

Master Systèmes Intelligents et Robotique
2017/2018

Rapport de stage :
*Planification de trajectoire en présence
d'obstacles fortement dynamiques*

Poncelet Renaud

4 août 2018

Résumé

Le sujet s'inscrit dans le domaine de la planification de mouvement et l'évitement d'obstacles pour les véhicules autonomes. Nous nous intéressons ici au calcul du mouvement d'un robot en présence d'obstacles mobiles très dynamiques, c'est-à-dire que leur vitesse est de l'ordre de la vitesse du robot ou plus rapide. Il est proposé d'utiliser une nouvelle représentation dynamique de l'environnement grâce à la notion de "reachable interaction sets" (RIS). En particulier, lors de ce stage, je me suis intéressé à étendre le modèle des RIS pour intégrer des contraintes de mouvements non-holonomes et à intégrer un modèle d'obstacles stochastiques.

Table des matières

1	Introduction	1
1.1	Présentation du stage	1
1.2	Présentation du problème	2
1.3	RIS : Reachable Interaction Sets	3
2	Travaux	6
2.1	Intégration de contraintes de mouvements pour les robots non-holonomes	6
2.2	Obstacles stochastiques	9
2.3	Approche RIS-Path	11
2.3.1	Adaptation de la méthode RIS aux obstacles stochastiques	11
2.3.2	Intégration de l'approche RIS dans une approche globale	13
2.3.3	Gestions des cas d'urgence	13
2.4	Description des autres approches du benchmark présenté en Section 3.6	16
2.4.1	APF-SR	16
2.4.2	SES	16
2.4.3	DRT	16
2.4.4	RIS-APF	17
2.4.5	static-APF	17
2.4.6	dynamic-APF	17
3	Validation expérimentale	18
3.1	Présentation des environnements	18
3.1.1	Environnement "Open"	19
3.1.2	Environnement "Lanes"	20
3.1.3	Environnement "Door"	20
3.1.4	Scénarios "random"	20
3.1.5	Scénarios "spread"	21
3.2	Prise en compte des contraintes de mouvement pour les robots non-holonomes	21
3.3	Optimisation des différentes approches	23
3.4	Influence de la prise en compte du comportement stochastique des obstacles dans l'approche RIS-path	24
3.5	Comparaison des deux versions de RIS-path : RIS-Path Projection et RIS-Path Horizon	25
3.6	Résultat du RIS-path stochastique	26

3.6.1	Mouvement holonome du robot	26
3.6.2	Mouvement non-holonome du robot	29
	Conclusion	31
	Bibliographie	33

Chapitre 1

Introduction

1.1 Présentation du stage

Le stage a été effectué à l’Inria Paris, l’Institut national de recherche en informatique et automatique, au sein de l’équipe RITS pour Robotics & Intelligent Transportation Systems sous la direction d’Anne Verroust-Blondet et Pierre de Beauvais doctorant au sein de l’équipe RITS.

L’équipe est centrée sur les domaines de la robotique et des véhicules autonomes. Les domaines concernés sont donc : le traitement des images et des signaux, la fusion de données multi-capteurs, la planification de trajectoires, de manœuvres, et de tâches de contrôle-commande du véhicule, les outils de programmation temps réel distribués, les télécommunications, la modélisation, le contrôle et l’optimisation des systèmes de transport.

Les objectifs de ces recherches est l’amélioration du transport routier en terme de sécurité, d’efficacité, de confort et de minimisation des nuisances. L’approche technique est centrée sur les aides à la conduite, pouvant aller jusqu’à une automatisation totale.

L’équipe dispose d’une flotte de véhicules instrumentés avec différents types de capteurs (caméra, LIDAR, etc.) et des moyens de calculs et de simulation. Un système expérimental basé sur des véhicules urbains totalement automatisés est installé sur le site de Rocquencourt.

L’équipe participe activement au développement des "cybercars", véhicules urbains du futur avec conduite totalement automatisée. Des collaborations ont été établies avec divers acteurs industriels du secteur comme Renault, Robosoft, Valeo, AKKA Technologies et Yamaha. Des collaborations suivies sont en cours avec divers laboratoires étrangers : UC Berkeley, Jiao Tong University (Shanghai), Université Technique de Nayang (Singapour), NAIST (Japon), University de Tongji (Shanghai), Technical University of Sofia (Bulgarie), Seoul National University (Corée), ...

1.2 Présentation du problème

La planification de mouvement dans un environnement dynamique est d'une grande importance dans de nombreuses applications robotiques. Dans le contexte d'un robot mobile autonome, il faut calculer une trajectoire sans collision du début à la fin parmi les obstacles mobiles et statiques. (Voir figure 1.1). Ces problématiques sont abordées dans l'ouvrage de Minguez et al. [22].

Certaines approches calculent une trajectoire en présence d'obstacles fixes. Dans [19], le chemin est trouvé à l'aide de la méthode des "rapidly exploring random tree" (RRT) qui consiste à construire un arbre progressivement en ajoutant comme nœuds des points au hasard dans l'environnement et à les intégrer en les reliant au nœud le plus proche sans que le segment les reliant ne rencontre un obstacle, puis en trouvant le chemin le plus court dans le graphe ainsi obtenu. Une méthode construisant un graphe de visibilité est présentée dans [14] : un graphe formé des arêtes qui relient les sommets des obstacles polygonaux et qui n'intersectent pas les obstacles est calculé. La trajectoire construite correspond au plus court chemin dans ce graphe.

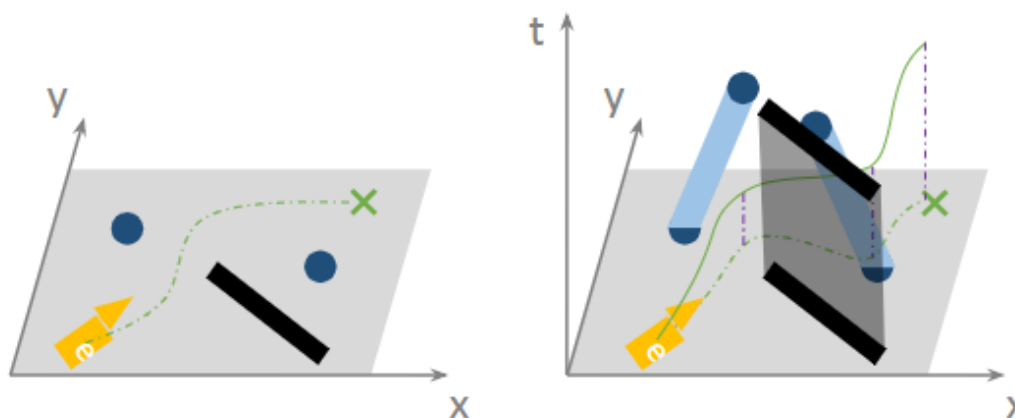


FIGURE 1.1 – Schéma représentant une planification de mouvement statique à gauche et dynamique à droite

De nombreuses méthodes ont été proposées pour calculer le mouvement d'un robot en présence d'obstacles mobiles.

Certaines approches utilisent l'espace des configurations, qui représente les transformations possibles pouvant être appliquées au robot se déplaçant parmi les obstacles. Cet espace des configurations peut être étendu à un espace de configuration spatio-temporel comme dans [8, 12] en considérant le mouvement des obstacles dynamiques. Dans [8], l'espace de configuration spatio-temporel est approximé par une liste de tranches d'espace de configurations et le robot est censé se déplacer à une vitesse supérieure à celle de n'importe quel obstacle. Dans [26] la vitesse maximale des obstacles en mouvement est connue et leur mouvement est représenté par des disques qui grandissent avec le temps. Leur but est de calculer une trajectoire entre ces disques en expansion.

D'autres méthodes résolvent le problème dans l'espace des vitesses, telles que l'approche par fenêtre dynamique (dynamic window) [11], par cône de collision (col-

lision cone) [2] et les obstacles de vitesse ou "velocity obstacles" (VO) [10],[25],[27], entre autres. Notons que dans le cas des VO, le robot et les obstacles sont supposés avoir la même limite supérieure de vitesse.

Les méthodes de champs de potentiel (potential field) [18], qui consistent à définir une fonction répulsive agissant sur le robot en fonction de l'environnement, ont été étendues aux obstacles dynamiques dans [17],[13] et [15]. Dans [17], la vitesse de l'obstacle est intégrée dans la définition de la fonction répulsive potentielle. Dans [13], les fonctions répulsives potentielles tiennent compte à la fois de la position relative et de la vitesse relative du robot par rapport à la cible et aux obstacles. Dans [15], les fonctions potentielles ne sont pas modifiées mais une nouvelle formulation de la vitesse planifiée du robot est introduite : elle est déterminée par les vitesses relatives et les positions relatives entre le robot, les obstacles et la cible. La combinaison d'un moteur à logique floue (fuzzy logic) avec un planificateur de trajet utilisant les champs de potentiel électrostatiques (electrostatic potential fields) est proposée dans [24].

Pour les environnements fortement dynamiques et dont le mouvement des obstacles comporte des incertitudes, une autre approche, intégrant des positions atteignables stochastiques ou "stochastic reachable sets" (SR sets), c'est à dire la répartition de la probabilité de présence d'obstacles au mouvement stochastique dans un futur proche, avec des champs de potentiel artificiel ou "artificial potential field" (APF) a été proposée dans [3] pour des obstacles au mouvement stochastique se déplaçant soit sur des segments de ligne ou sur des arcs de cercles et étendue dans [21] pour des obstacles qui peuvent passer de l'un à l'autre type de trajectoire. Dans [17],[13] et [15], les méthodes basées sur les "artificial potential field" ne considèrent que les positions instantanées des obstacles et non les positions futures pour calculer les champs potentiels, alors que dans [3] une prédiction du mouvement des obstacles, la distribution future des obstacles stochastiques en mouvement, est décrite par des SR sets qui sont utilisés pour générer des champs potentiels. Dans [4], [6] et [23], les méthodes sont basées sur les RRT [16] et utilisent les SR sets pour tester la probabilité de collision et s'assurer qu'elle est inférieure à un seuil avant chaque ajout d'un nœud dans l'arbre.

Dans le cadre de ce stage, les obstacles mobiles sont fortement dynamiques, c'est-à-dire que leur vitesse est de l'ordre de la vitesse du robot ou plus rapide. Le calcul du mouvement du robot s'appuie sur le calcul de zones du plan, les RIS, "Reachable Interaction Sets" qui sont présentées dans la suite.

1.3 RIS : Reachable Interaction Sets

Le problème d'un robot se déplaçant dans un environnement dynamique est NP-complet [1]. On ne peut donc pas trouver le chemin optimal dans \mathbb{R}^3 dans le cas général ($\mathbb{R}^2 \times \mathbb{R}$: deux dimensions d'espaces et une dimension de temps). Il est proposé d'utiliser une nouvelle représentation dynamique de l'environnement pour y intégrer des prédictions sur la position des obstacles dans un futur proche. Pour cela on utilise une représentation spatio-temporelle. La méthode consiste à détecter les zones de collisions possibles dans un futur proche dans $\mathbb{R}^2 * \mathbb{R}$. Pour cela il faut trouver les positions atteignables : "reachable sets" (RS) dans un futur proche par le

robot, ainsi que les positions futures des obstacles dans cet espace spatio-temporel. On utilise les modèles définis dans [9]. Les intersections entre ces différentes positions atteignables représentent les interactions possibles entre le robot et les obstacles dans un futur proche. Les surfaces obtenues dans $\mathbb{R}^2 * \mathbb{R}$ sont projetées sur le plan associé à l'instant présent ($t = 0$). On se ramène donc à un calcul de trajectoire dans un espace de dimension \mathbb{R}^2 . Or, il est possible de trouver le chemin optimal dans \mathbb{R}^2 [20].

