  (*Explorable Map*) la partie de  $\Omega$  explorable par le robot.

$$J(\mathbf{X}(t)) = \int_0^T |\dot{\mathbf{X}}(t)| dt \quad (2)$$

$J$  est une fonctionnelle fonction de la position initiale du robot, elle donne la longueur du chemin parcourue par le robot avant que  $t = T$ .

$T$  étant l'instant à partir du quel la map est explorée au maximum des capacités du robot, c'est à dire :  $KM = EM$ .

Pour savoir si  $KM = EM$ , on calcul les contours du domaine connu, si tous les contours sont fermés, alors  $KM \subset EM$ , si de plus les mesures de  $KM$  et de  $EM$ , à savoir leur surface, sont égales alors, on peut raisonnablement dire que  $KM = EM$ .

Pour plus de simplicité lors de l'étude de la fonctionnelle, on la modifie légèrement.

$$J(\mathbf{X}(t)) = \frac{1}{2} \int_0^T \dot{\mathbf{X}}(t)^2 dt \quad (3)$$

## IV Avancement

### IV.1 1ère semaine :

- Recherche documentaire autour des méthodes pour définir l'espace mathématique dans lequel le robot évolue
- Recherche documentaire autour des méthodes d'exploration multi-robot
- Définition de l'espace étoilé relatif champ de vision du robot
- Définition de la fonctionnelle du robot
- 

### IV.2 2ème semaine :

- Début de la rédaction de l'état de l'art autour des méthodes de planification de trajectoire
- Définition de la fonctionnelle  $J$
- Définition de la borne supérieure de l'intégrale de  $J$
- Correction du comportement du lidar pour l'exploration de la map
- Implémentation des collisions du robot avec la map
- 

Note sur les LiDAR (*Wikipedia*):

La télédétection par laser ou LIDAR, acronyme de l'expression en langue anglaise « light detection and ranging » ou « laser imaging detection and ranging » (soit en français « détection et estimation de la distance par la lumière » ou « par laser »), est une technique de mesure à distance fondée sur l'analyse des propriétés d'un faisceau de lumière renvoyé vers son émetteur.

À la différence du sonar qui utilise des ondes acoustiques et du radar qui emploie des ondes électromagnétiques de plus basse fréquence (ondes radio), le lidar utilise des ondes électromagnétiques proches de la lumière visible (du spectre visible, infrarouge ou ultraviolet). La lumière utilisée par le lidar est presque toujours issue d'un laser, ce qui permet d'avoir une source lumineuse directionnelle, monochromatique, polarisée, de haute amplitude et cohérente.

Le principe de la télémétrie (détermination de la distance d'un objet), qui concerne une grande partie des applications du lidar, requiert généralement l'utilisation d'un laser impulsif. La distance est donnée par la mesure du délai (aussi appelé « temps de vol ») entre l'émission d'une impulsion et la détection d'une impulsion réfléchie, connaissant la vitesse de la lumière. Quand la source laser est entretenue et modulée en fréquence, on parle de lidar FMCW (Frequency Modulated Continuous Wave).

#### IV.2.1 Recherche à faire

Faire un histogramme des temps de recouvrement de la map en définissant plusieurs critères ou de manière aléatoire afin de dégager des comportements qui semblent "optimaux", recommencer afin de trouver le minimum global ou du moins local de manière expérimentale.

Implémentation de l'évitement d'obstacle.

Approfondir la compréhension autour de FMM et des équations Eikonales

Regarder comment le relier avec un portrait de phase, équation différentielle. Tracer la trajectoire  $x_1$  en fonction de la trajectoire  $x_2$ . Portrait de phase de système dynamique aléatoire ? Est ce possible ?

Système dynamique avec contrôle (optimal dans la suite)

## V Multi-robots control

This algorithm aims to coordinate the exploration of unknown area by a robot swarm.

### V.1 Operation

At the beginning of the exploration, we assign a number to each robot, the smallest one is the master. Another Implementation would be to choose the master accordingly to their battery level, the one with the higher battery level is the master. Indeed, the master robot is the one that consumes more due to all the calculation it will do. To keep it simple, we choose the smallest number.

The communication is based on a strong hierarchical structure, the master robot gives the direction to all robot links with it with a bigger number.

If the group splits, the master changes to the strongest robot in the group. Robots communicate with each other using light, detecting if the robot can communicate with another is made, in a first time, by ensuring visual contact. A more powerful approach is to simulate the travel of the light in the medium.

The master of a group is a hub for communication, each new direction is given by him using this method :

- Each robot in a group sends the master any part of the map that is new to it
- Each robot of a group sends the master the waypoint with the open frontier index it wants to explore.
- Once all robots of the group send its information to the master, the protocol for gathering information is given later, it computes the normalized cost table for each combination robot-waypoint. Values are given between 0 and 255 to send only one byte information ensuring low communication volume.

