

DIPLOME D'ÉTUDES APPROFONDIES
de l'Institut National Polytechnique de Grenoble
Spécialité : Imagerie, Vision et Robotique

**CONCEPTION D'UN SYSTÈME DE LOCALISATION POUR
UN ROBOT MOBILE : UTILISATION D'UN TÉLÉMÈTRE
LASER ET PLACEMENT D'AMERS DANS
L'ENVIRONNEMENT**

Présenté et soutenu publiquement par

Cédric PRADALIER

Date de soutenance : 26 Juin 2001

Responsable de projet : M. Sepanta SEKHAVAT

Projet préparé au sein du projet SHARP,
au Laboratoire GRaphique, VIsion, Robotique
à INRIA Rhône-Alpes
ZIRST, 655 av. de l'Europe
38330 Montbonnot St. Martin
FRANCE

Remerciements

Je tiens à remercier spécialement monsieur Sepanta Sekhvat pour m'avoir proposé ce sujet, pour son encadrement et l'aide précieuse qu'il m'a accordée pour la rédaction de ce document.

Je souhaite adresser ma plus vive gratitude aux membres du jury pour m'avoir fait l'honneur d'être rapporteurs de ce travail.

J'adresse également mes remerciements à monsieur Christian Laugier, Directeur de Recherche INRIA et responsable du projet SHARP, pour m'avoir accueilli dans son équipe, accordé sa confiance et avoir mis à ma disposition le matériel nécessaire à la réussite de ce travail.

Je remercie enfin tous les membres de l'équipe SHARP et du service des moyens robotiques de l'INRIA Rhône-Alpes qui m'ont aidé à surmonter les différentes difficultés pratiques.

Résumé : Le but du travail présenté dans ce mémoire est la mise en oeuvre de méthodes de localisation grâce à des amers artificiels pour un robot équipé d'un capteur télémétrique.

Nous avons voulu traiter tout le processus de localisation : d'une part l'utilisation des données télémétriques pour le calcul de la position absolue du robot, et d'autre part le choix du placement des amers dans l'espace de travail.

Localisation du robot dans son espace de travail : Trois points sont abordés dans cette partie : l'extraction des données, leur mise en correspondance avec des données de référence et leur traitement par un filtre de Kalman.

Choix du placement des amers : Dans cette partie, nous traitons trois approches de ce problème : deux méthodes permettant de gérer analytiquement certains cas particuliers et une méthode numérique pour déterminer une solution optimale dans le cas général. Un planificateur de trajectoire utilisant ce placement optimal est alors envisagé.

Mots clés : Localisation, amers artificiels, mise en correspondance, optimisation des capacités de localisation.

Abstract : In this paper, we introduce the setting up of localization methods thanks to artificial landmarks for a robot equipped with a laser range finder.

We aim at tackling the whole localization process : firstly, the use of range data to get the absolute position of the robot and secondly, the placement of the landmarks in the workspace.

Concerning the localization process, we present the data extraction and matching methods, and the use of Kalman filtering techniques to merge different data sources.

Regarding the landmark placement problem, we define methods able to give an exact solution in special cases and a numerical, optimal solution in the general case. A trajectory planner using this optimal solution is then introduced.

Keywords : Localization, artificial landmarks, data association, optimization of localization capacities.

Table des matières

1	Introduction	1
1.1	État de l'art	4
1.1.1	La localisation	4
1.1.2	Téléométrie laser	4
1.1.3	Mise en correspondance	5
1.1.4	Intégration des données	6
1.1.5	Placement des amers	6
1.1.6	Planification orientée capteur	7
1.2	Présentation du plan	7
2	Module de localisation intégrant odométrie et données laser	9
2.1	Modèles utilisés	9
2.2	Extraction des points d'intérêt	11
2.2.1	Inadaptation des méthodes d'extraction de segments	11
2.2.2	Méthode utilisée	11
2.3	Identification des points d'intérêt	13
2.3.1	Approche naïve	14
2.3.2	Utilisation d'invariants géométriques	14
2.3.3	Amélioration par recherche de la clique maximale	17
2.4	Calcul de la position	21
2.5	Intégration des données de localisation dans un filtre de Kalman	23
2.5.1	Définition	23
2.5.2	Période d'échantillonnage et latence	25
2.5.3	Gestion de l'absence temporaire de localisation	26
2.6	Conclusion sur la localisation	26
3	Placement des amers dans l'espace de travail	28
3.1	Problématique et modèles	28
3.2	Solution analytique pour une portée infinie	29
3.2.1	Définition d'une zone interdite	30
3.3	Étude de maillages de l'espace de travail	32
3.3.1	Zone de visibilité d'une balise.	33
3.3.2	Étude d'une maille carrée	34
3.3.3	Étude d'une maille triangulaire	36
3.3.4	Commentaires	36

3.4	Solution numérique pour un capteur réel	37
3.4.1	Visibilité d'une balise : \mathcal{V}	37
3.4.2	Espace de localisation	37
3.4.3	Construction de l'espace de localisation	38
3.4.4	Optimisation de l'espace de localisation	40
3.4.5	Résultats de l'optimisation	41
3.5	Conclusion sur l'organisation de l'espace de travail	44
4	Ouverture sur la planification	46
4.1	Définition du problème	46
4.2	Problèmes soulevés par l'utilisation du graphe de localisation	47
4.3	Conclusion sur la planification	48
5	Conclusion	49

Chapitre 1

Introduction

Avec le développement de la puissance de calcul, la robotique autonome peut prendre son envol vers les applications extérieures. Alors que la robotique a débuté en intérieur, les applications en milieu extérieur se multiplient : intervention sous-marine, intervention spatiale ou robotique automobile. En intérieur, l'environnement étant modelé par l'homme, les applications bénéficient de caractéristiques très utilisées mais sur lesquelles on ne pourra pas compter dans un environnement naturel : sol plat, murs verticaux, angles droits, environnement de taille limité et facilement représentable en mémoire... A contrario, l'environnement extérieur étant rarement aussi parfait, les chercheurs en robotique sont amenés à explorer de nouveaux domaines. C'est dans ce contexte que travaille l'équipe SHARP de l'INRIA Rhône-Alpes, en développant des méthodes visant l'autonomie (manoeuvres automatiques, planification de trajectoire, évitement d'obstacles) ou la semi-autonomie (assistance à la conduite, déplacement de véhicules en trains virtuels). Pour expérimenter les méthodes développées dans ce cadre, l'INRIA dispose de véhicules autonomes : les Cycabs, voitures à 4 roues motrices équipées de différents capteurs odométriques (encodeurs sur les roues,...) et extéroceptif (capteur télémétrique laser et gyroscope).

Dans le cadre de ces expérimentations, les chercheurs de l'équipe SHARP ont besoin d'un outil de localisation robuste et efficace qui leur permette de maintenir une estimation aussi précise que possible de la position absolue du robot à partir des données de télémétrie laser.

La localisation est un problème fondamental de la robotique : en effet, il n'est pas possible de contrôler un robot sans estimation précise de sa position. Les données issues de capteurs proprioceptifs (odométrie) sont généralement insuffisantes car la position du robot ne peut être estimée que par rapport à sa position précédente. Pour compenser l'accumulation des erreurs, le processus de localisation a besoin de données externes absolues. Ces données sont alors être fusionnées avec les informations odométriques, afin d'avoir la meilleure estimation globale de la position. Dans l'étude que nous présentons ici, les données externes sont issues du repérage, par un capteur embarqué, d'amers artificiels placés dans l'environnement. La position de ces amers dans l'environnement doit être connue avec précision.

Le but du travail présenté dans ce document est donc de mettre en place des méthodes de localisation applicables sur les Cycabs. Ces méthodes doivent couvrir tout le processus de localisation : le choix des amers utilisés pour la localisation, celui de leur emplacement

FIG. 1.1 – Le Cycab



dans l'espace de travail, et le processus de localisation en lui-même. En premier lieu, nous aborderons donc les problèmes de l'extraction des données télémétriques, de leur mise en correspondance avec une carte en mémoire puis de la fusion des données extéroceptives et proprioceptives. En second lieu, nous verrons comment concevoir l'espace de travail du robot de façon à maximiser ses capacités de localisation. Une fois l'espace de travail bien conçu, il sera possible d'envisager un planificateur de trajectoire garantissant de bonnes capacités de localisation.

Avant de voir comment les problèmes que nous abordons ont été traités dans la littérature et de présenter le plan détaillé de notre étude, il est nécessaire de définir clairement quelques notions clés.

Capteur utilisé : Le capteur que nous utilisons sur les Cycabs est un télémètre laser Sick LMS-219. Les spécificités de ce capteur seront décrites en section 2.1, mais il est intéressant de donner un aperçu de ce capteur dès maintenant. Placé à une cinquantaine de centimètre du sol, sur la face avant du Cycab, il balaye un demi-plan horizontal en donnant un couple de mesures tous les demi-degrés : la distance mesurée par le capteur et l'intensité du faisceau réfléchi. L'intérêt de cette dernière donnée vient du fait que la valeur de l'intensité n'est remarquable que sur certains matériaux très réfléchissants de type catadioptré. Nous avons considéré ces mesures fiables jusqu'à 20 mètres (le constructeur annonce alors une erreur de mesure inférieure à 7 cm) On nommera **ouverture** angulaire l'angle α sur la figure 1.2. Sauf mention explicite du contraire, cette ouverture sera de 180 degrés.

FIG. 1.2 – Modèle du capteur utilisé

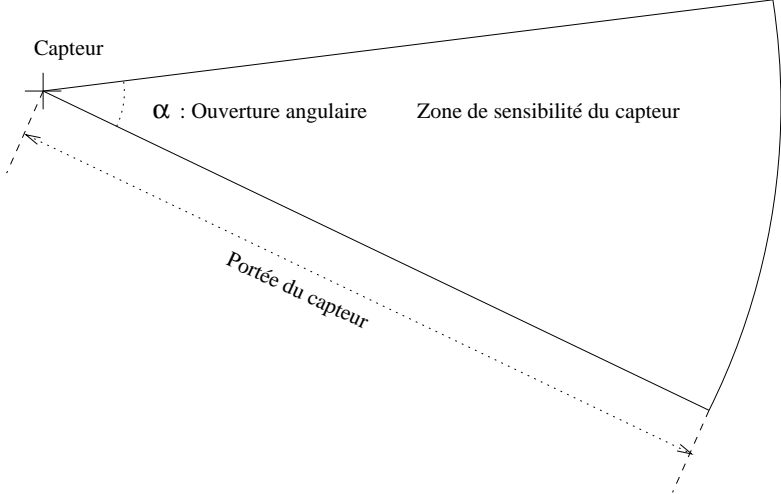
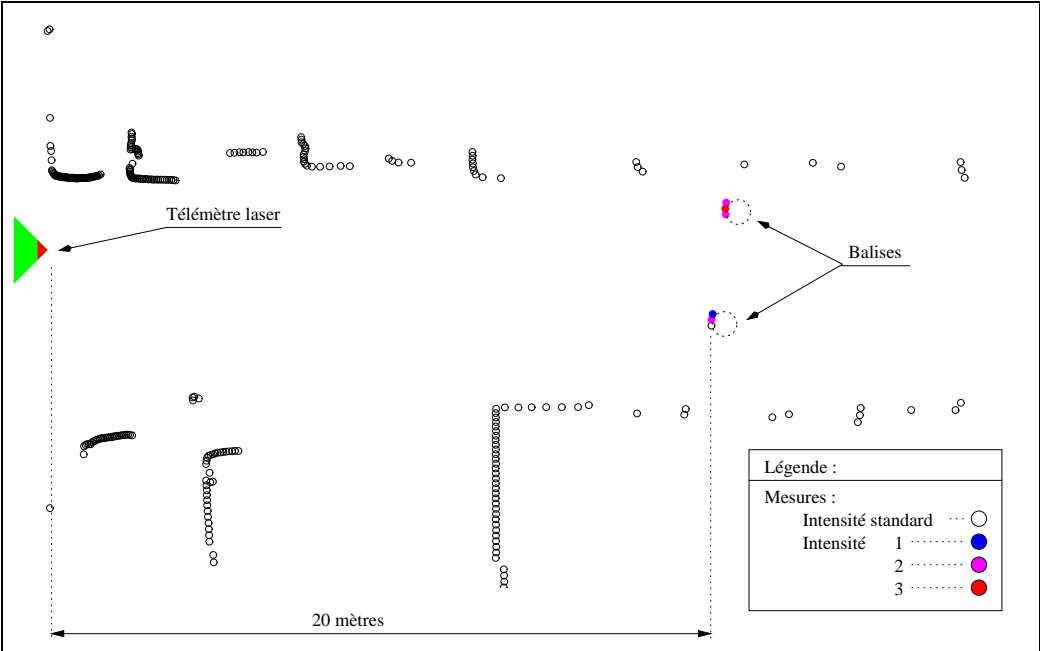


FIG. 1.3 – Exemple de résultat renvoyé par le Sick sur le parking de l'INRIA.