Ces projections, utilisées comme obstacle, ont été nommées "reachable interaction sets" (RIS) et correspondent à des régions du plan. La notion de RIS a été introduite dans [7] pour représenter les zones d'interaction robot-obstacles à un horizon de temps donné. Par exemple, Figure 1.2 ① et ② un robot assimilé à un point se déplaçant à vitesse constante est représenté par un cône (en bleu) dans l'espace spatio-temporel. ③ représente le mouvement du robot avec contrainte non-holonomique. Le mouvement d'un obstacle (en orange) se déplaçant en ligne droite est représenté par un cylindre et le mouvement d'un obstacle qui tourne est une hélice. Les RIS (en rouge) correspondent à la projection des intersections (en vert) du cône (en bleu) et du cylindre (en orange).

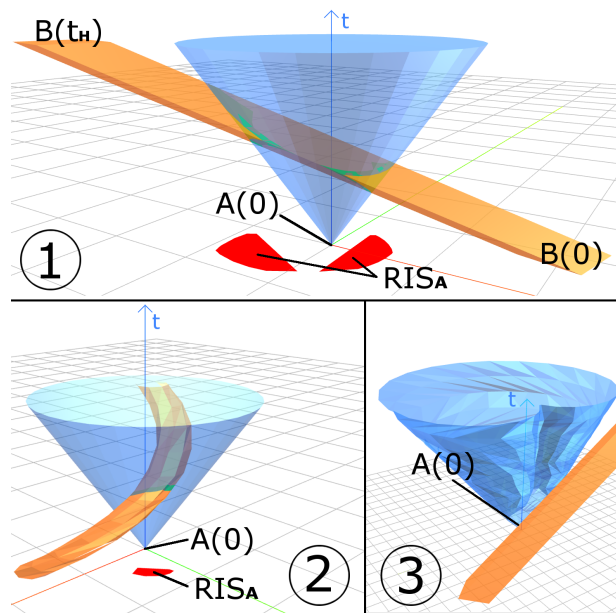


FIGURE 1.2 – Reachable sets dans l'espace *plan × temps* et reachable interactions sets projetés à l'instant $t = 0$

Ces RIS, intégrés dans des méthodes de planification de trajectoire s'avèrent très efficaces en présence d'obstacles fortement dynamiques : on remplace les obstacles dynamiques par les régions planaires correspondant aux RIS lors du calcul de la trajectoire à chaque instant. Ils améliorent aussi les méthodes réactive APF en remplaçant les obstacles dynamiques par les RIS pour le calcul du champ potentiel artificiel.

Dans le cadre de mon stage de master j'ai étendu le modèle des RIS pour intégrer des contraintes de mouvements non-holonomes pour le robot (Section 2.1). Ensuite, j'ai considéré des modèles d'obstacles au mouvement stochastique (Section 2.2) et

finalement, j'ai intégré mes apports à une méthode de calcul de trajectoire en présence d'obstacles statiques utilisant les RIS comme obstacles : RIS-Path. La méthode RIS-path a été modélisée et développée par Pierre de Beaucorps : doctorant dans l'équipe RITS. (Section 2.3). Tous ces développements ont été testés et comparés avec des approches existantes sur un ensemble de scénarios (Section 3.6).

Chapitre 2

Travaux

2.1 Intégration de contraintes de mouvements pour les robots non-holonomes

Pour un robot holonome se déplaçant à vitesse constante, les positions atteignables par le robot ou "reachable sets" (RS) sont sur un cône de rayon Vt où V est la vitesse du robot et t le temps (Voir Figure 1.2, ① et ②). Pour un robot non-holonome, cette surface peut être définie pour des modèles simples de robots (cf. [9]). Le robot de Dubins a été choisi pour sa simplicité et permet de bien décrire les contraintes de déplacement d'un robot non holonome dans une première approche. Les RS sont maintenant définies par le système suivant : (Voir Figure 2.1)

$$\begin{cases} \dot{\theta} = \omega \\ \dot{x} = \omega \sin \theta \\ \dot{y} = \omega \cos \theta \end{cases} \quad (2.1)$$

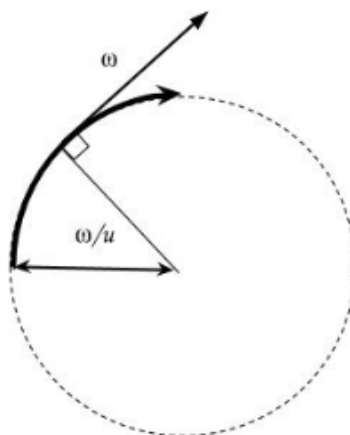


FIGURE 2.1 – robot de Dubins

où u et ω sont les commandes du système. En considérant que le robot se déplace avec une commande u constante respectant :

$$-u_{max} < u < u_{max} \quad (2.2)$$

u_{max} étant un paramètre dépendant du robot représentant l'angle de braquage maximal; ainsi qu'avec une commande w respectant :

$$w = \min(w_{init} + a_{max}t, w_{max}) \quad (2.3)$$

où a_{max} et w_{max} sont des paramètres du robot représentant respectivement l'accélération maximale et la vitesse maximale; et où w_{init} représente la vitesse initiale. On obtient les RS pour le couple de commande (u, w) . La vitesse initiale, l'accélération maximale et la vitesse maximale sont connues et on adopte la loi de comportement de vitesse (2.3) comme loi de comportement du robot et cela quel que soit l'approche utilisée pour la planification. On supprime alors la commande w . On calcule les RS pour un ensemble de valeurs u respectant (2.2). Un exemple de surface est présentée en Figure 1.2, ③

En pratique, on simplifie encore ce modèle pour pouvoir calculer les RS au préalable. On discrétise les valeurs de w_{init} possibles entre 0 et w_{max} , pour chaque valeur de w_{init} on calcule les positions $x(t)$ et $y(t)$ pour toutes les valeurs de u et pour un temps discret. Ce qui transforme le système (2.1) :

$$\begin{cases} \theta(t + \Delta) = \theta(t) + u \times \Delta \\ x(t + \Delta) = x(t) + \omega \sin(\theta(t + \Delta)) \times \Delta \\ y(t + \Delta) = y(t) + \omega \cos(\theta(t + \Delta)) \times \Delta \end{cases} \quad (2.4)$$

Pour simplifier les notations on prendra le pas de temps $\Delta = 1$. On a donc :

$$\begin{cases} \theta(t + 1) = \theta(t) + u \\ x(t + 1) = x(t) + \omega \sin(\theta(t + 1)) \\ y(t + 1) = y(t) + \omega \cos(\theta(t + 1)) \end{cases} \quad (2.5)$$

On a donc pour chaque couche temporelle un polygone représentant les positions atteignables à cet instant.

Remarque 2.1 *Pour n'importe quelle valeur de u non-nulle il existe un temps au bout duquel le robot revient dans son état initial (position et orientation). Ce comportement n'est pas désiré pour le robot. On décide donc de supprimer l'ensemble des points tel que :*

$$x(t + 1) * x(t) < 0 \quad (2.6)$$

Et de ne pas calculer les positions futures des points ainsi supprimés.

On obtient un ensemble de points représentant la surface des RS. La surface est construite en reliant les lignes polygonales successives correspondant aux instants $t, t + 1, \dots, t + t_H$. t_H étant l'horizon temporel de prédiction avec des facettes triangulaires.

Pour obtenir un maillage suffisamment précis, la discrétisation en u est de 65 valeurs. Ceci est raisonnable pour les lignes polygonales correspondant aux instants

proches de l'horizon temporel mais induit une discrétisation trop fine des lignes polygonales correspondant aux instants plus proches de t . En effet, les lignes polygonales ont tendance à s'étendre spatialement et à s'allonger. De plus, elles comportent de moins en moins de points au cours du temps (Voir la Remarque 2.1). Pour conserver un écart raisonnable entre deux points successifs d'une ligne polygonale, on supprime des points des lignes de faible longueur (correspondant aux instants proches de l'instant t). Pour conserver la forme des RS ces points sont supprimés de manière régulière le long des lignes polygonales en fonction du ratio entre le nombre de points souhaité et le nombre de points initial de chaque ligne.

Pour construire la surface correspondant au robot de Dubins, il nous faut relier les lignes polygonales successives par des faces régulières. Or, le fait de supprimer des points des lignes polygonales, nous empêche d'effectuer une mise en correspondance systématique en considérant le paramétrage u de chaque ligne polygonale. On sait cependant que les points sont organisés par u croissant.

Il existe différents cas de figure suivant les deux lignes que l'on souhaite relier :

- Les lignes polygonales sont ouvertes ou fermées (on est avant ou après le premier demi-tour complet).
- Les deux lignes polygonales ont le même nombre de points ou non.

Dans le cas où les lignes considérées ont le même nombre de points, on décide de relier les points comme le montre la Figure 2.2. Si en revanche les couches n'ont pas le même nombre de points on va procéder de la même manière en rajoutant régulièrement un triangle supplémentaire comme le montre la Figure 2.3. Si la surface est fermée on relie les points extrêmes entre eux. Si la surface est ouverte on ne les relie pas.

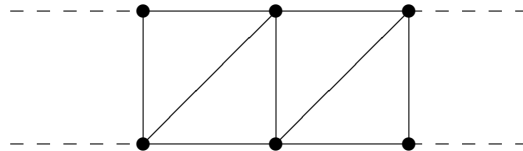


FIGURE 2.2 – Figure illustrant le maillage réalisé pour construire la surface des RS entre deux couches ayant les même nombre de points

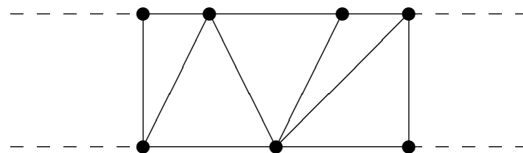


FIGURE 2.3 – Figure illustrant le maillage réalisé pour construire la surface des RS entre deux couches n'ayant pas les même nombre de points

Pour introduire un triangle supplémentaire, on procède de la même manière que pour supprimer des points. On ajoute régulièrement des triangles en fonction du ratio entre les nombres de points des deux couches polygonales.

Une fois la surface construite, il nous faut calculer l'intersection de la surface avec les volumes correspondant aux obstacles comme dans la Figure 1.2.