	Robot 1	Robot 2	Robot 3	Robot 4	Robot 5
<b>WP 1</b>	<b>23</b>	87	234	56	192
<b>WP 2</b>	245	76	<b>11</b>	68	39
<b>WP 3</b>	90	<b>21</b>	73	<b>50</b>	164
<b>WP 4</b>	132	58	49	77	<b>25</b>
<b>WP 5</b>	181	<b>13</b>	66	<b>39</b>	70

Table V.1: Example of costs table for a group of 5 robots with 5 different waypoints

In the case, there is less waypoint than robot, the group splits in two groups.

	Robot 1	Robot 2	Robot 3	Robot 4	Robot 5
<b>WP 1</b>	<b>61</b>	125	93	47	<b>59</b>
<b>WP 2</b>	88	<b>12</b>	<b>53</b>	<b>29</b>	174

Table V.2: Example of costs table for a group of 5 robots with 2 different waypoints



In the case, there is less robot than waypoint, each robot explore a zone.

	Robot 1	Robot 2	Robot 3	Robot 4	Robot 5
<b>WP 1</b>	<b>42</b>	134	212	63	189
<b>WP 2</b>	215	98	<b>19</b>	75	48
<b>WP 3</b>	102	<b>32</b>	95	68	141
<b>WP 4</b>	142	74	58	89	<b>31</b>
<b>WP 5</b>	193	<b>27</b>	78	<b>54</b>	83
<b>WP 6</b>	156	63	<b>17</b>	99	115
<b>WP 7</b>	173	49	132	82	<b>23</b>
<b>WP 8</b>	204	<b>57</b>	146	71	94

Table V.3: Example of costs table for a group of 5 robots with 8 different waypoints

- The master, after computing the cost table, distribute the waypoint across the group minimizing the cost combination. For instance, in the Table V.1, Table V.2 and Table V.3, bold number are the minimum cost for each robot and each waypoint, however the minimum combination of cost is given by the blue one.

### V.1.1 Meeting of two groups

If 2 groups met, both master share all the information they have without moving. Once the transfert is done, the strongest master take control of all the group and continue the exploration.

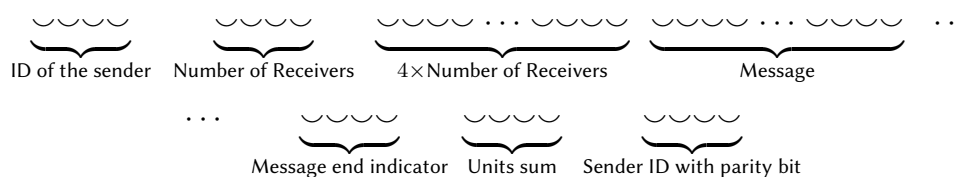
## V.2 Inter-robot communication protocol

For each robot one at a time in the group. The master asked for informations. Listen for the answer. If an answer is given, the master validate the tranfert of information. Else, retry, retry and retry and skip it. Compute thing. Give an order to the robot. The robot listen for the order. If an answer is received, the robot validate the tranfert of information. Else, retry, retry and retry and skip it.

## V.3 Encoding

To encode the message, we use a specific message frame. This frame ensures that if part of the message is incorrectly transmitted, such as when a bit in the message is flipped, the receiver will detect the error and request the message to be resent until it is received correctly. This method is known as Automatic Repeat reQuest (ARQ).

The message includes two verification mechanisms, unit sum check and parity bit: one to check the integrity of the transmitted message and another to confirm the identity of the sender.



### V.3.1 Description

**Emitter ID:** 〰〰〰〰

The unique 4-bit identifier for the sender of the message (range: 0-15).

**Number of Receivers:** 〰〰〰〰

Optional 4-bit field indicating the number of receivers. If there is only one receiver, this field is set to 0000.

**Receiver IDs:** 〰〰〰〰 ... 〰〰〰〰

Each receiver's ID is encoded in 4 bits. For multiple receivers, these IDs are listed sequentially.

**Message Content:** 〰〰〰〰 ... 〰〰〰〰

Type indicators (up to 15 indicator types):

Type indicator	Type	Data size
0000	Message end	4 bits
0001	Integer	8 bits
0010	Integer	32 bits
0011	Float	32 bits
0100	String	8 bits for length + 8 bits per character
0101	List of int8	16 bits for length + 8 bits per integer
0110	List of float	16 bits for length + 32 bits per float
0111	Message type ID	8 bits
1000	Robot info	168 bits
1001		
1010	Binary message	16 bits for length + 1 bit per binary bit
1011		
1100		
1101		
1110		
1111		

Table V.4

**Units sum:** 〰〰〰〰

4 bits representing the sum of the bit in the message frame modulo 16.

**Emitter ID with Parity:** 〰〰〰〰

The emitter ID is repeated with a parity bit. The parity ensures the total number of 1s in the message frame up to the emitter ID with parity is correct. If odd, the last bit is flipped.

## V.4 Decoding

## V.5 Implementation

To implement it, we define a table of message with their specification, for each message received, there is a frame of response with some test.