Espace de travail : Tout au long de ce document, on considérera que l'espace de travail dans lequel évolue le Cycab est polygonal. Ceci est assez réaliste car le Cycab est un véhicule conçu pour servir de véhicule d'appoint en milieu urbain. De plus, toutes nos expérimentations se sont déroulées sur le parking de l'INRIA Rhône-Alpes, environnement polygonal peuplé d'objets dont l'approximation polygonale est très proche de la forme réelle (par exemple, les véhicules particuliers du personnel de l'INRIA).

Configurations et espace des configurations : On désignera par configuration du robot le couple formé de sa position et de son orientation. Nous considérerons toujours qu'il existe un repère absolu dans lequel la position s'exprime par un couple (x, y) . En notant θ l'angle correspondant à l'orientation du véhicule, une configuration sera représentée par le triplet (x, y, θ) . L'espace des configurations est l'ensemble des configurations possibles.

Amers utilisés : Pour la localisation du robot, nous utiliserons des amers artificiels (désignés indifféremment sous le nom de "balises"). Ces repères seront décrits dans la section 2.1. Ce sont des cylindres de 15 cm de diamètre et 1 mètre de haut, recouverts d'une surface très réfléchissante, particulièrement adaptée à notre télémètre laser. La position de ces amers dans l'espace de travail sera connue avec précision. Le couple formé du polygone représentant l'environnement et de l'ensemble des positions absolues des amers sera désigné sous le nom de "carte de l'environnement" ou "modèle de l'environnement".

1.1 État de l'art

1.1.1 La localisation

La question de la localisation est un problème fondamental de la recherche en robotique. Des solutions complètes de localisation ont donc déjà été étudiées et présentées. Cependant, la majorité de ces approches correspondent à un robot particulier équipé de capteurs particuliers. Ainsi, Crowley, dans [Cro89], présente la description complète du système de localisation d'un robot équipé de capteurs ultrasons : modélisation des capteurs et de leur imprécision, mise en correspondance avec un modèle, intégration des données et filtrage par filtre de Kalman. De même, dans [BSB⁺00], une méthode complète est présentée, utilisant une caméra et des repères artificiels spécialement conçus pour leur facilité d'extraction. Alors que les méthodes précédentes essaient d'évaluer directement la position du robot (et l'incertitude sur cette position), d'autres méthodes évaluent la distribution de la probabilité que le robot soit en un point donné. Ainsi, dans [BFHS96], l'espace de travail est discrétisé en cases et, à chaque instant, la probabilité de trouver le robot dans chaque case est évaluée, à partir de la position précédente et des données des capteurs.

1.1.2 Télémétrie laser

De nombreuses équipes utilisent un capteur télémétrique laser pour résoudre leurs problèmes de localisation. Ainsi, dans [GNB00], Guivant et al. utilisent un tel capteur pour réaliser la localisation d'un véhicule automobile en extérieur. Pour cela, ils utilisent à la fois des repères artificiels et naturels. L'avantage principal de ce type de capteur

est la qualité des données recueillies. En effet, les données d'une caméra, par exemple, doivent être extraites de l'image (contours, points d'intérêts, mise en correspondance, reconstruction 3D...) et celles issues de capteurs infrarouges ou ultrasons sont généralement entachées d'une grande incertitude. A contrario, les données de télémétrie laser ont de nombreux avantages : précision des distances et des angles mesurés, grande portée et projection aisée des données dans l'espace de travail du robot. Pour toutes ces raisons, Wallner dans [Wal97], Charbonnier et Strauss dans [CS] et Arras et al. dans [ATS00] utilisent un capteur laser pour localiser un robot en intérieur en détectant les segments formés par les murs. Dubrawski et Siemiatkowska [DS98] et Lu et Miliotis [LM94] ont choisi quant à eux de ne pas faire reposer leurs méthodes sur les spécificités (murs, portes) contenues dans l'environnement. Lu et Miliotis décrivent une méthode permettant de déterminer la transformation – rotation et translation – qu'a subi le robot entre deux balayages. Dubrawski et Siemiatkowska utilisent pour leur part une notion d'histogramme d'angle pour estimer la rotation subie par le capteur. Enfin, Guibas et al., dans un article très théorique ([GMR95]), présentent des algorithmes permettant de mettre en correspondance l'environnement capté par un capteur télémétrique et une carte de l'environnement afin d'estimer la position du capteur. Dans le cadre de cette étude, les techniques d'extractions de segments se sont révélées inadaptées. Nous avons donc choisi de définir un **automate de Mealy** capable de traiter les données télémétriques brutes pour en extraire les positions des balises potentielles.

A partir des données télémétriques laser, le calcul final de la position du robot se ramène souvent à un problème géométrique (intersection de cercles ou de droites). Dans leur article [BG94], Betke et Gurvits fournissent une méthode permettant de déterminer la position du robot lorsque l'on dispose des angles relatifs entre trois balises. L'utilisation de représentations en **nombres complexes** des objets manipulés permet d'obtenir une solution élégante à ce calcul. Appliquée au cas de données issues de télémétrie laser, cette représentation simplifiera énormément les expressions utilisées (cf. section 2.4).

1.1.3 Mise en correspondance

Quel que soit le capteur utilisé, un problème conséquent vient du besoin d'associer les données venant des capteurs aux données décrivant l'environnement en mémoire. Ainsi, dans [BSTL95], Becker et al. définissent avec précision le problème de la navigation en utilisant des repères, qu'ils soient artificiels ou naturels : une bonne balise artificielle doit en général être facile à détecter et facile à identifier. Dans le cas contraire, l'ensemble des balises détectées doit être mis en correspondance avec l'ensemble des balises de la carte de l'environnement. De façon très générique, le procédé de mise en correspondance de deux ensembles peut être exécuté en utilisant la distance de Hausdorff ([Ols98] et [Ruc95]). Cependant, la distance de Hausdorff est surtout efficace lorsque l'on essaye de faire correspondre deux ensembles de points suffisamment grand ; pour des ensembles trop petits (typiquement 2 à 6 points dans nos expérimentations), ses résultats ne sont pas satisfaisants car trop sensibles aux erreurs de mesures.

Le problème de la mise en correspondance est un problème essentiel de la vision par ordinateur : [Sch96] donne des méthodes pour associer des données issues d'une caméra et un modèle de l'environnement. Ces méthodes, comme celle présentée dans [TCT98] utilisent généralement des propriétés conservées par l'acquisition vidéo : des invariants

projectifs. Dans le cas d'un télémètre laser, nous avons choisi d'utiliser des **propriétés invariantes aux transformations orthonormales** (rotation, translation). Après étude, les notions d'aires et de longueurs nous ont paru les mieux adaptées à notre problème.

Grâce à la notion d'invariant, pour chaque objet observé, on obtient un ensemble de correspondance potentielle. Pour choisir le bon candidat, on pourra utiliser les méthodes classiques utilisées en vision [Sch96] ou celle présentée dans [BNRDW00] qui utilise la théorie des graphes.

1.1.4 Intégration des données

Les données issues de la phase de mise en correspondance permettent d'obtenir une estimation de la position du robot. Pour fusionner les données issues des différents capteurs extéroceptifs (laser, caméra, GPS...) avec celles issues des capteurs proprioceptifs (odométrie) et pour intégrer ces données au cours du temps, des méthodes particulières sont nécessaires. Le filtre de Kalman ou sa version non linéaire, le filtre de Kalman étendu sont généralement utilisés pour réaliser cette fusion : [Cro89] et [Lam96] expliquent toutes les phases de la construction de ce filtre. Pour étendre cette notion, [CB96] présente un filtre du deuxième ordre inspiré du Kalman, [HBBC96] étudie la difficulté de maintenir à la fois l'état du robot et l'état de l'environnement et [RB97] étudie la difficulté de fusionner des données dont les fréquences d'acquisition sont différentes. Par ailleurs, [BSS96] et [AO01] présentent des alternatives au filtre de Kalman utilisant la méthode des moindres carrés ou des raisonnements probabilistes. Enfin, Large et al. dans [LHSL00] utilise un réseau de neurones pour apprendre le modèle cinématique d'un véhicule automobile et filtrer les données odométriques. De même, [BF96] décrit une méthode permettant de fusionner les données issues de l'odométrie et du gyroscope (la gyrodométrie) pour obtenir une meilleure précision sur l'estimation de la position du robot : le gyroscope n'est pris en compte que lors des variations particulièrement rapides de l'orientation (chocs, passage sur des irrégularités du sol, etc...). Dans notre cas, le **filtre de Kalman étendu** s'est révélé tout à fait adapté. Cependant, il a fallu mettre en place des mécanismes particuliers pour gérer l'**asynchronisme** entre données laser et odométrie.

1.1.5 Placement des amers

Dans toutes les méthodes des sections précédentes, celles qui utilisent des repères artificiels supposent que ces repères sont déjà installés dans l'espace de travail. Cependant, si on désire pouvoir utiliser ces repères pour la localisation en tout point de l'espace de travail et si on veut minimiser le nombre de balises utilisées, leur placement est un problème compliqué. Dans [TOLA95], le problème du placement de balises dans un espace de travail polygonal est défini. La méthode cherche à minimiser les erreurs de localisation dans le cas où un seul amer permet de connaître la position et l'orientation du robot. Une autre façon de garantir les capacités de localisation est de résoudre le problème du placement dans un polygone simple (cf. section 3.3) puis de répliquer ce polygone simple sur tout l'espace de travail avec les techniques évoquées dans [Gud00]. Enfin, Briggs s'attaque à un problème symétrique dans [BD96] : comment placer un capteur (par exemple une caméra) dans un environnement polygonal pour observer au mieux le robot. Elle utilise dans cet article la notion de polygone de visibilité (ensemble des points visibles d'une position donnée) décrite dans [GMR95] et [AGTZ98], notion qui sera très utile en section 3.4.

A notre connaissance, la spécificité de notre problème n'avait pas été abordée dans la littérature. En effet, pour pouvoir se localiser il est nécessaire de détecter plusieurs amers. Nous avons donc défini une méthodologie permettant de trouver un **placement optimal des amers dans l'espace de travail** afin d'assurer les capacités de localisation dans la majeure partie de l'espace des configurations.

1.1.6 Planification orientée capteur

Lorsque l'on dispose de l'ensemble des balises visibles en tout point de l'espace de travail, on peut envisager un planificateur de trajectoire qui prévoit en certains points de la trajectoire, voire en tous, les repères qui vont être visibles et ceux dont l'utilisation va permettre la localisation la plus précise. Ainsi, dans [MA98], Madsen et Andersen introduisent des critères permettant d'évaluer la qualité d'un triplet de balises. De même, dans [RGI96], des méthodes d'optimisation numériques permettent de déterminer le sous-ensemble des balises potentiellement visibles qui pourra être détecté de façon optimale. Par ailleurs, [AMdIES98] décrit un planificateur qui détermine à quels moments le robot devra essayer de se localiser par rapport aux repères de l'environnement, en déterminant s'il est plus intéressant de s'arrêter pour lancer la localisation ou de le faire en continuant le mouvement. Enfin, [LL92] présente un planificateur permettant de trouver une trajectoire sûre dans un espace où la localisation extéroceptive n'est possible que dans certaines zones de l'environnement (disques autour des balises).

Le placement optimal des balises dans l'espace de travail a permis de mettre en évidence la nécessité de définir un **planificateur de trajectoire spécifique**. Les trajectoires générées par ce planificateur auront l'avantage de garantir les capacités de localisation à chaque instant.

1.2 Présentation du plan

L'ensemble du processus de localisation sera étudié dans ce document. Le chapitre 2 sera consacré à la localisation par triangulation. En 2.2 l'extraction des balises des données laser sera étudiée en utilisant un automate de Mealy. Les méthodes de mise en correspondance des balises détectées avec la carte de l'environnement seront décrites en 2.3 en utilisant des techniques issues de la recherche en vision et la notion de clique maximale. A partir des données de la mise en correspondance, il suffit de calculer la position du robot. C'est ce qui sera abordé en 2.4 en portant une attention particulière aux conséquences des erreurs de mesures. Finalement, l'intégration des données par un filtre de Kalman sera décrite en 2.5.