Nous avons utilisé la bibliothèque logicielle de calcul géométrique pyMesh [28] qui ne prend en entrée que des volumes. Pour ce faire, nous avons construit un volume de faible épaisseur autour de la surface des RIS calculée précédemment. Cela donne le maillage ci-dessous :

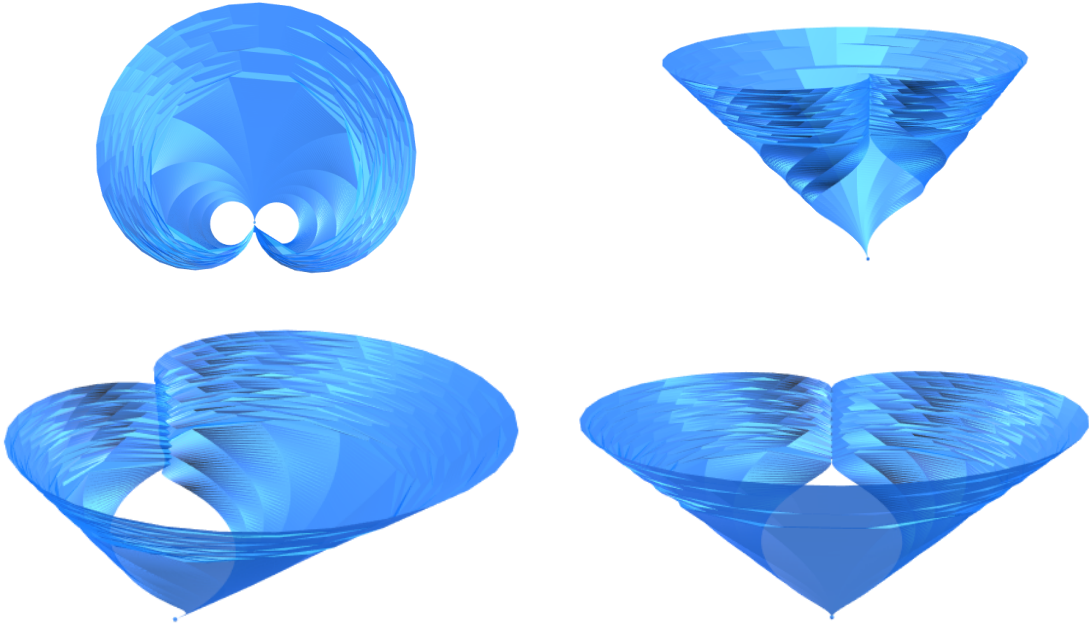


FIGURE 2.4 – Maillage obtenu pour la surface de Dubins.

Une visualisation 3D est disponible en ligne <http://deboc.fr/ris/>. L'approche RIS utilisant les RS basés sur le modèle de Dubins a été validée dans [7] sur un ensemble de tests similaires à ceux présentés dans [5]. Les résultats seront détaillés en Section 3.2.

2.2 Obstacles stochastiques

Quand le mouvement des obstacles mobiles n'est pas connu avec certitude, celui-ci peut être modélisé par un modèle stochastique. Le modèle des RIS peut être étendu à ce cas. Dans le but de comparer les RIS avec les méthodes existantes comme dans [5], nous avons implémenté les approches suivantes :

- APF-statique [18], qui construit un champ potentiel artificiel à partir des zones occupées par des obstacles à chaque instant.
- APF-dynamique [17], qui construit elle aussi un champ potentiel artificiel à partir des zones occupées par des obstacles mais en tenant compte de leur vitesse.
- APF-SR [3], qui construit un champ potentiel artificiel à partir d'un champ scalaire. Ce champ scalaire est construit en additionnant les probabilités de présence des obstacles de l'instant t à l'instant $t + t_H$.

- SES [4], qui construit un RRT en testant à chaque ajout d'un point dans l'arbre sa probabilité d'être dans un obstacle en utilisant un seuil.
- DRT [6], qui procède de la même manière que la méthode SES en modifiant la loi de sélection des points du RRT pour autoriser un risque plus grand quand les instants considérés sont éloignés de l'instant présent.

Pour chaque obstacle une prédiction des positions futures basée sur la méthode de Monte-Carlo utilisée dans [4] est introduite. On pose un modèle d'obstacles stochastiques comme suit :

- L'obstacle se déplace sur une trajectoire définie (ici les trajectoires sont des lignes droites)
- A chaque itération on tire une vitesse aléatoirement parmi k vitesses possibles. Cela correspond au système suivant :

$$\begin{cases} V_{obstacle}(i, t) &= random(v_1, \dots, v_k) \\ X_{obstacle}(i, t) &= X_{obstacle}(i, t - \Delta t) + V_{obstacle}(i, t) \times \Delta t \end{cases} \quad (2.7)$$

Où $V_{obstacle}(i, t)$ est la vitesse de l'obstacle à l'instant t lors de la $i^{\text{ème}}$ itération de Monte-Carlo, (v_1, \dots, v_k) sont les k vitesses possibles de l'obstacle, $X_{obstacle}(i, t)$ est la position de l'obstacle à l'instant t lors de la $i^{\text{ème}}$ itération de Monte-Carlo et Δt est le pas de temps.

On peut calculer pour chaque itération de Monte-Carlo :

$$P(x, y, i, t) = \begin{cases} 1 & \text{Si } (x, y) \text{ est occupé par un obstacle à la} \\ & i^{\text{ème}} \text{ itération de Monte-Carlo à l'instant } t \\ 0 & \text{Sinon} \end{cases} \quad (2.8)$$

En prenant la moyenne de toutes les itérations on obtient :

$$P_{occupation}(x, y, t) = \frac{1}{N} \sum_{i=1}^N P(x, y, i, t) \quad (2.9)$$

On obtient une probabilité de présence pour le type de modèle d'obstacles stochastiques présenté en (2.7). Cette probabilité peut être calculée au préalable pour être stockée puis utilisée lors du calcul de trajectoire. Voici sous forme d'images les répartition de probabilités obtenues :

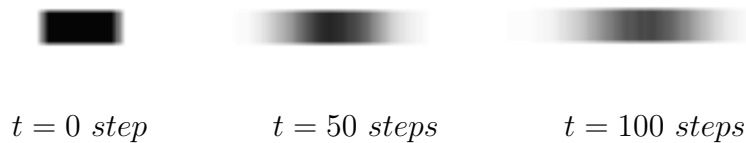


FIGURE 2.5 – Images représentant la probabilité de présence d'un obstacle à trois instants différents dans le cas d'un mouvement le long d'une ligne droite. Plus un pixel tend vers le noir plus sa probabilité d'être occupé par l'obstacle est élevée.

Pour utiliser cette répartition de probabilités les trois méthodes SES, DRT et APF-SR calculent à chaque instant la position (ou future position) relative du robot par rapport aux obstacles. Cette position relative permet, grâce à la répartition calculée au préalable, de déduire la probabilité que cette position a d'être occupée par les obstacles. SES et DRT utilisent cette information pour construire un "rapidly exploring random tree" (RRT) alors que APF-SR l'utilise pour calculer un "artificial potential field" (APF).

Nous avons obtenu des performances similaires à celles présentées dans [5]. Voir en section 3.3.

2.3 Approche RIS-Path

La méthode basée sur les RIS a été adaptée aux obstacles stochastiques pour la comparer aux méthodes existantes au benchmark présenté dans [5]. De plus, d'autres modifications ont été adoptées pour améliorer la méthode RIS.

2.3.1 Adaptation de la méthode RIS aux obstacles stochastiques

Pour intégrer avec des obstacles stochastiques respectant (2.7) dans la méthode RIS; nous avons décidé de seuiller la probabilité de présence pour être capable de définir un intérieur et un extérieur aux obstacles. On utilise un seuil dynamique inspiré de celui utilisé dans DRT [6]. Le seuil est défini comme ci-dessous :

$$P_{accept}(t) = \begin{cases} P_{min} & \text{Si } t \leq \tau_1 \\ \alpha(t) & \text{Si } \tau_2 > t > \tau_1 \\ P_{max} & \text{Si } t \geq \tau_2 \end{cases} \quad (2.10)$$

avec :

$$\alpha(t) = \frac{P_{max} - P_{min}}{\tau_2 - \tau_1} \times (t - \tau_1) + P_{min} \quad (2.11)$$

cela donne la courbe ci-dessous :

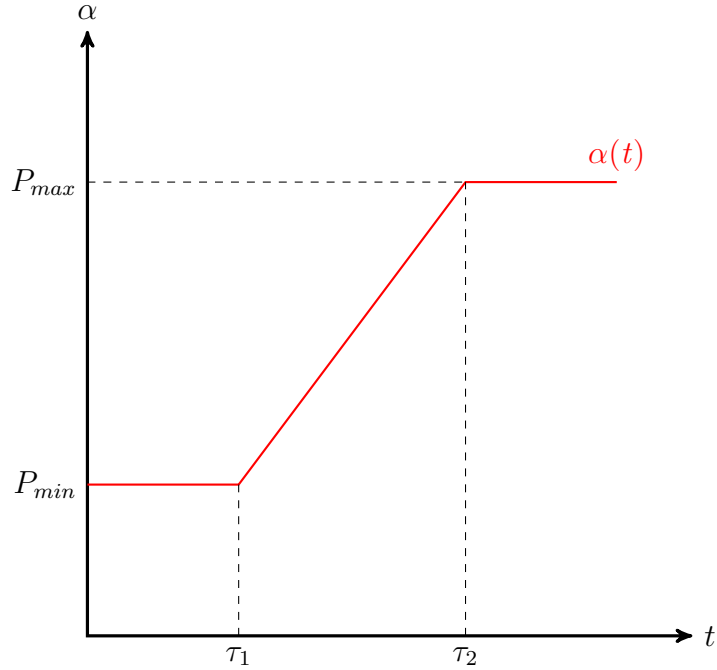


FIGURE 2.6 – Seuil dynamique permettant de définir à quelle probabilité se trouve la limite de l’obstacle

Où $P_{accept}(t)$ est le seuil dynamique défini comme une fonction du temps ayant pour paramètres : P_{min} le seuil minimal, P_{max} le seuil maximal, τ_1 le temps durant lequel le seuil est au minimum et $\tau_2 - \tau_1$ le temps de montée du seuil (voir Figure 2.6).

On peut définir à chaque $P_{occupation}(x, y, t)$ les pixels qui sont à l’intérieur de l’obstacle et ceux qui sont à l’extérieur. On peut donc construire en reliant les polygones obtenus l’équivalent stochastique des RS. De la même manière que pour les autres méthodes, les RS peuvent être calculés au préalable. En calculant l’intersection de ceux-ci avec les RS du robot, on obtient une version stochastique des RIS. L’apport de cette adaptation de la méthode aux obstacles stochastiques est présenté en Section 3.4.

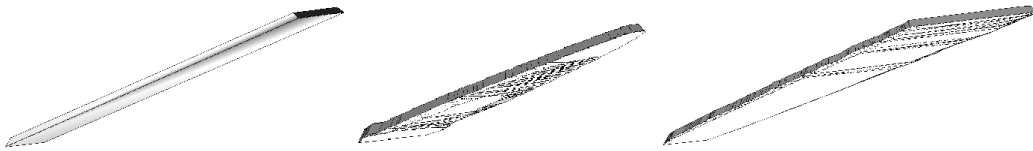


FIGURE 2.7 – Maillage d’un obstacle à vitesse constante à gauche et maillage d’obstacles stochastiques à droite dans l’espace spatio-temporel. *Remarque : La forme de l’obstacle stochastique dépend du seuil défini en (2.11). En effet, on a un seuil dynamique au centre et un seuil constant à droite*

2.3.2 Intégration de l'approche RIS dans une approche globale

En utilisant le graphe de visibilité [20], qui permet de calculer le chemin optimal parmi des obstacles statiques polygonaux et en remplaçant les obstacles par les RIS, on obtient de très bons résultats en présence d'obstacles très dynamiques. La méthode intitulée RIS-path a été introduite et développée par Pierre de Beaucorps. En particulier, les graphes de visibilité ont été étendus pour les robots non holonomes par Pierre de Beaucorps pour que le chemin obtenu respecte les contraintes de mouvement du robot.