Message ID table:

Message ID	Description
00000000	Ask to repeat the last message
00000001	Last message received correctly
00010000	Ask the connected robot set
00010001	Send the connected robot set
00100000	Ask to all where they want to go
00100001	Send next waypoint position with frontier ID
00100010	Send robot info
00100011	Send next position to reach
00110000	Ask for live grid map synchronisation
00110001	Send updated live grid map
01110000	Ask info gained since last connection
01110001	Send nothing gained
01110010	Send what was gained on the live-grid map
01010101	Transmit message from another robot

Information sent for each message type could be found in the **Appendix ...**

Robot info: 2 *float32* for position, 1 *float32* for velocity, 1 *float32* for rotation, 1 *float32* for angular velocity, 1 *int8* for energy

Connected robot set: 4 bits for set length + 4 bits per robot id

## V.6 Analysis scheme

### V.6.1 Delimited range of application

Can be for all type of environment

### V.7 Robustness to failure and uncertainty

Must handle all possible cases, for instance if a robot is in a dead lock and doesn't succeed in exit it

### V.8 Completeness

Must explore all the domain or accomplish all the task given to it

### V.9 Effectiveness

Must spread the work efficiently

### V.10 Speed

Must be a rapid algorithm

## **VI Partie à développer**

### **Création du monde**

- Automate cellulaire
- Méthode de square marching appliquée à des triangles

### **Capteurs utilisés**

Minimalisme, quels capteurs sont nécessaire et suffisant pour accomplir la mission donnée ?

- Fonctionnement d'un lidar
- Accéléromètre

### **Planification de trajectoire**

- Différence avec la planification de chemin (si existe ?)
- Définition de la trajectoire optimale
- Description mathématique dans notre cas
- Manière de le résoudre, temps nécessaire, possibilités

### **Évitement d'obstacles**

#### **Essais**

- Diagramme de Voronoi
- DFS (Depth-First Search)

#### **Méthode par subdivision successive**

- Isotropic waves generator to validate or reject the method
- Cellular automata

### **Création d'une carte de caractéristique sur un maillage cartésien uniforme**

### **Planification dynamique de trajectoire**

### **Contrôle du robot**

- Cinématique du robot utilisé
- Modèle intégré

## Temps de calcul

- Liste des optimisations
  - Subdivision de la carte
  - Détection d'obstacles par le Lidar
    - \* Fonction `move_on_line` (parallèle avec le computer graphics)

## Non encore implémenté

- Sauvegarde dynamique de l'exploration

## References

- [1] S. M. LaValle. Planning Algorithms. Cambridge University Press, Cambridge, U.K., 2006. Available at <http://planning.cs.uiuc.edu/>.
- [2] National Geographic. Greenland's secret caves: Inside the hidden world of ice, 2025. Consulté le 10 janvier 2025.
- [3] Huu tho Dang. Underwater robots for karst and marine exploration : A study of redundant AUVs. PhD thesis, 2021. Thèse de doctorat dirigée par Lapierre, Lionel SYAM - Systèmes Automatiques et Micro-Électroniques Montpellier 2021.
- [4] Patricia Kambesis. The importance of cave exploration to scientific research. Journal of Cave and Karst Studies, 69, 04 2007.
- [5] Latombe Jean-Claude. Robot Motion Planning. Springer US, Boston, MA, 1991.
- [6] Bhattacharyya Aneeta, Singla Ekta, and Dasgupta Bhaskar. Robot path planning using silhouette method. In 13th National Conference on Mechanisms and Machines, pages 12–13, January 2008.
- [7] Lozano-Pérez Tomás and Wesley Michael A. An algorithm for planning collision-free paths among polyhedral obstacles. Communications of the ACM, 1979.
- [8] Garrido Santiago, Moreno Luis, Abderrahim Mohamed, and Martin Fernando. Path planning for mobile robot navigation using voronoi diagram and fast marching. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2376–2381, 2006.
- [9] Zhu David and Latombe Jean-Claude. New heuristic algorithms for efficient hierarchical path planning. IEEE Transactions on Robotics and Automation, 7:9–20, 1991.
- [10] Kedem K. and Sharir M. An efficient motion-planning algorithm for a convex polygonal object in two-dimensional polygonal space. Discrete and Computational Geometry, 5(1):43–76, 1990.
- [11] Avnaim Francis, Boissonnat Jean-Daniel, and Faverjon Bernard. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. In Proceedings of the 1988 IEEE International Conference on Robotics and Automation, volume 3, pages 1656–1661, 1988.
- [12] Koren Y. and Borenstein J. Potential field methods and their inherent limitations for mobile robot navigation. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, volume 2, pages 1398–1404, 1991.
- [13] Nancy M. Amato and Ying Wu. A randomized roadmap method for path and manipulation planning. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 113–120. IEEE, 1996.
- [14] David Hsu, Robert Kindel, Jean-Claude Latombe, and Steven Rock. Randomized kinodynamic motion planning with moving obstacles. The International Journal of Robotics Research, 21(3):233–255, 2002.
- [15] Christian Nissoux, Thierry Siméon, and Jean-Paul Laumond. Visibility based probabilistic roadmaps. Advanced Robotics, 13(2):223–244, 1999.
- [16] Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti, and Renato Vidoni. Path planning and trajectory planning algorithms: A general overview. Mechanisms and Machine Science, 29:3–27, 03 2015.