Le chapitre 3 traitera d'un aspect de la localisation en amont de la triangulation : le placement des balises dans l'espace de travail. Trois approches de complexité croissante seront abordées selon les modèles du capteur et de l'environnement utilisés : en 3.2 on supposera que l'on dispose d'un capteur de portée infinie mais dont l'ouverture angulaire est limitée et inférieure à π radians (cas d'une caméra par exemple). Le placement des balises peut alors être déterminé analytiquement. En 3.3, on étudiera un environnement polygonal grand par rapport à la portée du capteur et un capteur observant un demi-plan (ouverture de π radians) : le capteur Sick correspond à un tel capteur. On peut alors montrer que s'il est possible de définir le placement des balises dans un polygone simple

de façon à pouvoir se localiser en chacun de ses points, on peut couvrir l'espace de ce polygone et garantir la localisation en tout point. Cependant, cette idée demande de placer des balises dans l'espace de travail, elles devront donc être considérées comme des obstacles par le robot. Enfin, avec un capteur d'ouverture et de portée quelconques, on peut définir l'**espace de localisation** du robot (cf. section 3.4) et utiliser des méthodes d'optimisation numérique pour placer les balises. Celles-ci seront placées de façon à maximiser l'espace où la localisation est possible dans l'espace de travail.

Enfin, à partir de la notion d'espace de localisation, on verra qu'il est possible d'envisager la construction d'un planificateur de trajectoire tenant compte des capacités de localisation du système. Typiquement, pour un couple de points d'une même composante connexe de l'espace de localisation, il sera possible de planifier une trajectoire garantissant une capacité de localisation en tout point (voire un nombre minimum de balises visibles). On pourra aussi, dans le cas général, mettre en évidence les parties de la trajectoire où la localisation ne pourra plus être assurée et chercher à minimiser ces portions. Le chapitre 4 permettra de décrire les pistes ouvertes par les notions précédentes.

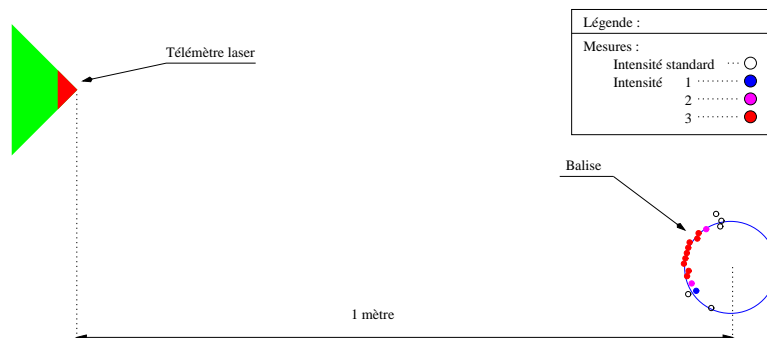
Chapitre 2

Module de localisation intégrant odométrie et données laser

2.1 Modèles utilisés

Les méthodes de localisation qui vont être présentées dans ce chapitre ont d'abord été pensées pour satisfaire les besoins de repérage du véhicule Cycab, elles sont donc adaptées à l'organisation matérielle du Cycab. Le rayon laser du capteur Sick balaye un plan situé devant le Cycab et renvoie, tous les demi-degrés, un couple (d, i) (cf. figure 2.1). d est la distance, mesurée grâce au temps de vol, entre le capteur et le premier obstacle rencontré par le rayon laser, bornée par 80 mètres pour le mode de fonctionnement que nous utilisons. i est l'intensité de l'écho reçu par le capteur, codé entre 0 et 7. La valeur 0 correspond à l'intensité courante d'un écho. Toute valeur non nulle caractérise un matériau très réfléchissant : phare de voiture, miroir... La précision du capteur est de l'ordre de 0.5% de la distance mesurée : 5 cm à 10 mètres, 10 cm à 20 mètres.

FIG. 2.1 – Exemple de données renvoyée par le Sick sur une balise à 1 mètre.



On peut donc concevoir un système de localisation fondé sur la détection de balises réfléchissantes (cf. figure 2.1 et 1.3) et leur mise en correspondance avec les données d'une carte de l'environnement. La difficulté qui apparaît alors vient du besoin de repérage en milieu urbain : les phares des voitures et leurs plaques d'immatriculation sont en général

conçus pour être très réfléchissants et sont placés à une cinquantaine de centimètre du sol. Dans un environnement de type parking, les véhicules en stationnement constitueront donc autant de source d'erreurs. L'extraction et l'identification des balises constituent donc des difficultés particulières qui seront traitées dans les sections 2.2 et 2.3. Il faut noter qu'il n'est pas intéressant de chercher à résoudre ce problème en choisissant une hauteur particulière du capteur. Tout d'abord, selon la marque et le type de véhicule, les hauteurs de la plaque d'immatriculation et des phares varient énormément. Ensuite, pour des applications de suivi de véhicule, il peut être intéressant d'utiliser la détection de ces surfaces.

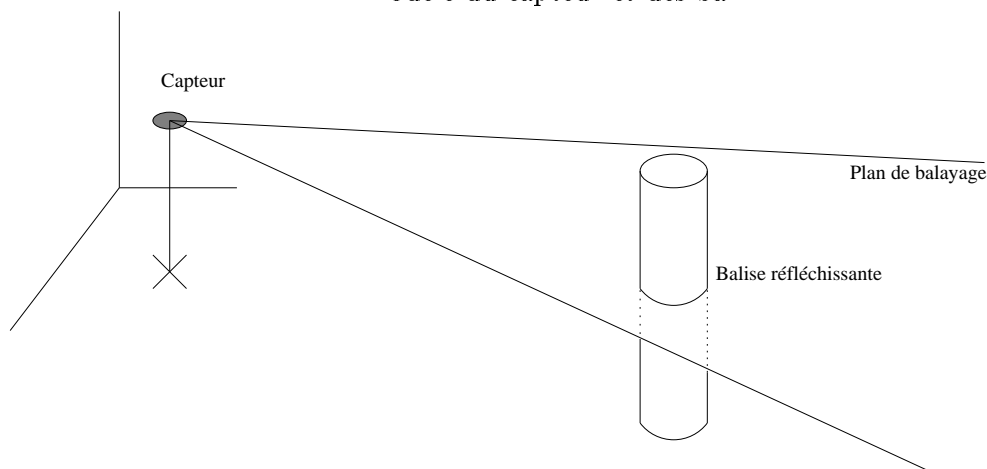
A partir des données d'identification, le calcul de la position est relativement simple : il sera traité en section 2.4 en mettant l'accent sur les conséquences des erreurs de mesure.

En outre, le Cycab dispose d'un gyroscope permettant de mesurer la vitesse angulaire du robot. Le capteur dont nous disposons n'est pas d'une très grande qualité mais en utilisant des techniques de filtrage et de fusion appropriée, il pourra tout de même être utilisé pour améliorer l'estimation de la position.

Enfin, le Cycab dispose de données odométriques obtenues à partir d'encodeurs fixés sur les roues du véhicules. Ces données sont elles aussi assez peu fiables car des problèmes de conception mécaniques du robot provoquent des glissements même à faible vitesse. Toutes ces données peuvent être fusionnées grâce à des techniques de filtrage telle que le filtre de Kalman (section 2.5).

Choix des balises : Pour la localisation du Cycab, nous avons besoin d'amers facilement détectables, quelles que soient la position et l'orientation du robot. Or, même pour des surfaces aussi réfléchissante que les catadioptrés, la qualité de la détection diminue très fortement lorsque l'angle entre la surface et le rayon incident (laser) augmente. Cette constatation nous a amenés à choisir des balises cylindriques. On verra par ailleurs (équation 2.1) que ce choix réduit naturellement l'erreur sur l'estimation de la position angulaire du capteur.

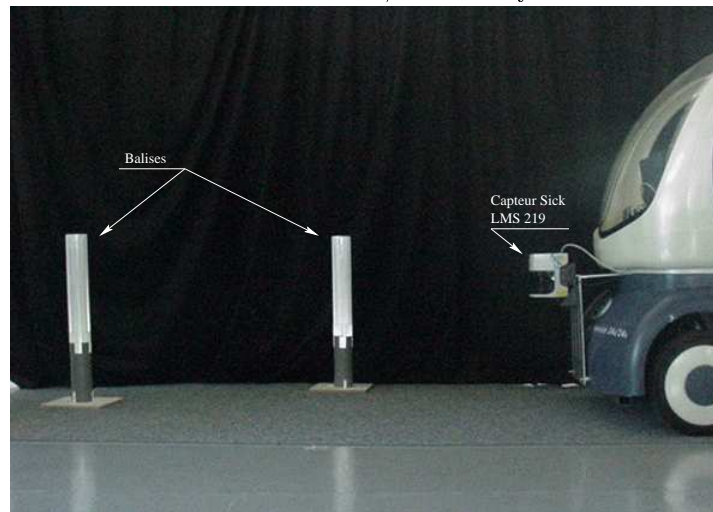
FIG. 2.2 – Modèle du capteur et des balises



Par ailleurs, l'horizontalité du plan de balayage n'est pas assurée à tout instant : le plan peut bouger en fonction des imperfections de la route ou du nombre de personnes

dans le véhicule. Or une variation de seulement un degré du plan de balayage provoque un déplacement vertical de 34 cm sur une balise située à 20 mètres du capteur. Il faudrait donc que les balises soient des cylindres dont la surface réfléchissante couvre une cinquantaine de centimètres autour du plan de balayage pour pouvoir assurer une détection fiable jusqu'à 20 mètres. Dans le cadre de nos expérimentations, les balises étaient des cylindres de 15 cm de diamètre, couvrant seulement 25 cm autour du plan de balayage et qui se sont révélées fiables jusqu'à 15 mètres.

FIG. 2.3 – Balises, Sick et Cycab



2.2 Extraction des points d'intérêt

2.2.1 Inadaptation des méthodes d'extraction de segments

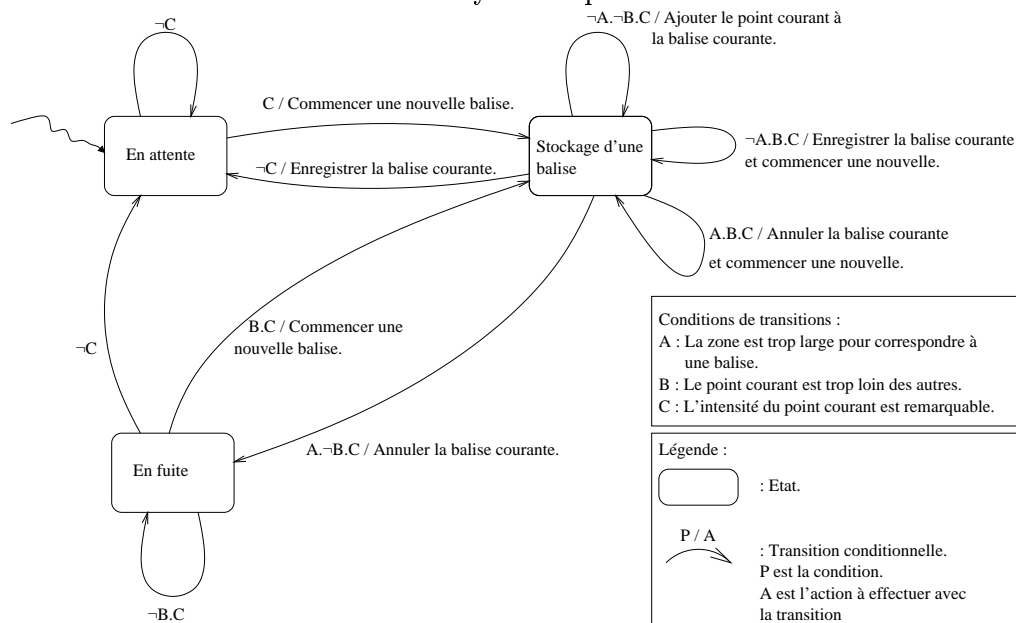
Au cours des dernières années, les capteurs télémétriques Sick ont été de plus en plus utilisés en intérieur comme en extérieur. L'approche habituelle commence par l'extraction des segments présents dans l'image. Cette donnée est ensuite utilisée pour le repérage : soit par mise en correspondance avec une carte de l'environnement, soit par extraction des intersections comme points d'intérêt. Cependant, un robot de type voiture est amené à se déplacer dans des environnements difficilement mémorisables. En effet, dans le cas d'un parking, on ne peut pas préétablir de carte précise de l'environnement, car celle-ci dépendrait de la position des véhicules en stationnement. Pour cette raison, les solutions de localisation utilisant des repères naturels (murs, portes,...) – solutions très utilisées en robotique intérieure – ont été rejetées.

2.2.2 Méthode utilisée

Puisque les méthodes d'extraction de segments ne sont pas adaptées à l'extraction de nos repères, une autre méthode doit être envisagée. Pour utiliser au mieux nos données, cette méthode doit prendre en compte leur aspect séquentiel et l'indication d'intensité

donnée par le capteur. L'utilisation d'un automate de Mealy¹ pour traiter cette séquence de mesures et extraire les balises potentielles nous a donc paru assez naturelle. De plus, l'utilisation d'un automate permettra, de faciliter le portage de l'extraction sur un micro-contrôleur dédié. Cette automate doit répondre aux spécifications suivantes :

FIG. 2.4 – Automate de Mealy utilisé pour l'extraction des balises.