2.3.3 Gestions des cas d'urgence

La méthode RIS-Path décrite plus haut peut mener à des cas que nous appelons cas d'urgence. Nous avons donc introduit des heuristiques pour résoudre ces cas particuliers :

- Lorsque le robot est dans un RIS le risque de collision est plus important qu'en dehors d'un RIS. Il faut donc en sortir au plus vite.
- Lorsque aucun chemin n'est trouvé, le robot se stoppe, or si le robot est à l'arrêt les RS calculés pour le robot ne sont plus valables car le robot ne respecte plus le comportement prévu par l'équation (2.3). Il faut donc recalculer les RS à l'arrêt.

Cas du robot dans un RIS

La méthode de planification de mouvement vise à créer un chemin qui ne passe pas par les RIS, mais il arrive que le robot se retrouve à l'intérieur d'un RIS. Cela est dû au fait que l'horizon du robot est limité, sur cet horizon donné on considère que le robot se déplace à vitesse constante et que certains obstacles se déplacent à grande vitesse ce qui rend ces approximations plus fragiles.

On calcule donc deux ensembles de RIS différents. D'abord, les RIS classiques qui possèdent une marge (Voir la Figure 2.8. Puis, les RIS "d'urgence" sans marge.



FIGURE 2.8 – Dessin représentant le RIS sans marge en bleu et la marge du RIS en marron. Le RIS sans marge est inclus dans le RIS avec marge.

Pour trouver le chemin le plus court pour sortir du RIS avec ou sans marge on cherche le point libre le plus proche du robot, c'est à dire le point P_0 situé sur le

bord du RIS le plus proche du centre du robot. Le segment de droite reliant le centre du robot et P_0 représente le chemin de sortie.

Cette méthode n'est valable que pour un robot holonome. Dans ce cas d'un robot non-holonome on trace un certain nombre de trajectoires (les expériences ont été réalisées avec 5 trajectoires) à braquage constant (u constant dans l'équation (2.1)) respectant les contraintes du robot non holonome (c'est à dire respectant l'équation (2.5)). La trajectoire sortant le plus rapidement du RIS est alors celle qui est choisie.

Plusieurs cas sont alors possibles :

Algorithme 1 : Construction du chemin

```

if robot dans un RIS avec marge then
  if robot dans un RIS sans marge then
    Trouve le chemin le plus court pour sortir du RIS sans marge
  else
    Trouve le chemin le plus court pour sortir du RIS avec marge
    if Le chemin pour sortir du RIS avec marge passe par un RIS sans marge
    then
       $PATH \leftarrow []$ 
    end if
  end if
   $PATH \leftarrow [Chemin\ de\ sortie\ du\ RIS]$ 
else
  Trouve le chemin le plus court pour atteindre le but sans passer par les RIS
   $PATH \leftarrow$ 
  [Chemin le plus court pour atteindre le but sans passer par les RIS]
end if

```

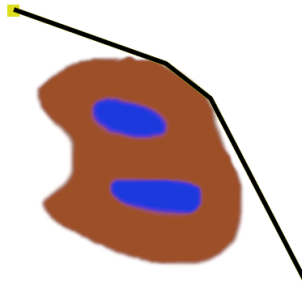


FIGURE 2.9 – Chemin (en noir) pour un robot (en jaune) en dehors d'un RIS avec marge.

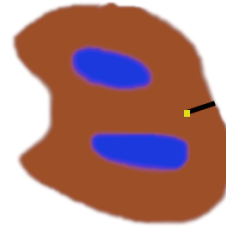


FIGURE 2.11 – Chemin (en noir) pour un robot (en jaune) dans la marge d'un RIS.

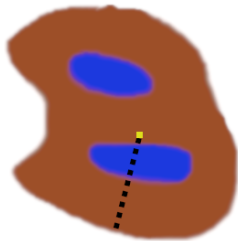


FIGURE 2.10 – Chemin (en noir) passant dans un RIS sans marge pour un robot (en jaune) dans la marge d'un RIS. Le chemin n'est pas valide



FIGURE 2.12 – Chemin (en noir) pour un robot (en jaune) dans un RIS sans marge.

Cas du robot à l'arrêt

Malgré tout, si le chemin est vide, soit parce qu'aucun chemin n'a été trouvé, soit parce que le chemin pour sortir du RIS avec marge passe par un RIS sans marge (comme en Figure 2.10, alors le robot ne bouge pas. Cela peut correspondre au comportement voulu. Cependant, il faut s'assurer que la position d'arrêt est sûre car à l'arrêt les RS du robot ne correspondent pas à ceux qui ont été calculés. En effet, quand le robot est à l'arrêt, les RS sont sur une droite verticale passant par sa position actuelle dans l'espace-temps.

Pour vérifier que la position est sûre, on vérifie que cette droite n'intersecte pas les RS des obstacles. Si l'intersection est vide, la position est sûre. Le robot peut donc rester immobile jusqu'à l'instant suivant où un calcul de chemin sera effectué.

Dans le cas contraire la position n'est pas sûre. Il faut alors générer un chemin pour quitter cette position. Une heuristique est introduite. Nous avons identifié deux possibilités qui semblent satisfaisantes :

- La première est de lister les obstacles dangereux dans le cas où le robot resterait à l'arrêt. Pour tous ces obstacles, on projette l'ensemble de leurs RS sur le plan à $t = 0$ et on ajoute ces projections aux RIS. Sachant que la position n'est pas sûre, le robot se retrouve alors dans un RIS dont il va sortir au plus vite en respectant l'heuristique présentée plus haut.
- La deuxième solution est de réduire l'horizon de prédiction dans le but de

réduire les RIS jusqu'à trouver un chemin. La position actuelle n'étant pas sûre, le robot ne peut pas rester à l'arrêt. L'objectif de cette heuristique est de trouver un chemin en réduisant les exigences de planification au court terme.

Les deux approches ont leurs avantages et leurs défauts. La première permet de forcer le robot à sortir de la zone de danger sans réduire l'horizon de prédiction, mais le chemin choisi est moins bon que le chemin trouvé si on réduisait l'horizon. Cependant, une réduction trop forte de l'horizon rend le robot aveugle à l'environnement. Les deux versions de RIS-path ont été réalisées puis testées sur un certain nombre de scénarios dans le cas d'un robot holonomique comme dans le cas d'un robot non-holonyme. Cette expérience est présentée en 3.5.

2.4 Description des autres approches du benchmark présenté en Section 3.6

Les méthodes testées dans les évaluations expérimentales sont rapidement présentées ci-dessous.

2.4.1 APF-SR

La méthode APF-SR utilise la répartition de probabilité de présence des obstacles présentée en section 2.2. En additionnant les probabilités obtenues de l'instant t à t_H elle obtient un champ scalaire. C'est à partir de ce champ scalaire que la méthode calcule un champ potentiel artificiel.

2.4.2 SES

La méthode SES utilise la répartition de probabilité de présence des obstacles présentée en section 2.2. La méthode construit un RRT. Pour ce faire elle sélectionne un point de l'environnement au hasard qu'elle relie au nœud le plus proche dans l'arbre (initialement la position du robot est ajoutée dans l'arbre). Sur ce segment, elle calcule le point qui peut être atteint en une itération depuis la position du nœud. Enfin, elle teste que la probabilité de présence d'un obstacle en ce point à l'instant $t + 1$ est inférieure à un seuil. Si tel est le cas le point est ajouté à l'arbre. Sinon, il n'est pas ajouté. Ensuite, la méthode recommence cette procédure jusqu'à avoir ajouté un nombre de points maximal autorisé. Une fois cet arbre construit, à chaque point est associé un coût calculé en fonction de la distance du point au but et de la probabilité de présence d'un obstacle en ce point. Le point ayant le coût le plus bas est sélectionné par la méthode. Celui-ci ainsi que tous ses parents constituent un chemin.

2.4.3 DRT

La méthode DRT est une extension de SES. Elle remplace le seuil de probabilité statique par un seuil dynamique. L'idée est d'augmenter ce seuil avec le temps.

2.4.4 RIS-APF

La méthode RIS-APF utilise les RIS comme obstacles pour calculer un champ de potentiel artificiel.

2.4.5 static-APF

La méthode statique-APF utilise la position des obstacles à chaque instant pour calculer un champ de potentiel artificiel.

2.4.6 dynamic-APF

La méthode dynamic-APF est une extension de static-APF qui tient compte de la vitesse des obstacles pour calculer un champ de potentiel artificiel.

Chapitre 3

Validation expérimentale

Dans tous les scénarios présentés le robot part d'un point donné et doit rejoindre un point cible sans avoir de collision avec un obstacle fixe ou mobile. Si le robot rejoint le point cible sans collision, le scénario se termine avec succès. Sinon on considère que c'est un scénario en échec. On appelle "taux de succès" ou "résultat" le ratio du nombre de succès divisé par le nombre de scénarios. Il est exprimé en pourcentage.

3.1 Présentation des environnements

Trois types d'environnements ont été développés pour tester les performances des différentes méthodes. Les environnements possèdent des obstacles statiques et dynamiques. Les obstacles statiques permettent de mettre en évidence la capacité des différentes méthodes à fonctionner dans des conditions particulières. Ici, nous travaillerons avec différents environnements :

- "Open", qui n'a pas d'obstacle statique et où les obstacles mobiles n'ont pas de direction particulière.
- "Lanes", où les obstacles mobiles suivent une voie et où il n'y a pas d'obstacle statique.
- "Door", où le robot doit franchir une porte et où les obstacles dynamiques n'ont pas de direction particulière.

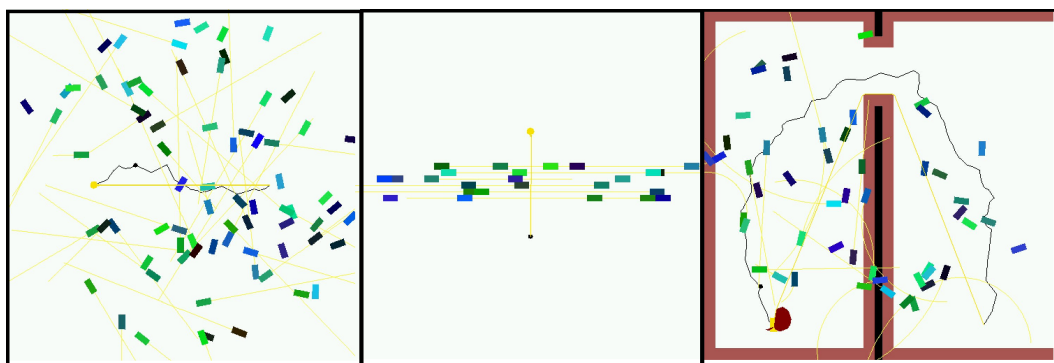


FIGURE 3.1 – Image montrant les différents environnements : "Open", "Lanes", "Door" (de gauche à droite)

On différencie l'environnement du scénario. Les environnements détaillés ci-dessus sont déclinés en différents scénarios qui dépendent du comportement des obstacles dynamiques. Nous travaillerons avec différents scénarios :

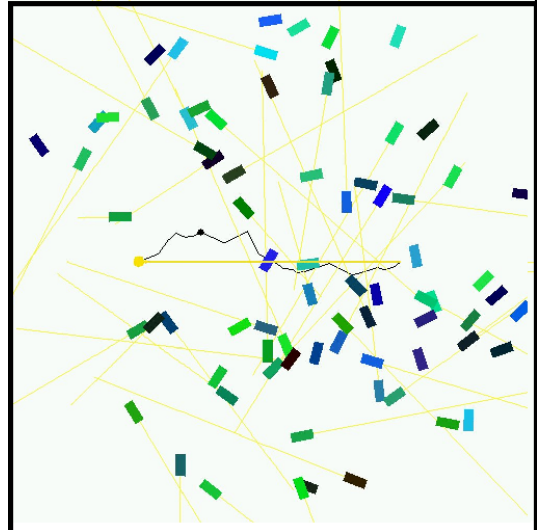
- "random" pour aléatoire, c'est un scénario où la vitesse des obstacles est choisie au hasard à chaque pas de temps.
- "spread" pour séparé, qui est aussi un type de scénario avec des obstacles stochastiques mais dont les vitesses possibles peuvent être à la fois très lentes et très rapides

Pour chaque type de scénario le nombre d'obstacles peut varier. Un scénario est donc défini par son type et par son nombre d'obstacles. On notera donc "spread-75" un scénario de type "spread" dans lequel évoluent 75 obstacles dynamiques. Les scénarios de type "random" sont des scénarios inspirés de [5]. Cependant les valeurs introduites dans [5] ont été modifiées en respectant les échelles pour uniformiser les scénarios. Les approches présentées dans [5] obtiennent des résultats similaires dans les scénarios que nous avons adaptés. Les autres scénarios sont des variantes introduites pour tester ces approches de façon plus approfondie, que ce soit en terme de densité d'obstacles, de valeurs de vitesse, ou de présence d'obstacles statiques.

Je vais présenter ci-dessous les différents environnements puis les différents scénarios.

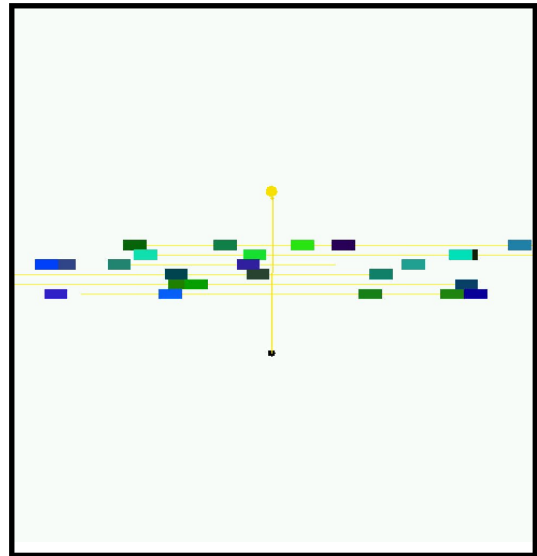
3.1.1 Environnement "Open"

- Largeur : 1000 *pixels*
- Longueur : 1000 *pixels*
- Position initiale : [250, 500]
- Destination finale : [750, 500]
- Vitesse maximale du robot : 1.34 *pixels/step*
- Type de robot : ponctuel



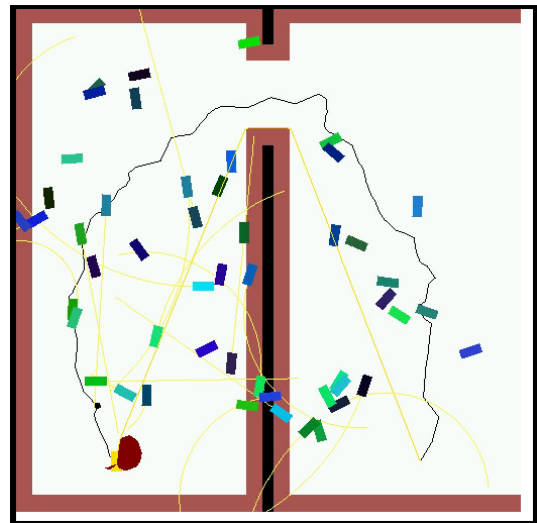
3.1.2 Environnement "Lanes"

- Largeur : 1000 *pixels*
- Longueur : 1000 *pixels*
- Position initiale : [500, 350]
- Destination finale : [500, 650]
- Vitesse maximale du robot : 1.34 *pixels/step*
- Type de robot : ponctuel



3.1.3 Environnement "Door"

- Largeur : 1000 *pixels*
- Longueur : 1000 *pixels*
- Position initiale : [200, 900]
- Destination finale : [800, 900]
- Vitesse maximale du robot : 1.34 *pixels/step*
- Type de robot : ponctuel
- Obstacles statiques : représentés en noir sur la figure de droite. Les zones marrons représentent l'obstacle statique avec une marge.



3.1.4 Scénarios "random"

- Forme des obstacles :
 - Type : rectangle
 - Longueur : 42 *pixels*
 - Largeur : 18 *pixels*
- Vitesses possibles des obstacles : [1.13, 1.69, 2.24, 2.81, 3.375] *pixels/step*
- Type de mouvement :
 - Vitesse : Norme de vitesse choisie à chaque période de ré-échantillonnage dans une liste échantillonnée
 - Orientation : Fixée au hasard lors de l'initialisation
 - Angle de braquage : Trajectoire rectiligne

- Période de ré-échantillonnage : 1 *step*
- Position initiale : Choisie au hasard à l'initialisation

3.1.5 Scénarios "spread"

- Forme des obstacles :
 - Type : rectangle
 - Longueur : 42 *pixels*
 - Largeur : 18 *pixels*
- Vitesses possibles des obstacles : [0.8, 0.8, 2.24, 4.0, 5.36, 5.36] *pixels/step*
- Type de mouvement :
 - Vitesse : Norme de vitesse choisie à chaque période de ré-échantillonnage dans une liste échantillonnée
 - Orientation : Fixée au hasard lors de l'initialisation
 - Angle de braquage : Trajectoire rectiligne
 - Période de ré-échantillonnage : 1 *step*
- Position initiale : Choisie au hasard à l'initialisation

3.2 Prise en compte des contraintes de mouvement pour les robots non-holonomes

Dans le but de connaître l'apport de la prise en compte des contraintes de mouvement dans les RS des robots non-holonomiques, nous comparons les résultats de la méthode RIS-path pour un robot non-holonomique dont les RS calculés au préalable tiennent compte des contraintes de mouvement du robot dans un cas et n'en tiennent pas compte dans l'autre. En pratique, la première méthode utilise la surface de Dubins comme RS alors que la deuxième utilise le cône généré pour les robots holonomes.

L'utilisation des RS non holonome avec des obstacles non stochastiques améliore légèrement le taux de succès comme le montre une figure réalisée dans un environnement "door" dans un scénario non présenté ici et réalisé par Pierre de Beaucois et résumé dans la Figure 3.2

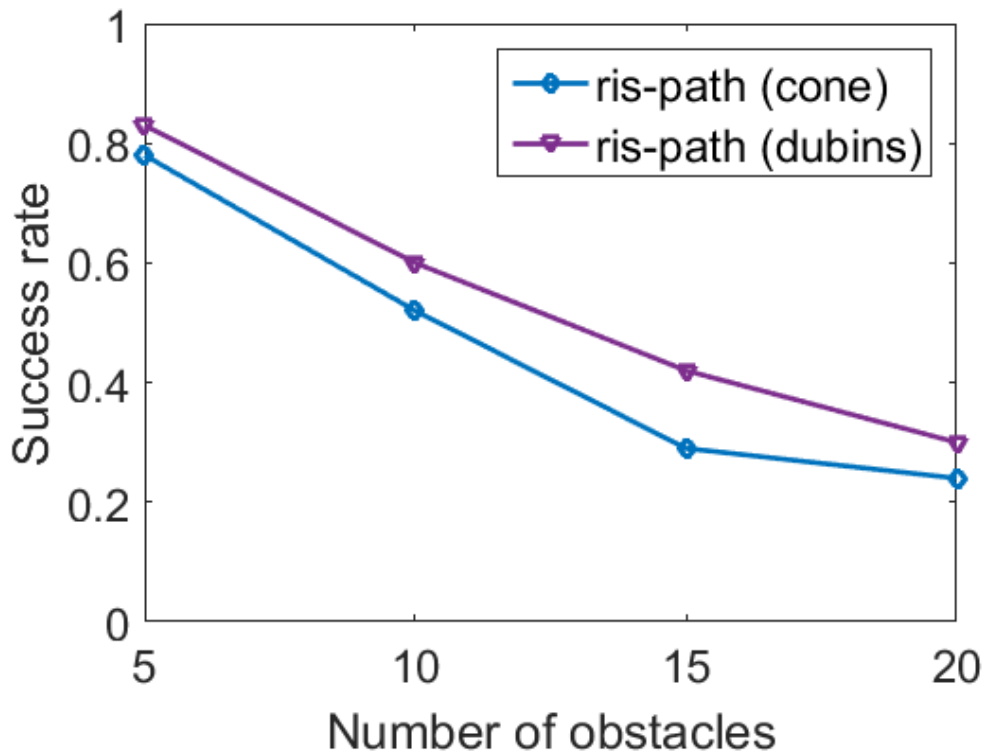


FIGURE 3.2 – Comparaison dans un environnement "door" de la méthode RIS-path avec comme RS un cône puis une surface de Dubins

Dans le cas d'obstacles stochastiques, on obtient dans "open" pour des scénarios de type "spread-75" les résultats suivants :

	cône	Dubins
résultats	0,84	0.54

Tableau 3.1 – Comparaison du taux de succès de RIS-path sur des scénarios de type "spread-75" dans l'environnement "open" pour deux robots non-holonomes : l'un avec des RS de robot holonome l'autre avec des RS tenant compte des contraintes non-holonomes. (Le taux de succès est calculé sur 110 scénarios pour chaque robot)

On voit que la prise en compte des contraintes non holonomes dans le calcul des RIS réduit très fortement le taux de succès. Ce résultat n'a pas pu être interprété faute de temps.

Une tentative d'amélioration a été de réduire l'horizon temporel de façon à ce que le robot n'ai pas le temps de faire un tour complet. Le taux de succès s'améliore et devient égal à 0.63%. Mais il reste encore inférieur aux résultats obtenus avec le cône. Dans le benchmark présenté en Section 3.6 les approches basées sur les RIS n'utilisent pas le calcul du RIS avec contrainte non-holonomique.

3.3 Optimisation des différentes approches

Dans le but de comparer les approches testées sur les RIS avec d'autres méthodes de l'état de l'art, les différentes méthodes ont été optimisées pour les scénarios utilisés pour la comparaison. Pour cela les différents paramètres des approches ont été échantillonnés et évalués séparément pour essayer de se rapprocher de leurs optimums. Pour l'approche DRT les paramètres optimisés sont :

- "switchTime" : le seuil de probabilité de présence au-dessus duquel un point du RRT est supprimé est dynamique dans DRT. Le paramètre "switchTime" est l'instant à partir duquel le seuil passe d'une valeur constante à une loi qui est fonction du temps.
- "nSafety" : qui est la longueur minimale d'un chemin en nombre de points pour pouvoir être valide.
- "nSafetyEmergency" : qui est la valeur que prend nSafety en cas d'urgence.