- Il doit détecter les zones où l'intensité est remarquable.
- Dans de telles zones, il doit détecter les zones trop larges (plaques d'immatriculation par exemple).
- Il doit aussi détecter les ruptures dans la suite des points : i.e. les variations trop brutales de la distance mesurée, indiquant que le rayon laser touche un objet devant ou derrière la balise courante.
- Enfin, il doit, pour chaque balise détectée et validée, retourner l'ensemble des points considérés comme étant sur la balise.

Algorithme représenté par l'automate de la figure 2.4 : De façon générale, l'automate recherche les suites de points d'intensité remarquable dont la largeur est raisonnable et qui ne présentent pas de rupture. Lorsqu'une rupture est détectée, la balise courante est validée et la recherche d'une nouvelle peut commencer. Si la suite de points est trop large (état "en fuite"), l'automate ignore tous les points jusqu'à une rupture ou jusqu'à la fin de la zone dont l'intensité est remarquable.

Localisation d'une balise à partir de sa suite de points : Comme on l'a vu précédemment, l'intensité réfléchiée par une balise dépend de l'angle entre le rayon incident et la normale à la surface de la balise au point d'impact. Il semble donc raisonnable

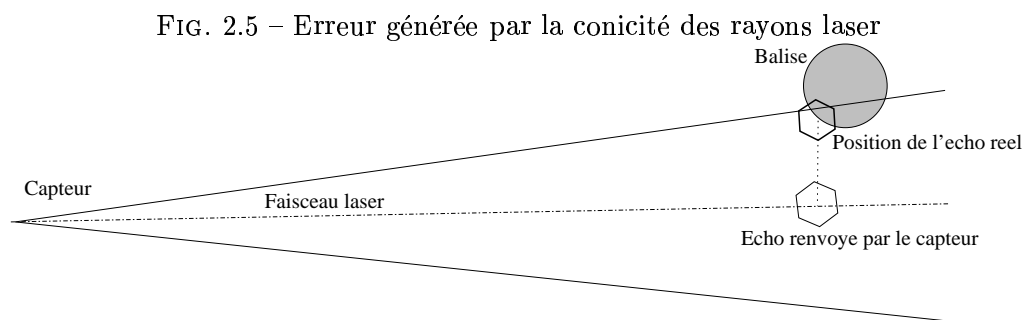
¹Automate d'états fini où les actions éventuelles sont associées aux transitions

de faire l'approximation suivante : la direction du capteur à la balise détectée est approximativement égale à la moyenne des directions du capteur aux points relevés sur la balise, pondérées par l'intensité mesurée. En notant C le capteur, B la balise, $\{P_i\}_{i=1..n}$ les points mesurés et $I(P_i)$ l'intensité associée à P_i ,

$$\vec{CB} = \frac{\sum_i I(P_i) \cdot \vec{CP}_i}{\sum_i I(P_i)} \quad (2.1)$$

Pour ce qui est de la distance CB , sur une balise circulaire, le point le plus près du capteur est le point situé sur la droite liant le capteur au centre de la balise. On peut donc considérer que la plus petite distance mesurée sur la balise est une bonne approximation de la distance du capteur à la surface de la balise. Finalement, en ajoutant le rayon de la balise, on obtient une bonne estimation de sa position réelle par rapport au capteur. Expérimentalement, les résultats obtenus sont tout à fait satisfaisants (cf. figure 2.1).

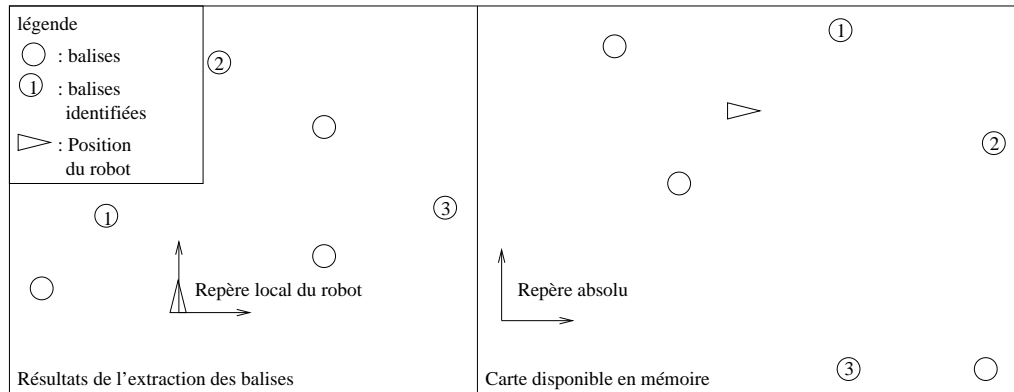
Une autre méthode envisageable consisterait mettre en correspondance un cercle du diamètre de la balise avec les points mesurés, par une méthode telle que la méthode des moindres carrés. Cependant, la différence entre la réalité physique du capteur et sa modélisation rendent cette approche problématique. En effet, les rayons laser envoyés par le capteur sont en réalité coniques, ce qui provoque la situation illustrée en figure 2.5. Les points mesurés ne sont donc pas obligatoirement sur la balise. Nous avons donc préférée l'approximation précédente.



2.3 Identification des points d'intérêt

L'extraction des balises potentielles des données laser fournit la position de ces balises par rapport au capteur. Cependant, pour pouvoir se localiser, il est nécessaire de déterminer les correspondances entre les balises relevées et la carte en mémoire. La difficulté que l'on rencontre ici vient de l'approche du problème qui a été choisie : pour faciliter leur construction et surtout pour limiter leur taille, les balises ne sont pas distinguables. En effet, sur nos plots cylindriques réfléchissants, le capteur laser renvoie toujours le même écho, quelle que soit sa position et quelle que soit la balise. Plusieurs solutions à ce problème sont envisageables : en supposant que l'on connaisse une estimation de la position du robot et en connaissant sa précision, il est possible, dans certaines conditions, d'identifier les balises (section 2.3.1). Cependant, cette méthode manque de robustesse : en utilisant des algorithmes issus des techniques de vision par ordinateur, on peut obtenir des résultats beaucoup plus fiables (section 2.3.2 et 2.3.3).

FIG. 2.6 – Problème de l'identification : les balises vues par le robot (à gauche) sont numérotées sur la carte en mémoire (à droite).



2.3.1 Approche naïve

La première approche que nous avons envisagée pour résoudre le problème de l'identification utilise les données odométriques : connaissant la position $k - 1$ et les données odométriques depuis cette position, on obtient une estimation de la position k (estimation qui peut être améliorée par filtrage, cf. section 2.5). A partir de cette position et de la carte en mémoire, on peut déterminer quelles sont les balises qui doivent être visibles et la position où elles devraient être détectées. On associe alors à chacune de ces balises, la balise détectée qui en est la plus proche. Évidemment, cette méthode n'assure aucune robustesse. Pour pouvoir fonctionner, le système doit remplir les conditions suivantes :

- Les échos ne correspondant pas à des balises doivent être rares et éloignés des échos corrects, ce qui, par définition, est impossible à assurer.
- La localisation doit être faite avec une fréquence suffisamment grande pour que les erreurs de l'odométrie ne s'accumulent pas trop. La configuration de notre plateforme ne permettait au mieux que trois localisations par seconde.
- Pour la même raison que ci-dessus, le robot doit se déplacer relativement lentement, ce qui est très limitant pour un véhicule automobile.
- Le robot ne doit pas se perdre : en effet, si le robot entre dans une zone où il ne peut pas se localiser (en conduite manuelle par exemple), les erreurs d'odométrie s'accumulant, il ne pourra pas se relocaliser lorsqu'il verra de nouveau des balises.

Toutes ses conditions aboutissent à la conclusion suivante : on ne peut pas utiliser cette méthode pour un fonctionnement générique et robuste du robot. Toutefois, nous l'avons utilisée dans le cadre de manoeuvres de parking automatiques et elle s'est révélée suffisante dans la plupart des cas.

2.3.2 Utilisation d'invariants géométriques

En observant les conditions nécessaires à la méthode précédente, on peut déterminer une cause d'échec générale : dans cette méthode, on cherche à identifier les balises en n'utilisant que leur position. On n'utilise donc qu'une information locale sans tenir compte de la globalité des données : distance entre les balises, forme des triangles formés par les triplets de balises. En effet, on peut constater que la transformation qui amène les

balises du repère absolu vers le repère local du robot est la composition d'une rotation et d'une translation. Ces transformations ne modifiant ni les angles ni les distances (ce sont des isométries), l'image d'un triplet de balises sera un triplet formant un triangle dont les longueurs des arêtes, les angles et l'aire ne seront pas modifiés (cf. figure 2.6). En identifiant des triplets de balises (comme dans [Sch96] ou [TCT98]), on obtient donc une information beaucoup plus riche car traduisant les relations indéformables entre les balises.

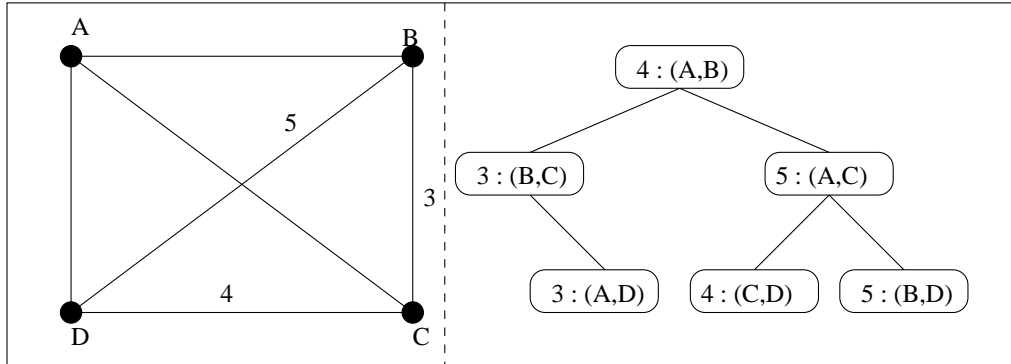
Problèmes liés aux symétries de l'environnement : S'il existe des motifs qui se répètent dans la carte des balises, il sera impossible de se localiser sans utiliser les données odométriques. Par exemple, si le motif observé est présent à trois endroits de la carte, on ne pourra pas identifier les balises de manière unique sans estimation de la position. Cependant, si on dispose de cette estimation, on pourra déterminer quelle est l'occurrence du motif qui est observée.

On pourrait vouloir concevoir l'environnement de façon à garantir l'absence de symétries. Cependant, il faut noter que tout couple de balises constitue un motif symétrique, car un segment est son propre symétrique par rapport à son centre. En outre, plus l'environnement est grand, plus la probabilité que deux motifs soient similaires (cf. paragraphes suivants) est grande. Avec les hypothèses que nous avons choisies, il est donc illusoire de vouloir construire un système de localisation absolue n'utilisant pas les données odométriques. De plus, ne pas utiliser ces données serait se priver d'une source d'information très discriminante. Pour toutes ces raisons, nous avons choisi de tenir compte de l'odométrie dans le processus d'identification.

Pré-traitement de la carte de l'environnement : Pour mettre en correspondance les segments formés par les couples de balises, on utilise leur longueur. On a donc besoin de pré-traiter les balises de l'environnement de façon à pouvoir identifier rapidement les segments dont la longueur est proche de celle d'un segment donné. Pour cela, on construit un arbre binaire de recherche (ABR) qui contient tous les couples de balises de l'environnement et dont la clef est la longueur du segment formé par les deux balises (cf. figure 2.7). Étant donné cet arbre, et un pourcentage d'erreur ϵ , les segments proches d'un segment de longueur l sont ceux dont la longueur est comprise entre $(1 - \epsilon)l$ et $(1 + \epsilon)l$. On peut donc obtenir cette liste de segments en $O(\log(s) + p)$, où s est le nombre de segments dans l'arbre et p le nombre de segments dans la liste. De plus, si on prend tous les segments possibles, on a $s = \frac{n(n-1)}{2} = O(n^2)$, où n est le nombre de balises dans l'environnement. Finalement, la liste est obtenue en $O(\log(n) + p)$. En pratique, pour un environnement suffisamment grand par rapport à la portée du capteur, certains couples ne pourront pas être visibles, ce qui réduit le coût de la recherche.

D'autre part, on veut aussi pouvoir mettre en correspondance des triplets de balises. Pour cela, on considérera que deux triplets sont identiques s'ils forment le même triangle (à une transformation orthonormale près). Il y a plusieurs manières de caractériser un triangle : trois longueurs, deux longueurs et un angle, une longueur et deux angles, une aire et deux longueurs... La caractérisation que nous allons choisir doit permettre d'utiliser un ABR pour trouver rapidement les triangles proches d'un triangle donné. Pour cela, nous avons choisi de construire un ABR dont la clef est l'aire des triangles et dans lequel nous

FIG. 2.7 – A gauche : 4 balises formant 6 segments. A droite : ABR correspondant.

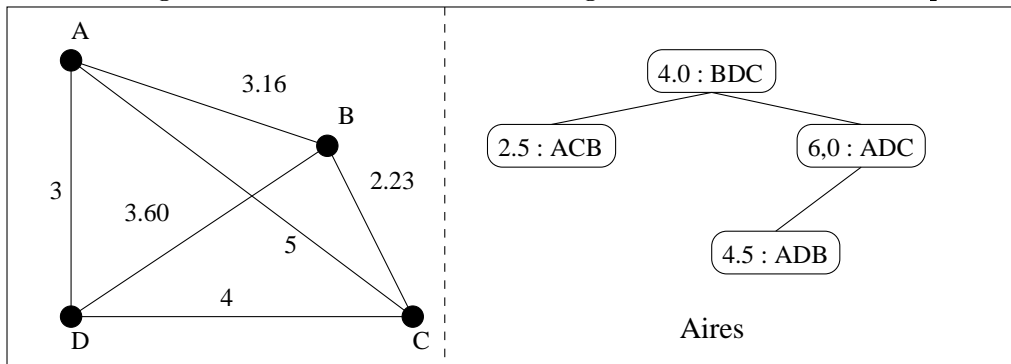


n'insérerons que des triangles directs² (cf. figure 2.8). Ainsi, lorsqu'un triangle (direct) sera recherché dans l'arbre, un ensemble de triangles d'aire proche sera sélectionné. Parmi ces triangles, on cherchera ensuite les triangles dont les trois arêtes sont proches de celles du triangle recherché. C'est ici qu'intervient l'utilisation de triangles directs : si $\mathcal{A} = (A_0A_1A_2)$ et $\mathcal{B} = (B_0B_1B_2)$ sont directs, on dira qu'ils sont équivalents si

$$\begin{aligned} \exists i \in \{0, 1, 2\}, (A_0A_1 = B_iB_{(i+1) \bmod 3} \quad \text{et} \quad A_1A_2 = B_{(i+1) \bmod 3}B_{(i+2) \bmod 3} \\ \text{et} \quad A_2A_0 = B_{(i+2) \bmod 3}B_i) \end{aligned}$$

Intuitivement, cette équation correspond à chercher une rotation de \mathcal{B} telle qu'il se superpose bien à \mathcal{A} . L'utilisation de triangles orientés simplifie donc la recherche de triangles similaires. Si les triangles ne sont pas orientés, cette condition n'est pas nécessaire pour que les triangles soient superposables.

FIG. 2.8 – A gauche : 4 balises formant 4 triangles. A droite : ABR correspondant.



Scores et choix des bons appariements : Les méthodes vues dans le paragraphe précédent permettent de déterminer rapidement la correspondance entre un objet observé et une liste d'appariements potentiels. Il faut désormais trouver une manière robuste de déterminer le meilleur appariement possible. Pour cela, suivant les recommandations de

² (ABC) est direct si $\vec{AB} \wedge \vec{AC} > 0$

[Sch96], on définit un score pour chaque appariement (observation,balise). Ce score correspond au nombre de fois où ce couple a été impliqué dans la mise en correspondance d'un segment ou d'un triangle. Plus ce score est élevé, plus il y a de couples qui sont cohérents avec cet appariement. Intuitivement, cela signifie que le polygone défini par les observations est identique à celui défini par les balises du modèle, à une transformation orthonormale près. Pour choisir les bons appariements, il suffit donc de choisir ceux qui ont le score maximum. Plusieurs stratégies sont alors possibles :

Winner takes all : Lorsqu'un couple (observation,balise) a un score maximum, il est sélectionné, et tous les appariement mettant en jeu cette observation ou cette balise sont éliminés. Méthode simple et rapide, mais peu robuste aux ambiguïtés.

Meilleur appariement croisé : Pour associer une observation à une balise, il faut que cet appariement soit le meilleur possible pour l'observation ET le meilleur possible pour la balise. Méthode un peu plus robuste mais encore trop sensible aux ambiguïtés, en particulier lorsqu'on dispose de peu d'observations.

Mesure d'ambiguïté : Pour chaque appariement, on définit une mesure d'ambiguïté correspondant à l'existence d'un appariement ayant un score proche et mettant en jeu la même observation ou la même balise. On ne garde que les maxima non ambigus. Méthode robuste mais très restrictive.

Pour plus de précision sur les différentes stratégies d'appariement, se rapporter à [Sch96].

L'utilisation d'une mesure d'ambiguïté semble la méthode la plus efficace lorsqu'on privilégie la robustesse. Cependant, en pratique, l'extraction des points d'intérêt, à partir des données télémétriques, ne donne que quelques points (rarement plus de six), il est donc dommage de rejeter des appariements possibles : la section 2.3.3 traitera donc d'une solution robuste et moins restrictive, utilisable sur des petits ensembles tels que ceux que nous utilisons.

Utilisation d'une estimation de la position : L'estimation de la position sera utilisée pour rejeter les faux appariements. En effet, lorsqu'on met en correspondance un couple de balises, on peut en déduire une position du robot. Si celle-ci est trop éloignée de l'estimation courante, on rejettera l'appariement. En pratique, les faux appariements donnant souvent des positions clairement aberrantes, une distance limite de 2 mètres (par rapport aux 10 cm de précision du télémètre laser) se révèle suffisante.

2.3.3 Amélioration par recherche de la clique maximale

La méthode présentée dans cette section est inspirée de [BNRDW00]. En utilisant la théorie des graphes, on peut remplacer la ligne 11 de la figure 2.9 par une méthode moins restrictive et mieux adaptée au petit nombre de points à mettre en correspondance.

Définition du graphe de correspondance (cf. figure 2.10) : A un instant donné, on dispose de balises $\{B_i\}_{i \in [1..n]}$ et d'observations $\{O_i\}_{i \in [1..p]}$ parmi lesquelles se trouvent les images de certaines balises. On définit un graphe \mathcal{G} dont les noeuds sont les couples $\{(B_i, O_j) | i \in [1..n], j \in [1..p]\}$. Les arêtes seront ajoutées lors des phases de mise en correspondance des segments et des triangles.

FIG. 2.9 – Algorithme de la mise en correspondance

```

1  Construction des ABR pour les longueurs et pour les aires.
2  Pour tout couple  $\mathcal{O}$  d'observations,
3      Pour tout couple  $\mathcal{B}$  de balises proche de  $\mathcal{O}$ ,
4          Si on connaît une estimation de la position,
5              Ajouter 1 au score des deux appariements générés.
6          Sinon
7              Ajouter 1 au score des quatre appariements possibles
              (Dans ce cas, deux faux appariements voient leur score augmenter).
8  Pour tout triplet  $\mathcal{T}_O$  d'observations,
9      Pour tout triplet  $\mathcal{T}_B$  de balises proche de  $\mathcal{T}_O$ ,
10         Ajouter 1 au score des trois appariements générés.
11 Retourner les appariements de score maximal dont l'ambiguïté
    est suffisamment petite.

```

- Lorsqu'on peut mettre en correspondance les segments (O_1, O_2) et (B_1, B_2) , on ajoute une arête reliant les noeuds (B_1, O_1) à (B_2, O_2) .
- De même, lorsqu'un triangle (O_1, O_2, O_3) est mis en correspondance avec un triangle (B_1, B_2, B_3) , on ajoute trois arêtes : $(B_1, O_1) \leftrightarrow (B_2, O_2)$, $(B_2, O_2) \leftrightarrow (B_3, O_3)$ et $(B_3, O_3) \leftrightarrow (B_1, O_1)$.

Intuitivement, les appariements qui sont liés par une arête sont cohérents l'un avec l'autre. Le plus grand ensemble d'appariements cohérents les uns avec les autres sera donc un sous-graphe de \mathcal{G} tel qu'il existe une arête entre tout couple de noeuds : il s'agit d'une clique maximum. Malheureusement, dans le cas général, la recherche d'une clique maximum est un problème NP-complet, donc difficilement compatible avec le temps réel dont a besoin un robot. Cependant, plusieurs aspects particuliers du problème de localisation rendent cette recherche abordable :

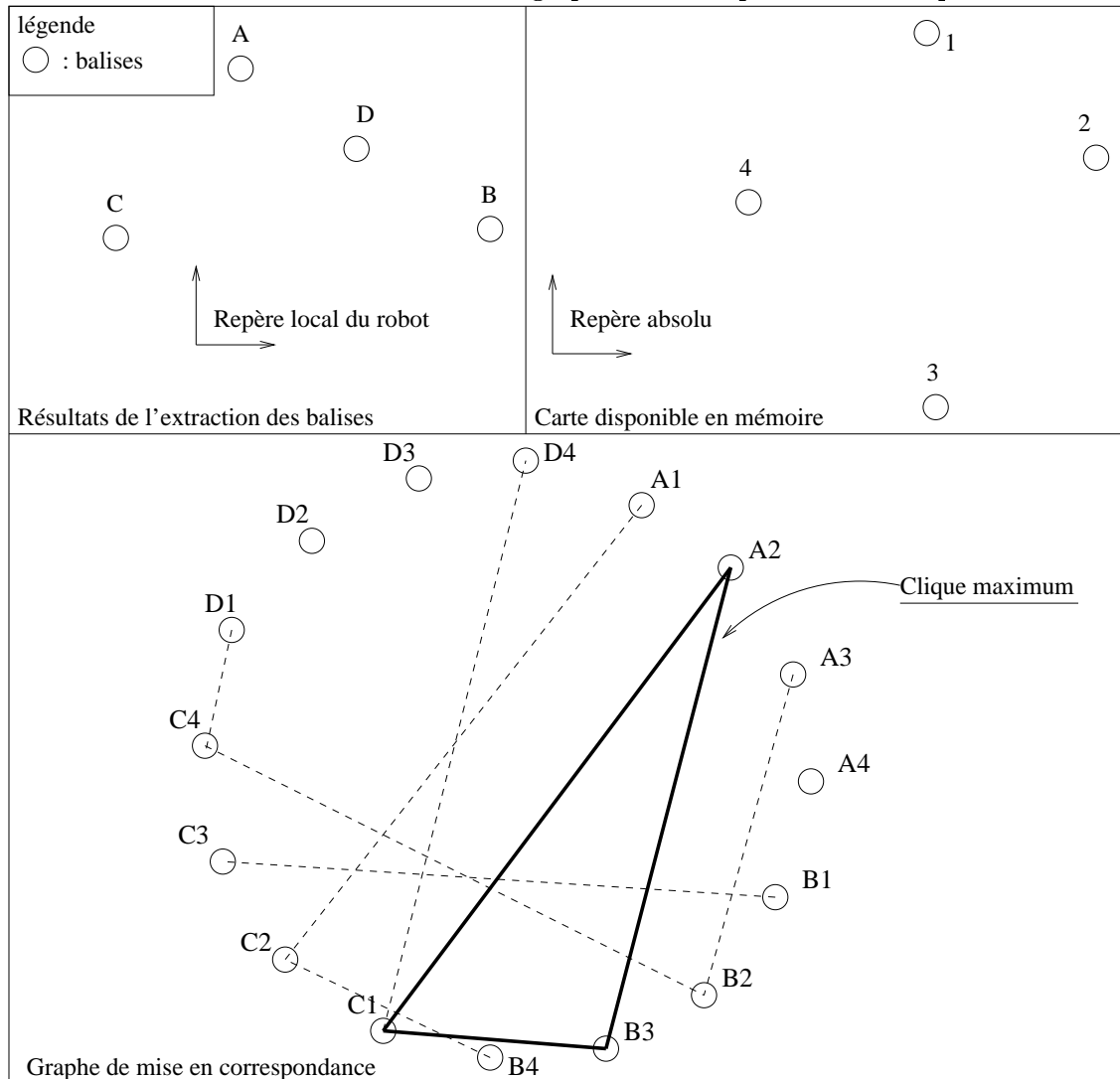
- Tout d'abord, si la phase de mise en correspondance des segments et des triangles est suffisamment discriminante, le graphe possède peu d'arêtes.
- Ensuite, comme on l'a déjà évoqué, le nombre d'observations est généralement assez faible, ce qui limite fortement la taille du graphe.
- Il existe des optimisations qui permettent de réduire la taille du graphe : par exemple, si on recherche une clique de taille n , tous les noeuds de cette clique devront avoir un degré supérieur à $n - 1$, il n'est donc pas nécessaire de considérer les noeuds de degré inférieur.

Avantage : La recherche du plus gros sous-ensemble d'appariements cohérents entre eux permet de répondre à deux problèmes :

1. trouver les appariements corrects.
2. rejeter les faux appariements et détecter les objets qui ont été pris pour des balises.

Si on ne dispose pas d'estimation de la position, seul le deuxième point pourra être résolu dans le cas général : il y aura autant de cliques maximums que de répétitions du motif observé dans la carte des balises.