Voici un exemple des résultats obtenus pour l'approche DRT [6] dans l'environnement "open" sur des scénarios de type "spread-150" :

switchTime		nSafety		nSafetyEmergency	
valeur	résultat	valeur	résultat	valeur	résultat
5	0.55	3	0.47	10	0.6
10	0.64	5	0.56	25	0.56
20	0.65	10	0.65	40	0.65
30	0.65	15	0.68	50	0.66
40	0.59	20	0.65	75	0.58
75	0.61	35	0.78	100	0.53
		80	0.09		

Tableau 3.2 – Tableau comparatif des résultats moyens de la méthode DRT sur 110 scénarios en fonctions de différentes valeurs de ses paramètres (Afin d'optimiser chaque valeur indépendamment, les autres paramètres sont laissés à une valeur de référence ($switchTime = 20$, $nSafety = 10$, $nSafetyEmergency = 40$)). Les valeurs en rouge sont choisies pour le suite.

Ces résultats ont été obtenus en affinant les paramètres en fonction de la tendance des résultats précédents (par exemple pour le paramètre "switch time" le calcul a d'abord été réalisé pour les valeurs 5, 30, 75 puis des valeurs ont été ajoutées au fur et à mesure pour faire ressortir le maximum). Certains résultats ne dégagent pas de tendance claire dans ce cas la valeur du paramètre ayant obtenu le meilleur score a été choisie malgré le manque de finesse du choix des valeurs.

De même pour SES dans l'environnement "open" pour des scénarios "random-75" :

nSafety	
valeur	résultat
10	0.83
25	0.96
35	0.81
50	0.46

Tableau 3.3 – Tableau comparatif des résultats moyen de la méthode SES sur 110 scénarios en fonctions de différentes valeurs de son paramètre. La valeur en rouge est choisie pour la suite.

3.4 Influence de la prise en compte du comportement stochastique des obstacles dans l'approche RIS-path

Dans le but de montrer l'influence de la prise en compte du comportement stochastique des obstacles sur le taux de réussite de la méthode RIS-path, on compare RIS-path avec une version dégénérée de la méthode qui ne tient compte que de la vitesse moyenne des obstacles pour créer leur RS et non pas en faisant un tirage de Monte-Carlo pour obtenir une répartition de probabilité de présence comme présenté en Section 2.3.1. La comparaison a été effectuée sur l'environnement "open" pour des scénarios de type "spread". Les résultats sont résumés par la Figure 3.3 :

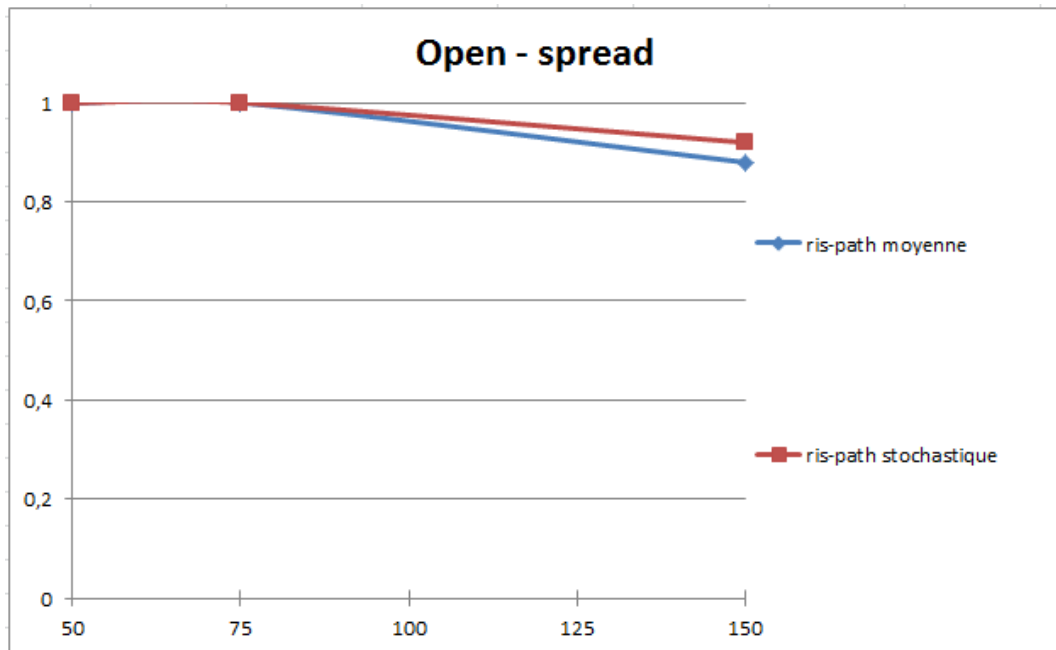


FIGURE 3.3 – Résultats des approches RIS-path en tenant compte ou non du caractère stochastique des obstacles sur l'environnement "Open" pour des scénarios de type "spread" (Les taux de réussite sont calculés sur 110 scénarios)

On peut voir que la prise en compte du caractère stochastique des obstacles n'a pas une grosse influence sur le taux de succès de l'approche RIS-path. Cela est sûrement dû au fait que pour tous les scénarios la vitesse des obstacles est mise à jour trop régulièrement. Cela a pour effet de réduire la variance de la probabilité de présence de l'obstacle.

3.5 Comparaison des deux versions de RIS-path : RIS-Path Projection et RIS-Path Horizon

Les deux versions de l'approche RIS-path présentées en Section 2.3 ont été comparées dans le but de garder la meilleure stratégie d'urgence pour l'approche finale. Pour cela les deux versions ont été comparées dans l'environnement "open" car l'on suppose que la présence d'obstacles statiques influence peu les résultats sachant que la différence entre les versions ne prend en compte que les obstacles dynamiques. En effet, pour la version "projection" le robot, s'il est à l'arrêt, ne peut être percuté par un obstacle statique et pour la version "horizon" les obstacles statiques ne sont pas impactés par la réduction d'horizon car leur projection sur le plan $t = 0$ reste la même.

Les deux versions ont été testées pour différents scénarios pour un robot holonmique puis non holonmique. Les scénarios choisis pour le robot holonmique sont de type "spread-150", le taux de réussite étant élevé pour l'ensemble des scénarios, choisir un type de scénario difficile permet d'avoir plus de chance de remarquer une différence notable entre les deux versions.

On obtient dans "open" pour des scénarios de type "spread-150" les résultats suivants 3.4 :

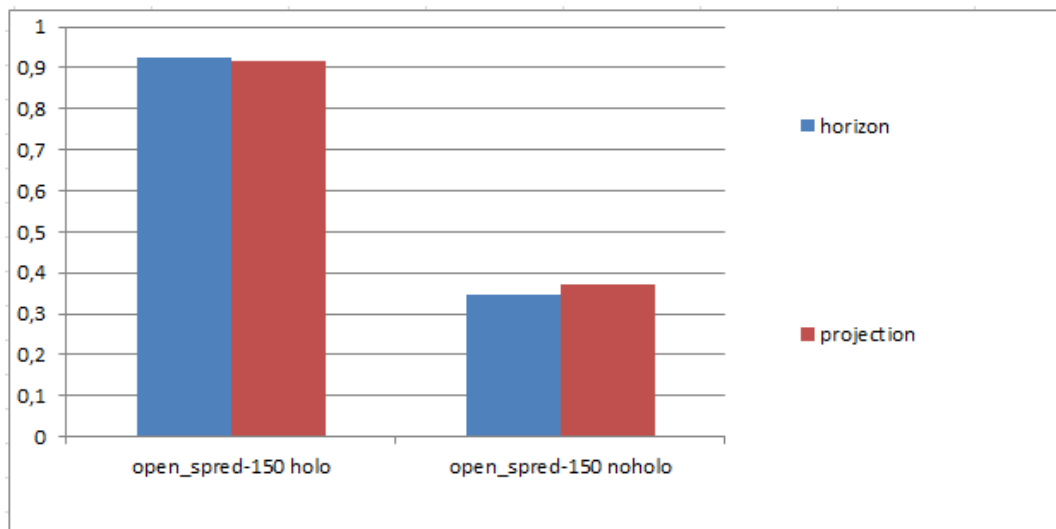


FIGURE 3.4 – Comparaison du taux de succès des deux versions de RIS-path sur trois type de scénarios dans l'environnement "open" (Le taux de succès est calculé sur 110 scénarios pour chaque type)

Les deux approches ont des résultats très proches. Dans la suite, c'est l'approche

horizon qui est implémentée.

3.6 Résultat du RIS-path stochastique

Dans cette dernière section, je vais vous présenter pour la version choisie de RIS-Path présentée en Section 3.5 et pour les approches de l'état de l'art optimisées en Section 3.3 un benchmark visant à mettre en lumière les points forts et les points faibles de l'approche RIS-Path par rapport aux approches de l'état de l'art présentées en Section 1.2.

3.6.1 Mouvement holonome du robot

Open

L'environnement "open" est un environnement classique pour la planification de trajectoire dans un environnement comportant des obstacles dynamiques. En effet, l'absence d'obstacle statique et la présence de très nombreux obstacles dynamiques a pour but de montrer l'aptitude des approches en présence de tels obstacles. Les résultats sont résumés par les Figures 3.5 et 3.6 :

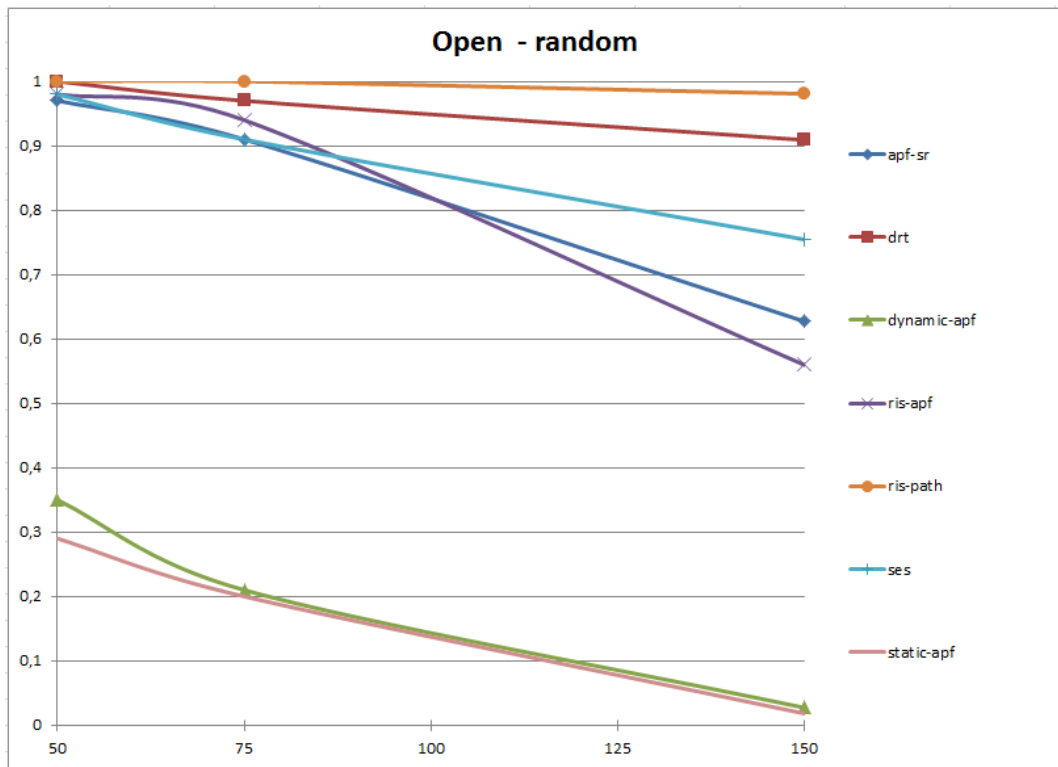


FIGURE 3.5 – Résultats des différentes approches de l'état de l'art sur l'environnement "Open" pour des scénarios de type "random" (Les taux de réussite sont calculés sur 110 scénarios)