FIG. 2.10 – Processus d'identification : graphe de correspondance et clique maximum



Inconvénient : Pour déterminer si une clique de taille p existe dans un graphe de taille n , la seule solution exacte consiste à essayer tous les sous-ensembles de noeuds du graphe de degré supérieur à $p - 1$ et de taille p . Il y a donc C_n^p sous-ensembles à tester a priori. La vérification d'un ensemble est quadratique (pour chaque noeud, on vérifie sa connexion à tous les autres noeuds). La complexité au pire est donc de $O(\binom{n}{p} \cdot p^2)$. En pratique, il n'y a que très peu de noeuds dont le degré est suffisant, il y a donc rarement plus d'une clique à tester. Dans ce cas, la complexité est seulement quadratique, ce qui est tout à fait raisonnable, d'autant plus que la clique maximum contient rarement plus de six noeuds. Pour toutes ces raisons, il est possible de traiter ce problème en temps réel : moins de 10 ms pour tout le processus, de l'extraction à la mise en correspondance.

FIG. 2.11 – Algorithme d'extraction d'une clique maximum

Cet algorithme est inspiré des méthodes introduites dans [Ski97]. On notera $\#E$ le nombre d'éléments d'un ensemble E .

```

#
# Recherche d'une clique maximale  $C_m$  (maximum local)
# (sous-graphe complet tel que si on ajoute un noeud, on perd la complétude)
#
  Trier l'ensemble des noeuds selon leur degré
   $C_m \leftarrow \{\text{Noeud de plus haut degré}\}$ 
  Pour chaque noeud  $N$ , par degré décroissant
    S'il existe une arête reliant  $N$  à tous les noeuds de  $C_m$ 
       $C_m \leftarrow C_m \cup N$ 
#
# Recherche d'une clique maximum  $\mathcal{C}$  (maximum global)
#
  Posons  $d_m$  le plus haut degré du graphe.
   $\mathcal{C} \leftarrow C_m$ 
  Pour  $i$  de  $\#C_m + 1$  à  $d_m + 1$ 
    S'il existe une clique de taille  $i$ 
       $\mathcal{C} \leftarrow$  une clique de taille  $i$ .
  Sinon
    Retourner  $\mathcal{C}$ 

```

Algorithme : L'algorithme de la figure 2.11 n'assure pas de trouver la clique maximum, mais **une** clique maximum. En effet, si la carte des balises possède des motifs répétés ou des symétries, il y aura plusieurs cliques de taille maximum. Sans estimation de la position du robot, il est alors impossible de déterminer la position réelle. Pour relativiser le problème, il faut remarquer que les méthodes précédentes ne pourraient pas non plus déterminer la position exacte dans ces conditions. Plusieurs solutions sont envisageables :

- Toujours avoir à disposition une estimation de la position : même une estimation très grossière est suffisante pour résoudre ce problème.
- Déterminer un ensemble de positions possibles et essayer de le réduire au fur et à mesure des déplacements du robot.

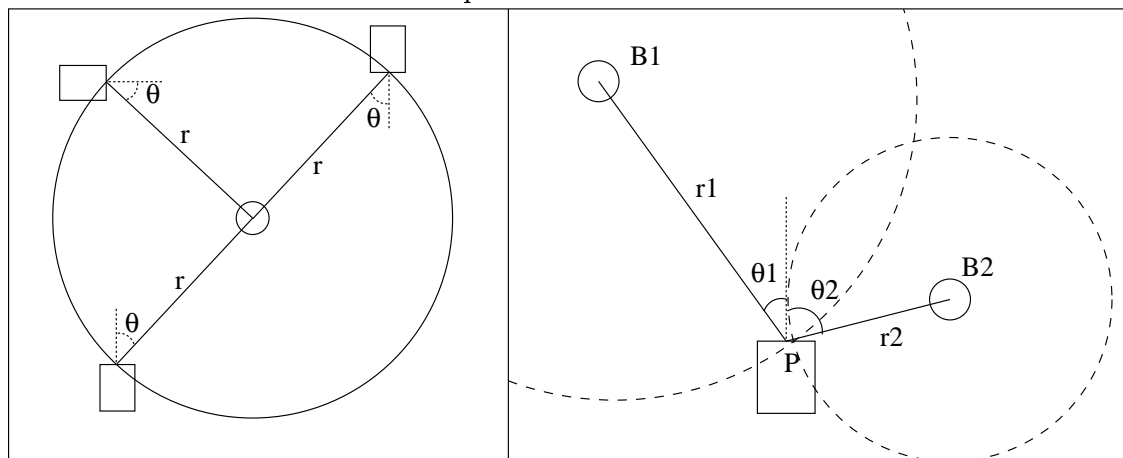
Malgré ces quelques amorces de solutions, le problème reste ouvert. Toutefois, en pratique, on dispose toujours d'une estimation de la position, ce qui rend la méthode de la clique

maximum très robuste et raisonnablement efficace.

2.4 Calcul de la position

Calcul avec deux balises : La phase d'identification nous fournit un ensemble de relevés télémétriques sur des points réels de l'espace de travail : il ne reste plus qu'à en déduire la position du robot. La première question qui se pose alors est le nombre minimum de balises nécessaires. Si on dispose d'une seule balise, relevée à une distance r , toute position sur le cercle de rayon r centrée sur la balise permet d'observer la balise avec le bon relevé (cf. figure 2.12). Une seule balise n'est donc pas suffisante. A contrario, si on dispose de deux balises $B_1 : (r_1, \theta_1)$ et $B_2 : (r_2, \theta_2)$, la position doit appartenir au cercle de centre B_1 , de rayon r_1 et au cercle de centre B_2 , de rayon r_2 . Il y a donc en général deux solutions dont une seule produit les bons angles.

FIG. 2.12 – Calcul de position et nombre minimum de balises



- A gauche : avec une seule balise, la position du robot n'est pas uniquement déterminée.
- A droite : avec deux balises, la solution est unique.

Utilisation de nombres complexes : On peut obtenir une expression simple de la position et de l'orientation du robot en utilisant une représentation en nombres complexes des diverses grandeurs (inspiré de [BG94]) : à la position $P = (x, y)$, on associe le complexe $p = x + iy$, à B_1 on associe le complexe b_1 et à B_2 , b_2 . De même, on exprime les relevés : $r_1 e^{i\theta_1}$ et $r_2 e^{i\theta_2}$. Avec ces éléments, on a :

$$\begin{aligned} p + r_1 e^{i\theta_1} e^{i\theta} &= b_1 \\ p + r_2 e^{i\theta_2} e^{i\theta} &= b_2 \end{aligned}$$

La multiplication des relevés par $e^{i\theta}$ exprime l'influence de l'orientation du robot sur l'angle des relevés.

De ces équation, on déduit :

$$e^{i\theta} = \frac{b_1 - b_2}{r_1 e^{i\theta_1} - r_2 e^{i\theta_2}}$$

$$p = b_1 - r_1 e^{i\theta_1} e^{i\theta}$$

On a donc immédiatement la position du robot et son orientation, sans changement de repère. En pratique, la norme de $d = \frac{b_1 - b_2}{r_1 e^{i\theta_1} - r_2 e^{i\theta_2}}$ n'est jamais exactement égale à 1 : on a $d = r e^{i\theta}$ et $r \approx 1$. Si l'identification est faite en utilisant la méthode naïve, la valeur de r peut être utilisée de façon à éliminer les fausses correspondances : r sera alors très éloigné de 1.

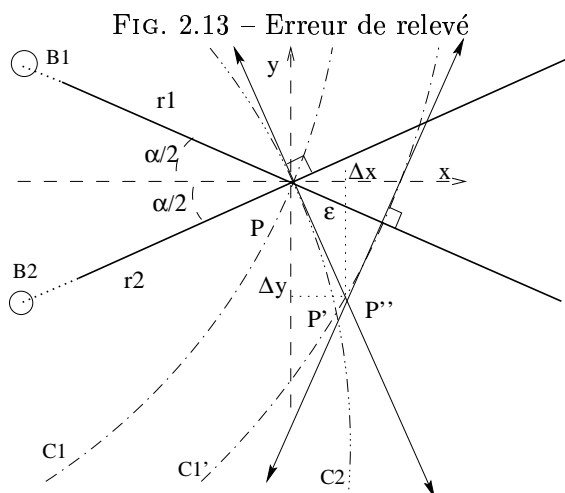
Avec plus de deux balises : Finalement, lorsqu'on détecte plus de deux balises, on les traite deux par deux et on fait la moyenne des positions obtenues. Pour minimiser l'influence des erreurs de mesures (section 2.4), les couples de balises sont choisis de façon à ce que $\theta_1 + \theta_2$ soit le plus proche de $\frac{\pi}{2}$. De cette façon, on arrive à obtenir une erreur inférieure à 10 centimètres en position et inférieure à 5 degrés en orientation. Il faut noter que le fait de faire la moyenne des positions est une approximation qui suppose implicitement que l'incertitude sur la position calculée est indépendante du couple de balises considérées. Nous avons envisagé un filtrage plus adapté (cf. section 2.5), mais qu'il a fallu l'abandonner pour des problèmes de temps réel.

Évaluation de la précision des mesures : Il est très difficile d'évaluer avec précision l'erreur de la localisation. Pour comparer la position calculée à partir des données laser à la position réelle, il faut faire la triangulation "à la main" en mesurant la distance de chaque balise au capteur avec un décimètre. Même si on peut espérer une précision de l'ordre du centimètre sur la mesure de distance, la précision sur la position du centre d'une balise est de l'ordre du centimètre et celle sur la position du point de référence du capteur est de l'ordre de deux centimètres. Finalement, on ne peut pas espérer une erreur inférieure à 4 centimètres sur la mesure des coordonnées de référence.

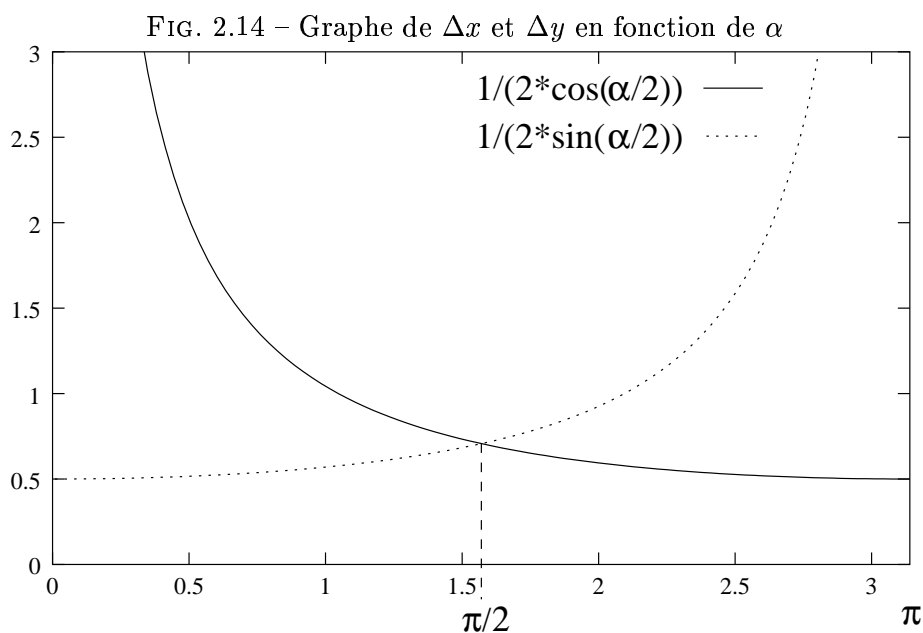
Pour ce qui est de l'erreur sur l'orientation du robot, nous ne disposons pas d'instrument donnant une mesure avec une précision inférieure à 2 degrés. Les mesures manuelles sont donc trop grossières pour évaluer la précision réelle du dispositif de localisation. Cependant, ces mesures donnent tout de même une borne supérieure à l'erreur de localisation, borne qui correspond à une précision tout à fait correcte compte tenu des dimensions du robot.

Influence des erreurs de mesure : Supposons que pour une position P du robot, les relevés vers deux balises soient r_1 et r_2 . On cherche à déterminer quelle serait l'influence sur P d'une erreur de mesure sur r_1 : si la distance mesurée à la balise B_1 est $r_1 + \epsilon$, alors le position P' est donnée à la place de P . Sans perte de généralité, on peut supposer que le repère absolu est tel que l'axe x soit la première bissectrice de l'angle $(\overrightarrow{PB_1}, \overrightarrow{PB_2})$ et y la seconde (cf. figure 2.13). Par ailleurs, en première approximation, on peut assimiler les cercles C'_1 et C_2 à leur tangente et assimiler P'' (l'intersection de ces tangentes) à P' . En notant $\overrightarrow{PP''} = (\Delta x, \Delta y)$, on obtient :

$$\begin{aligned} \Delta x &= \frac{\epsilon}{2 \sin(\alpha/2)} \\ \Delta y &= \frac{\epsilon}{2 \cos(\alpha/2)} \end{aligned}$$



On constate sur la figure 2.14 que l'influence des erreurs de mesure va être minimale pour



$\alpha = \pi/2$. Lorsqu'on choisit deux balises pour calculer la position du robot, il est donc préférable de les choisir avec un angle alpha aussi proche de $\pi/2$ que possible.