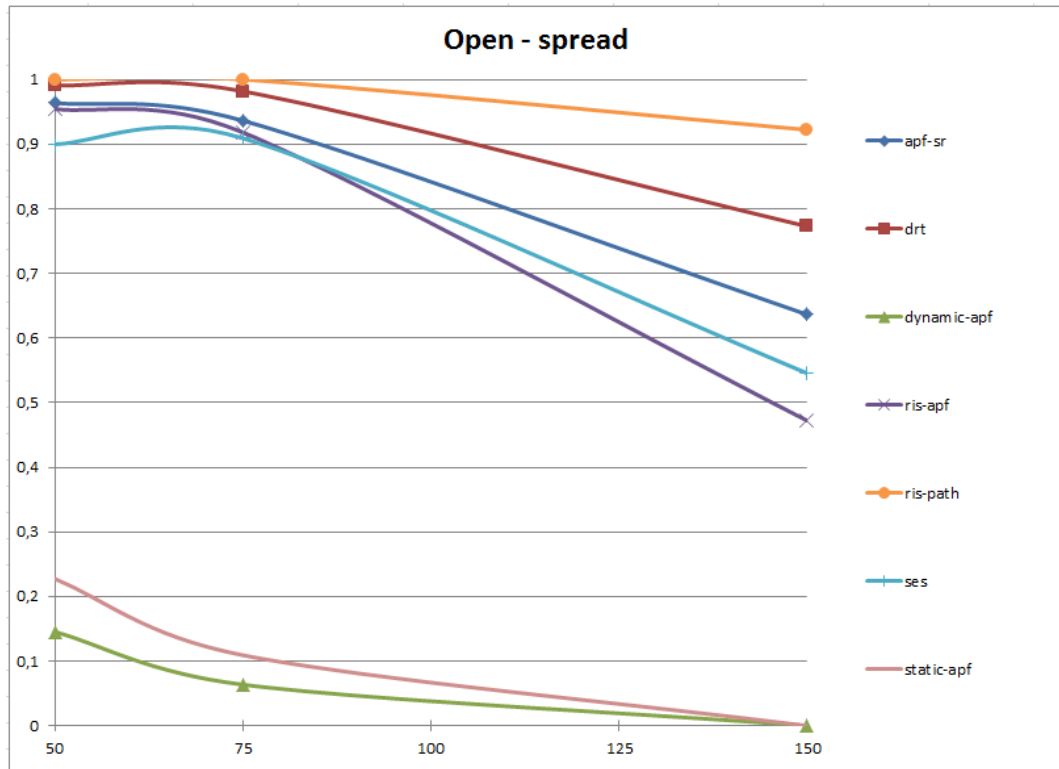


FIGURE 3.6 – Résultats des différentes approches de l'état de l'art sur l'environnement "Open" pour des scénarios de type "spread" (Les taux de réussite sont calculés sur 110 scénarios)

- Pour les scénarios "random" on voit se créer trois catégories d'approches :
- Les approches qui ne tiennent pas compte du comportement stochastique des obstacles :
 - statique-APF
 - dynamique-APF
 - Les approches qui tiennent compte du caractère stochastique des obstacles mais qui ne parviennent pas à gérer un trop grand nombre d'obstacles :
 - RIS-APF
 - APF-SR
 - SES
 - Et les approches qui tiennent compte du caractère stochastique des obstacles et qui arrivent à gérer un grand nombre d'obstacles :
 - DRT
 - RIS-path

On constate que l'approche RIS-path a un taux de succès sensiblement plus élevé que DRT. Cet écart est encore plus important dans les scénarios de type "spread" et lorsque le nombre d'obstacles augmente.

Lanes

Cet environnement "lanes" est intéressant car il met à l'épreuve la prévision que peuvent faire les approches. En effet, les approches trop réactives auront tendance

à s'engager sur la route dès que la première ligne est libre sans tenir compte de ce qui se passe sur les lignes suivantes. Bien souvent cela mène à des situations de non-retour. C'est également un cas de figure classique dans la planification de trajectoire dans un environnement comportant des obstacles dynamiques car il correspond à un robot devant traverser une route. Les résultats sont présentés dans les Figures 3.7 et 3.8 :

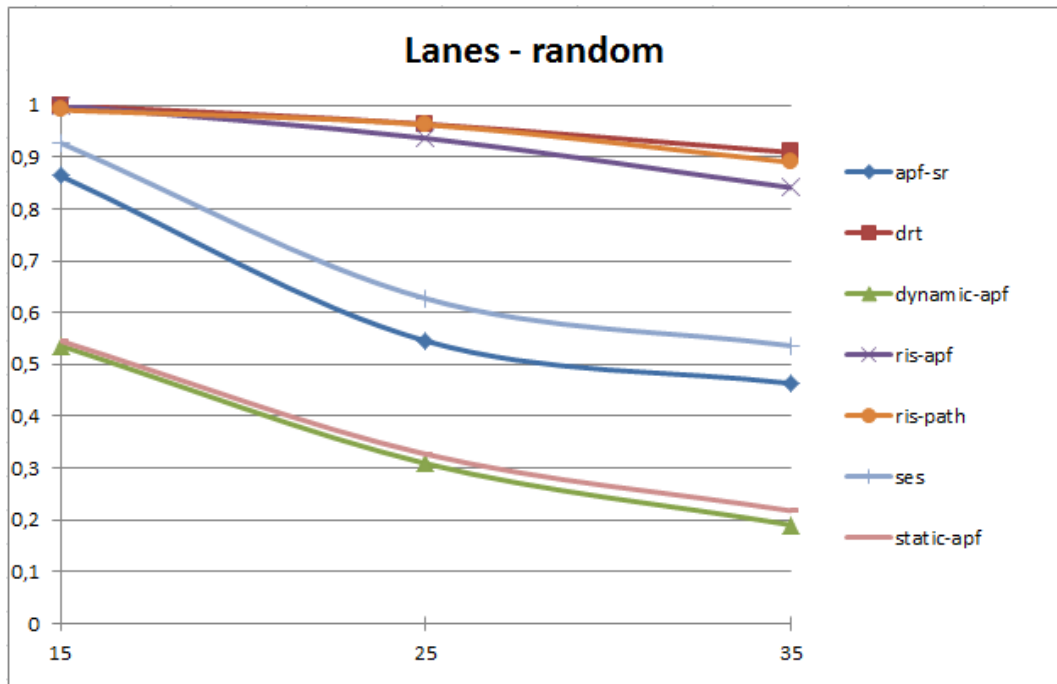


FIGURE 3.7 – Résultats des différentes approches de l'état de l'art sur l'environnement "Lanes" pour des scénarios de type "random" (Les taux de réussite sont calculés sur 110 scénarios)

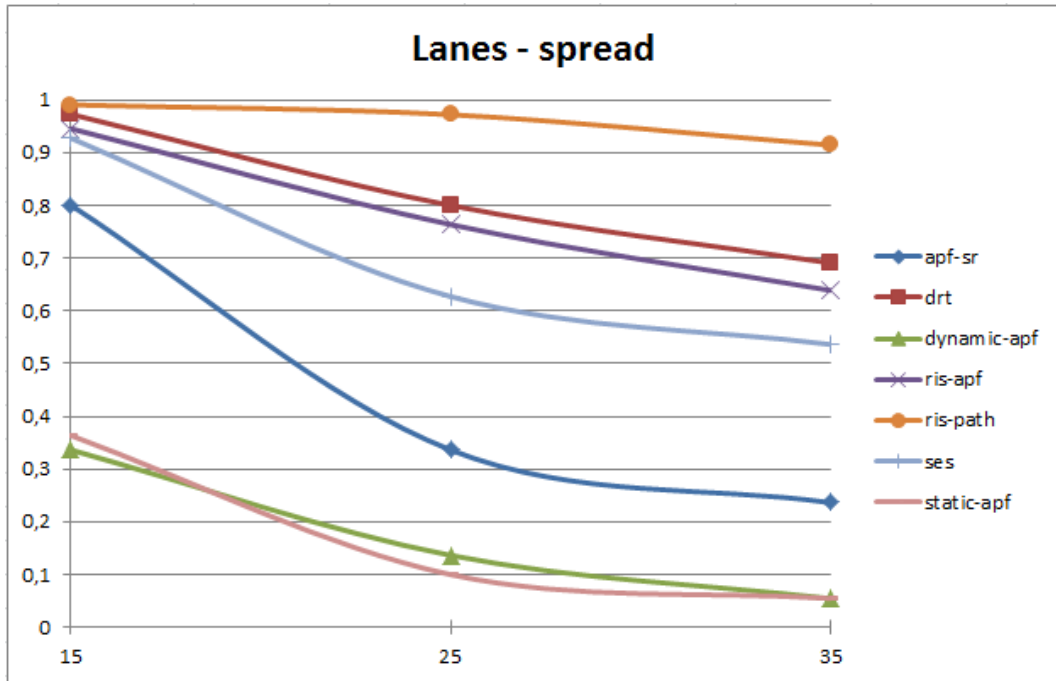


FIGURE 3.8 – Résultats des différentes approches de l'état de l'art sur l'environnement "Lanes" pour des scénarios de type "spread" (Les taux de réussite sont calculés sur 110 scénarios)

On peut noter sur les deux figures que les trois catégories les plus hautes persistent. On remarque quand même que le nombre d'obstacles n'étant pas aussi important que dans l'environnement "open" cet environnement favorise les approches qui permettent d'avoir la meilleur prévision sur le moyen terme et qui ainsi évitent les situations de non-retour. On remarque que la méthode RIS-APF (champs de potentiel artificiel construit grâce au RIS) est un peu moins performante que DRT et RIS-Path mais obtient un très bon résultat pour une approche réactive (APF-SR ayant un taux de succès bien inférieur) dans cet environnement par rapport aux résultats qu'elle avait obtenus dans l'environnement "open".

La Figure 3.8 nous confirme que l'approche RIS-Path supporte bien les obstacles ayant des vitesses maximales et minimales très disparates. Alors que les résultats des autres approches baissent en présence de ce type d'obstacles.