2.5 Intégration des données de localisation dans un filtre de Kalman

2.5.1 Définition

Pour se localiser, le Cycab dispose de trois outils :

- Des encodeurs incrémentaux sur chaque roue : l'odométrie.
- Un télémètre laser Sick LMS-219.
- Un gyroscope.

L'intégration des données de tous ces capteurs a été faite en utilisant un filtre de Kalman étendu comme décrit dans [Cro89] ou [Lam96]. Ce filtre permet d'estimer l'état d'un système dynamique pouvant être modélisé par un processus de Markov discret :

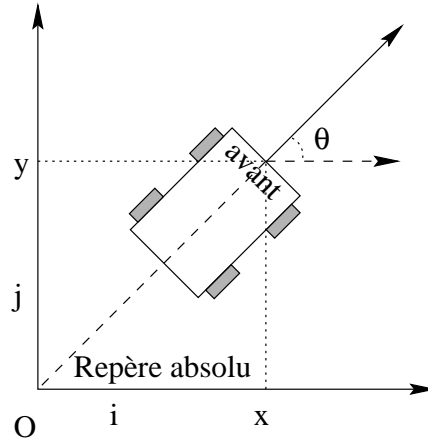
$$X_k = f(X_{k-1}) + Q_k$$

Où X_k représente l'état du système à la k -ième itération et Q_k est un bruit blanc³ dont on suppose la covariance connue. En outre, ce système dispose de capteurs qui donnent des observations Z_k dans l'état X_k :

$$Z_k = h(X_k) + R_k$$

Où R_k , comme Q_k , est un bruit blanc de covariance connue.

FIG. 2.15 – Modèle simplifié du Cycab



État du système : L'état X_k qui sera estimé doit permettre de bien décrire le système. Toutefois, plus la dimension de cet état est grande, plus les calculs nécessaires à l'exécution du filtre seront complexes (les matrices utilisées ont la dimension du vecteur d'état). L'état statique du Cycab peut être décrit par trois paramètres : sa position (x, y) et son orientation θ . Pour tenir compte de la dynamique du véhicule, on ajoute la vitesse linéaire $v = \|\dot{x}^2 + \dot{y}^2\|$ et sa vitesse de rotation $\dot{\theta} = \frac{d\theta}{dt}$. Ce qui donne :

$$X_k = \begin{bmatrix} x_k & y_k & v_k & \theta_k & \dot{\theta}_k \end{bmatrix}^T$$

³Bruit gaussien de moyenne nulle.

Fonction de transition : La fonction f permet de prévoir l'évolution du système d'un instant à l'autre. En pratique, on choisit :

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ v_{k+1} \\ \theta_{k+1} \\ \dot{\theta}_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + v_k \cos(\theta_k) dt \\ y_k + v_k \sin(\theta_k) dt \\ v_k \\ \theta_k + \dot{\theta}_k dt \\ \dot{\theta}_k \end{bmatrix} \quad (2.2)$$

Fonction d'observation : Nous disposons de trois capteurs, le capteur laser nous donne (x, y, θ) , le gyroscope nous donne $\dot{\theta}$ et il est possible de mesurer la vitesse de chaque roue pour en déduire la vitesse du Cycab. h pourrait donc se ramener à l'identité. Cependant, les problèmes de différences de période et de latence demandent un traitement particulier des mesures, qui sera discuté dans la section suivante. Comme on l'a déjà mentionné, la donnée laser résulte d'une moyenne de positions calculées à partir des balises identifiées. Pour éviter cette approximation, il serait préférable d'intégrer indépendamment les relevés sur chaque balise dans le filtre de Kalman. Cependant, pour satisfaire les contraintes temps-réel de notre plateforme, il n'était pas possible de faire plusieurs fusions. Nous avons donc dû nous contenter de cette position moyenne.

2.5.2 Période d'échantillonnage et latence

Une particularité du Cycab est la distribution de son architecture : d'une part, le contrôleur temps-réel du véhicule est une station VxWorks sur laquelle est connecté un gyroscope. D'autre part, le véhicule dispose d'un PC portable qui contrôle le capteur Sick. Ce PC est relié à la station VxWorks par ethernet et au Sick par une ligne série RS232 (limitée à 38kbds). Sur la station VxWorks, la période du contrôleur temps-réel est de 100ms. Sur le PC, pour cause de contraintes techniques, les données sont accessibles avec environ 300ms de retard.

La donnée laser Z_{k-3}^L est donc disponible en X_k uniquement lorsque k est divisible par 3. La donnée du gyroscope Z_k^G , elle, est accessible en X_k quel que soit k . Le principal problème vient de la latence associée à Z^L car elle contredit l'hypothèse markovienne à la base du filtre de Kalman. En effet, dans un système markovien, l'état courant ne dépend que de l'état précédent :

$$P(X_k | X_{k-1} \cdots X_0) = P(X_k | X_{k-1}) \quad (2.3)$$

Dans le cas où k est divisible par 3, l'équation 2.3 devient :

$$P(X_k | X_{k-1} \cdots X_0) = P(X_k | X_{k-1} \cdots X_{k-4}) \quad (2.4)$$

Le système ne peut donc pas être modélisé par un processus markovien. Par conséquent, il ne peut pas être traité immédiatement par un filtre de Kalman. Cependant, en mémorisant les 4 derniers états, il est possible de se ramener à un processus markovien. En pratique, on applique deux comportements :

1. Tant que l'on ne dispose pas de données laser, on utilise la donnée du gyroscope et la vitesse mesurée pour mettre à jour le filtre de façon standard. Dans la suite, le

mot “odométrie” désignera les données issues de ce filtrage. On a :

$$h = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

2. Lorsqu’une donnée laser Z_{k-3}^L est disponible, on réinitialise le filtre de Kalman à l’état X_{k-4} et on réévalue X_{k-3} en intégrant l’observation laser, le gyroscope et la vitesse. La fonction h est alors égale à l’identité. Ensuite, on utilise les données mémorisées (vitesses et données du gyroscope) pour estimer X_{k-2} , X_{k-1} et X_k .

De cette façon, l’hypothèse de Markov est bien vérifiée à chaque pas. De plus, la donnée laser est bien utilisée au moment où elle apporte une information.

2.5.3 Gestion de l’absence temporaire de localisation

L’utilisation de clique maximale rend la localisation laser très robuste et très fiable. Cependant, il existe deux principaux cas où la localisation ne peut être assurée :

- Perte de contact temporaire : les balises sensées être visibles sont momentanément masquées par un objet en mouvement (piéton, véhicule), ou bien un cahot du véhicule a rendu la détection impossible.
- Perte de contact longue : le véhicule sort de la zone où la localisation est possible (conduite manuelle), aucune balise n’est visible.

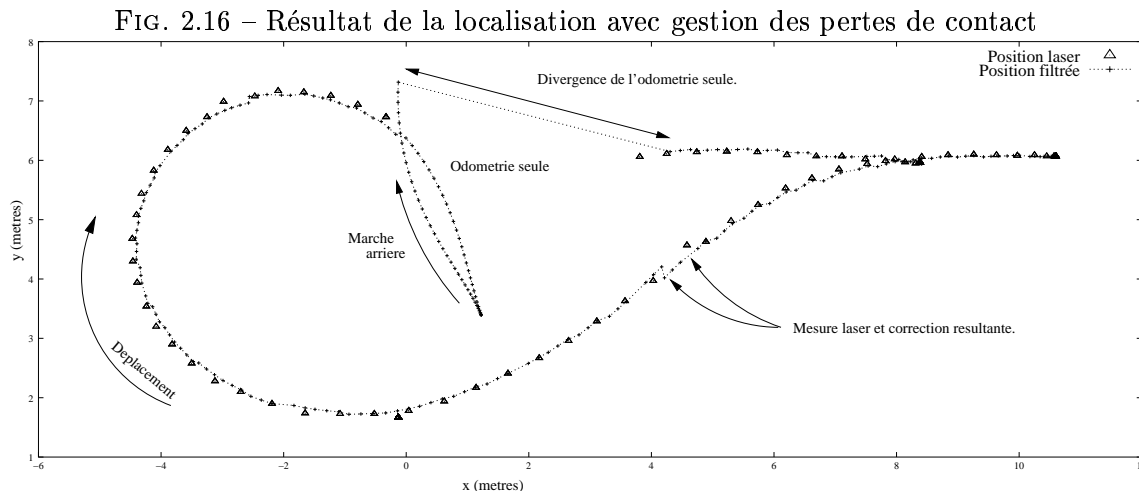
Dans ce cas, pendant toute la durée de la perte de contact, seule l’odométrie est utilisée pour maintenir une estimation de la position. Si la perte de contact n’est pas trop longue, on peut considérer que cette estimation reste correcte (5 à 10 secondes). Un problème se pose au moment où des balises sont de nouveau détectées. En effet, on a vu en 2.3 que lorsque le motif créé par les balises détectées se retrouve de manière non unique dans la carte de l’environnement, l’identification ne peut se faire qu’en utilisant une estimation de la position. Dans le cas d’une perte courte, la précision de l’estimation fournie par l’odométrie sera suffisante pour assurer une localisation fiable et robuste.

Dans le cas d’une perte de contact longue, au moment où on peut détecter de nouveau des balises, l’odométrie peut avoir accumulé trop d’erreur pour être utilisable. Il faut donc envisager un traitement particulier. Dans le cadre de nos expérimentations, nous avons un environnement suffisamment petit pour pouvoir considérer la position donnée par le laser comme la position réelle lorsqu’au moins trois balises avait été identifiées. Cette technique n’est pas généralisable à un environnement plus grand, mais dans notre cas, elle s’est révélée suffisamment fiable.

Dans le cadre d’une mission autonome, en cas de perte longue, il est envisageable de suspendre momentanément la mission pour exécuter une manoeuvre de localisation. Comment gérer ce mécanisme de suspension, quelle manoeuvre effectuer, quel critère utiliser pour considérer la relocalisation comme réussie,... ? Autant de questions qui n’ont pas pu être traitées dans le cadre de ce travail, mais qui présentent une difficulté et un intérêt certains.

2.6 Conclusion sur la localisation

Dans ce chapitre, nous avons présenté une méthodologie de localisation intégrant un capteur télémétrique laser, des amers artificiels et des données odométriques. Les trois



objectifs de la méthode ont été la précision, la robustesse et la fiabilité :

Précision : La précision de la méthode est issue de la précision intrinsèque du capteur laser, du traitement efficace des données laser brutes et du choix judicieux des couples de balises à utiliser pour déterminer la position.

Robustesse : L'utilisation d'invariants géométriques et d'un filtre de Kalman assurent la robustesse de la méthode face aux erreurs de mesure inévitables.

Fiabilité : La notion de clique maximum associée à une estimation efficace de la position par odométrie entre deux mesures télémétriques permet de garantir un maximum de fiabilité de l'estimation de position.

Cependant, l'initialisation ou la relocalisation après une perte de contact longue restent problématiques, surtout si l'environnement possède de nombreuses symétries. A l'initialisation, une intervention humaine peut être nécessaire pour résoudre cette ambiguïté. Après une perte de contact longue, la méthode permettant de lever l'ambiguïté reste à définir : elle dépendra certainement du type d'application envisagé. La meilleure solution reste encore de concevoir l'environnement de façon à ce qu'il n'y ait pas de perte de contact longue : c'est le problème qui sera abordé dans le chapitre suivant.

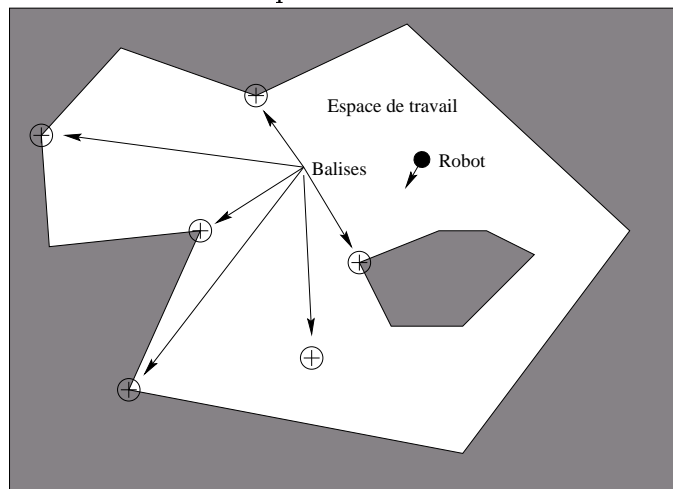
Chapitre 3

Placement des amers dans l'espace de travail

3.1 Problématique et modèles

Lorsque le thème de la localisation avec balises est abordé dans la littérature, on suppose généralement que ces balises sont déjà placées dans l'environnement et qu'il faut les utiliser au mieux. Cette approche convient tout à fait à l'utilisation de balises naturelles (angles entre deux murs, portes...). Cependant, si on utilise des balises artificielles, il est nécessaire de choisir où elles vont être placées dans l'espace de travail. De ce choix dépend l'existence de parties de l'espace de travail où la localisation sera impossible. De ce choix, dépend aussi la qualité de la localisation et sa fiabilité. Le choix du positionnement des balises dans l'espace de travail sera désigné sous le nom de “**problème du placement**”.