3.6.2 Mouvement non-holonome du robot

Les approches SES et DRT tiennent compte des contraintes de mouvement du robot. L'objectif de cette partie est de comparer les approches RIS-path et DRT pour un robot non-holonome. La comparaison n'est faite qu'avec DRT car l'objectif est de savoir si même sans tenir compte des contraintes de mouvement (Voir Section 3.2) RIS-path est toujours plus fiable que DRT pour des robots non-holonomes. Les expériences suivantes sont réalisées avec un robot de Dubins ayant les caractéristiques suivantes :

- $u_{max} = 0.09 \text{ rad/step}$
- $a_{max} = 0.7 \text{ pixels/seconde}^2$

Remarque 3.1 *Même si les RIS sont réalisés avec un cône comme RS pour le robot, le chemin lui tient compte des contraintes du robot non-holonyme. Le développement de cette recherche de chemin pour un robot non holonome a été réalisé par Pierre de Beaucorps.*

Open

Cette comparaison est faite sur l'environnement "open" pour des scénarios de type "spread". Les résultats sont résumés par la Figure 3.9 :

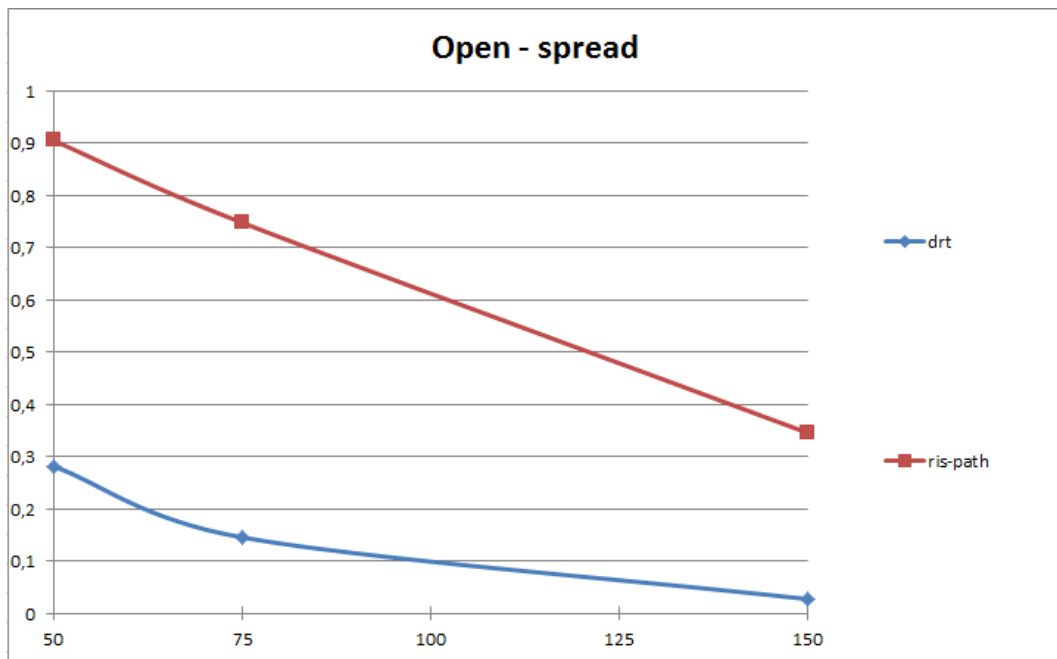


FIGURE 3.9 – Résultats des approches RIS-path et DRT sur l'environnement "Open" pour des scénarios de type "spread" (Les taux de réussite sont calculés sur 110 scénarios)

On constate que les deux méthodes sont très impactées par les contraintes du robot non holonome. Ce résultat est attendu car tous les scénarios sont beaucoup plus difficiles pour des robots non holonomes. Cependant, on voit que le taux de succès de l'approche DRT a bien plus chuté que l'approche RIS-path, même pour les scénarios impliquant 50 obstacles où RIS-path garde un taux de succès notable de 90,4%.

Conclusion

Dans ce stage j'ai développé des méthodes utilisées dans un modèle de planification de mouvement d'un robot en présence d'obstacles mobiles :

- J'ai étendu le modèle des RIS pour intégrer des contraintes de mouvements non-holonomes pour le robot (Section 2.1).
- Ensuite, j'ai considéré des modèles d'obstacles au mouvement stochastiques (Section 2.2).
- Finalement, j'ai intégré mes apports à une méthode de calcul de trajectoire en présence d'obstacles statiques et utilisant les RIS comme obstacles : RIS-Path (Section 2.3).
- J'ai ensuite réalisé un benchmark pour comparer les approches RIS et les approches de l'état de l'art (Section 3.6).

Les résultats suivants ont pu être constatés grâce aux évaluations expérimentales :

D'abord, l'approche RIS-path obtient de meilleurs résultats que les autres approches existantes dans l'état de l'art. Cela à la fois dans les scénarios inspirés du benchmark [5] et dans des scénarios plus compliqués.

De plus, le chemin obtenu par la méthode RIS-path est un chemin lisse (il a moins de discontinuités de courbure) que la sortie du DRT. Il est vrai qu'un véhicule holonome peut avoir des discontinuités de courbures dans sa trajectoire, cependant, un chemin lisse est une amélioration notable pour envisager une intégration sur un robot réel.

En outre, la prise en compte du caractère stochastique est très importante car il correspond à un cas plus réaliste. En effet, dans les cas concrets, les mouvements des obstacles ne sont pas connus de façon déterministe. Cependant, il n'est pas certain que les scénarios présentés dans [5] mettent suffisamment en avant cette amélioration car la variance des vitesses semble être trop faible. D'ailleurs dans [5] ainsi que dans les articles introduisant les approches stochastiques [3, 4, 6] il n'est pas mentionné si ces approches ont également été testées sur une approximation déterministe de la prédiction. Ceci aurait permis de distinguer la performance du modèle de prédiction de la performance de la méthode de planning seule. Nous n'avons pas encore eu le temps de trouver des scénarios appropriés pour mettre en évidence les apports de cette amélioration.

Par ailleurs, le fait d'introduire des heuristiques pour les cas d'urgences permet de grandement améliorer le taux de succès des approches RIS. En effet, les hypothèses faites pour modéliser les RIS ne sont pas respectées dans certaines situations critiques et la gestion de ces cas permet d'y remédier.

Enfin, les contraintes induites par le caractère non holonome d'un robot sont à considérer principalement dans les cas où le robot doit effectuer une manœuvre importante. Or, ce cas n'est pas vraiment envisagé lors de la recherche d'un chemin dans les RIS mais par des heuristiques d'urgences présentées en Section 2.3.3. Cela peut expliquer pourquoi la méthode RIS-path a un bon taux de succès pour les robots non holonomes. En dehors de ces cas d'urgences, le chemin est suffisamment dépourvu de manœuvres importantes à réaliser pour que le cône soit une bonne

approximation des RS du robot dans l'espace-temps. Cependant, tenir compte de ces contraintes ne devrait pas détériorer les résultats, il reste donc à résoudre ce problème.

Pour conclure, nous envisageons des pistes de développement telles que :

- Un raffinement du chemin de manière récursive qui est en cours de modélisation et de développement par Pierre de Beaucorps.
- Une amélioration du seuil dynamique permettant de limiter le ratio entre l'aire occupée par les RIS et la surface de l'environnement.
- Une méthode de planification utilisant les RIS permettant de générer un chemin et un profil de vitesse le long de ce chemin.

Bibliographie

- [1] John Canny and John Reif. New lower bound techniques for robot motion planning problems. In *Foundations of Computer Science, 1987., 28th Annual Symposium on*, pages 49–60. IEEE, 1987.
- [2] Animesh Chakravarthy and Debasish Ghose. Obstacle avoidance in a dynamic environment : A collision cone approach. *IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans*, 28(5) :562–574, 1998.
- [3] Hao-Tien Chiang, Nick Malone, Kendra Lesser, Meeko Oishi, and Lydia Tapia. Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2347–2354. IEEE, 2015.
- [4] Hao-Tien Chiang, Nathanael Rackley, and Lydia Tapia. Stochastic ensemble simulation motion planning in stochastic dynamic environments. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 3836–3843. IEEE, 2015.
- [5] Hao-Tien Lewis Chiang, Baisravan HomChaudhuri, Lee Smith, and Lydia Tapia. Safety, challenges, and performance of motion planners in dynamic environments. In *2017 International Symposium on Robotics Research (ISRR), Puerto Varas, Chile*, dec. 2017.
- [6] Hao-Tien Lewis Chiang, Baisravan HomChaudhuri, Abraham P Vinod, Meeko Oishi, and Lydia Tapia. Dynamic risk tolerance : Motion planning by balancing short-term and long-term stochastic dynamic predictions. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3762–3769. IEEE, 2017.
- [7] Pierre de Beaucorps, Anne Verroust-Blondet, Renaud Poncelet, and Fawzi Nashashibi. Motion planning among fast moving obstacles using reachable interaction sets. Article en cours de soumission, 2018.
- [8] Michael Erdmann and Tomas Lozano-Perez. On multiple moving objects. *Algorithmica*, 2(1-4) :477, 1987.
- [9] Andrey Fedotov, Valerii Patsko, and Varvara Turova. Reachable sets for simple models of car motion. In *Recent Advances in Mobile Robotics*. InTech, 2011.
- [10] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7) :760–772, 1998.

- [11] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1) :23–33, 1997.
- [12] Kikuo Fujimura and Hanan Samet. Motion planning in a dynamic domain. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 324–330. IEEE, 1990.
- [13] Shuzhi Sam Ge and Yun J Cui. Dynamic motion planning for mobile robots using potential field method. *Autonomous robots*, 13(3) :207–222, 2002.
- [14] Han-Pang Huang and Shu-Yun Chung. Dynamic visibility graph for path planning. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2813–2818. IEEE, 2004.
- [15] L Huang. Velocity planning for a mobile robot to track a moving target—a potential field approach. *Robotics and Autonomous Systems*, 57(1) :55–63, 2009.
- [16] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7) :846–894, 2011.
- [17] Nak Yong Ko and Bum Hee Lee. Avoidability measure in moving obstacle avoidance problem and its use for robot motion planning. In *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 3, pages 1296–1303. IEEE, 1996.
- [18] Jean-Claude Latombe. Robot motion planning (the kluwer international series in engineering and computer science), 1990.
- [19] Steven M LaValle. Rapidly-exploring random trees : A new tool for path planning. 1998.
- [20] Tomás Lozano-Pérez and Michael A Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10) :560–570, 1979.
- [21] Nick Malone, Hao-Tien Chiang, Kendra Lesser, Meeko Oishi, and Lydia Tapia. Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field. *IEEE Transactions on Robotics*, 33(5) :1124–1138, 2017.
- [22] Javier Minguez, Florent Lamiroux, and Jean-Paul Laumond. Motion planning and obstacle avoidance. In *Springer handbook of robotics*, pages 827–852. Springer, 2008.
- [23] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1) :33–55, 2016.
- [24] Nikos C Tsourveloudis, Kimon P Valavanis, and Timothy Hebert. Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic. *IEEE transactions on robotics and automation*, 17(4) :490–497, 2001.
- [25] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1928–1935. IEEE, 2008.

- [26] Jur Van Den Berg and Mark Overmars. Planning time-minimal safe paths amidst unpredictably moving obstacles. *The International Journal of Robotics Research*, 27(11-12) :1274–1294, 2008.
- [27] David Wilkie, Jur Van Den Berg, and Dinesh Manocha. Generalized velocity obstacles. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5573–5578. IEEE, 2009.
- [28] Quingnan Zhou. Pymesh. <https://pymesh.readthedocs.io>.