FIG. 3.1 – Espace de travail et balises



Dans ce chapitre, nous considérerons que l'espace de travail du robot est un polygone contenant éventuellement des trous polygonaux. Les balises seront supposées ponctuelles. Le robot sera assimilé à un point de coordonnées (x, y) doté d'une orientation θ . On considèrera de plus que le capteur télémétrique dispose d'une ouverture angulaire α

inférieure ou égale à π radians.

Comme dans le chapitre précédent, nous supposons que les balises ne sont pas distinguables, il faudra donc en voir au moins deux pour pouvoir se localiser (cf. section 2.4). Cependant, les méthodes d'optimisation de la section 3.4 seront généralisables à un capteur d'ouverture quelconque et à une localisation demandant un nombre quelconque¹ de balises.

Classification des problèmes de placement : Plusieurs paramètres doivent être pris en compte lorsqu'on doit aborder un problème de placement.

- Portée du capteur [en mètres] : on la considérera infinie si, abstraction faite des problèmes d'obstruction, tout point de l'espace de travail peut être visible de tout autre point avec une orientation convenable.
- Ouverture angulaire [en radian] : on la comparera à π . Si elle est inférieure à π , quel que soit le placement des balises, il existera toujours des zones où la localisation sera impossible.
- Présence d'obstacles et convexité de l'espace de travail. Si l'environnement est convexe sans obstacles, le problème du placement sera grandement simplifié. Dans le cas contraire, il faudra être capable de gérer les problèmes d'obstructions. On considérera qu'un environnement infini (sans bordures) est convexe.
- Placement des balises : selon que l'on autorise ou non le placement de balises dans l'espace de travail (et non uniquement sur la bordure), le problème se traitera de façon différente. Tout d'abord, on passe d'un problème de dimension 1 à un problème de dimension 2. Ensuite, si on accepte les balises dans l'espace de travail, il peut être nécessaire de les considérer comme des obstacles.

En fonction de ces paramètres, le traitement du problème du placement sera différent. Par exemple, avec une portée infinie et une ouverture quelconque, quel que soit l'espace de travail, on montrera analytiquement dans la section 3.2 qu'il suffit de placer les balises sur la bordure de l'espace de travail pour pouvoir se localiser. Le tableau 3.1 dresse un inventaire des problèmes qui seront abordés dans ce chapitre.

Section	Portée	Ouverture	Environnement	Obstacles	Placement	Solution
3.2	infinie	$\alpha < \pi$	quelconque	oui	bordure	analytique
3.3	finie	π	infini	non	dans l'env.	analytique
3.4	finie	quelconque	quelconque	oui	bordure ²	numérique

TAB. 3.1 – Sous-problèmes traités dans ce chapitre

3.2 Solution analytique pour une portée infinie

Dans cette section, on considère un cas particulier du problème du placement : D'une part, on se place dans un environnement polygonal. D'autre part on utilise un capteur dont la portée est infinie et l'ouverture strictement inférieure à π . Une telle situation peut

¹Supérieur à zéro.

²Choix arbitraire pour simplifier l'implémentation.

être obtenue, par exemple, en utilisant une caméra et un dispositif de reconstruction 3D³ en intérieur.

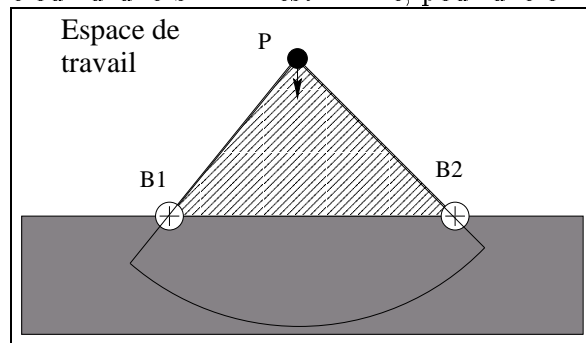
De plus, nous avons choisi de ne placer des amers que sur la bordure de l'espace de travail, et ce pour deux raisons :

- un tel placement permet de diminuer la dimension du problème.
- en plaçant les balises sur la bordure, on trouve une solution garantissant la localisation en tout point, il n'est donc pas nécessaire d'ajouter des balises dans l'espace de travail.

3.2.1 Définition d'une zone interdite

Avec un capteur dont l'ouverture est inférieure à π , on peut constater que, quel que soit le nombre de balises, on ne peut pas être sûr d'en voir toujours une. Sur la figure 3.2, pour une orientation du robot perpendiculaire au bord de l'environnement, aucune balise n'est visible lorsque le robot est dans la zone hachurée. Pour une orientation donnée, la position du robot qui fait passer les bords du cône de sensibilité par B_1 et B_2 définit un triangle où aucune balise n'est visible.

FIG. 3.2 – Zone où aucune balise n'est visible, pour une orientation donnée



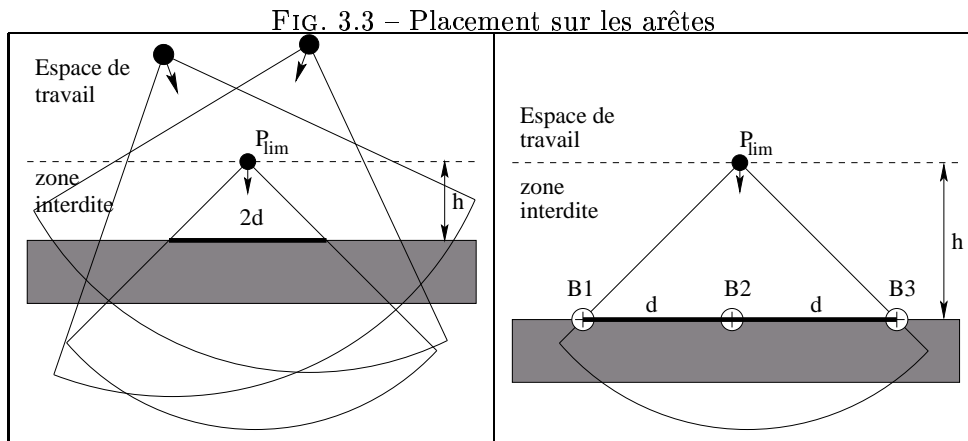
Pour résoudre ce problème, on définit une zone interdite sur les bords de l'espace de travail. Si le robot ne peut s'approcher à une distance inférieure à h du bord, il est possible de placer les balises de façon à en voir toujours deux. Pour cela, trois cas doivent être étudiés : les arêtes de l'espace de travail, les sommets convexes et les sommets concaves.

Arêtes de l'espace de travail : Sur les arêtes, assez loin des sommets, on obtient la situation illustrée par la figure 3.3. La position P_{lim} , à une distance h du bord, orientée perpendiculairement au bord, définit la plus petite section visible de la bordure (en gras). Si on note d la moitié de la longueur de cette section, et si on place des balises espacées de d sur le bord, on sera assuré de voir toujours au moins deux balises. En pratique, on a :

$$d = h \tan(\alpha/2) \quad (3.1)$$

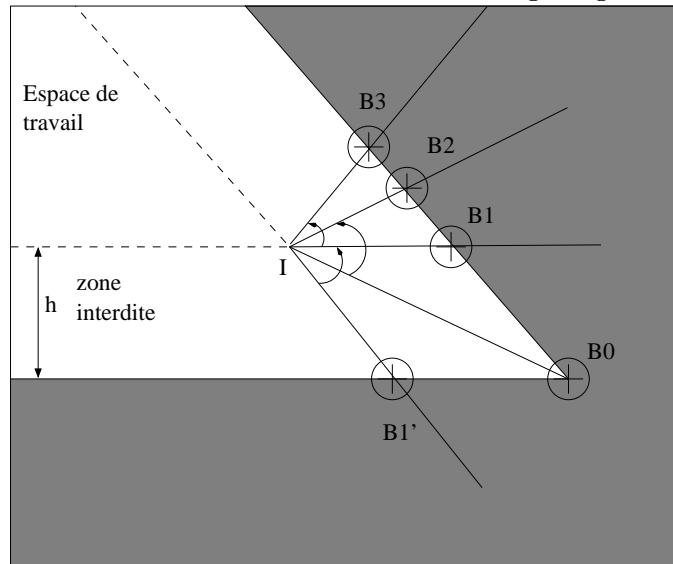
A contrario, si les balises sont plus éloignées, on peut trouver une position et une orientation qui permettent de voir moins de deux balises. Il s'agit donc bien de la limite supérieure de l'espacement entre deux balises.

³Pour obtenir l'équivalent d'un capteur télémétrique.



Sommets concaves : On peut montrer qu'il suffit de placer une balise sur le sommet, puis tous les $h \tan(\alpha/2)$ sur chaque arête incidente.

FIG. 3.4 – Placement des balises dans un angle aigu convexe



Sommets convexes : Dans ce cas, on cherche à optimiser le nombre de balises utilisées en choisissant astucieusement leur placement. Ce placement est illustré sur la figure 3.4. On commence par placer une balise B_0 sur le sommet et le robot en I (c'est de cette position que la section visible de la bordure est la plus petite) orienté vers B_0 . On place alors deux balises aux limites du cône de sensibilité : B_1 et B_1' . De cette façon, on est assuré qu'une des deux sera visible si B_0 l'est. En restant en I, pour $j \geq 0$, on oriente ensuite le robot de façon à ce que B_j soit à la limite du cône de sensibilité et que B_{j+1} soit visible. On place alors B_{j+2} sur l'autre limite du cône. On place des balises ainsi jusqu'à ce que l'orientation nécessaire dépasse la perpendiculaire au bord de l'espace de travail. A partir de ce moment, on les place tous les $h \tan(\alpha/2)$. On traite de la même façon l'arête

sur laquelle on a placé B'_1 .

De cette façon, lorsqu'une balise disparaît du champ de vision, une autre la remplace. Pour étendre ce raisonnement au reste de l'espace, on utilise la même remarque que pour le placement sur les arêtes : si deux balises sont toujours visibles au point I, alors, comme la section visible de la bordure à partir d'autres points est plus grande que celle visible à partir du point I, on est sûr qu'il y aura toujours au moins deux balises visibles.

Possibilités de généralisation : Dans le cas d'un robot doté d'un capteur d'ouverture $\alpha = \pi$ et de portée finie p évoluant sur un réseau de routes, on utilise des méthodes géométriques et analytiques exactes pour montrer le résultat suivant :

Si la portée est supérieure à deux fois la largeur de la route sur un tronçon de route rectiligne, il suffit de placer des balises tous les $p/2$ sur chaque bord de la route pour garantir la localisation. Dans le cas de virage en angle droit ou obtus, ou de croisement en "T", si on se ménage une zone interdite suffisamment grande et avec quelques balises de plus, il est toujours possible de se localiser.

Les démonstrations de ces résultats sont très calculatoires et ne peuvent être généralisées à des environnements plus compliqués, elles ne présentent donc pas de réel intérêt théorique et ne seront pas développées dans ce travail.

Commentaires : Nous avons présenté un algorithme permettant de placer des amers de façon quasi optimale dans un environnement polygonal. Le placement des amers sur les sommets, ainsi qu'une condition pour le placement sur les arêtes ont été définis. L'intérêt de cette méthode vient de ce qu'elle montre que dans certain cas, un raisonnement géométrique permet d'obtenir une solution optimale exacte.

Cependant, il faut remarquer que les hypothèses faites sur le capteur (ouverture inférieure à π , portée infinie) soulèvent des problèmes particuliers.

- En premier lieu, considérer que le capteur a une portée infinie ne donne une solution que pour un sous-ensemble des problèmes de placement : l'environnement doit être suffisamment petit par rapport à la portée du capteur, ce qui est très limitant.
- En second lieu, cette section met en évidence les problèmes liés à l'utilisation d'un capteur télémétrique d'ouverture inférieure à π . Avec un capteur d'ouverture $\alpha = \pi/3$ et une zone interdite de $h = 1$ mètre, il faut placer une balise tous les 57 centimètres. Deux problèmes se posent alors : tout d'abord, si l'environnement est relativement grand, il faut énormément de balises (≈ 1500 sur le parking de l'INRIA) ; ensuite plus les balises sont proches l'une de l'autre, plus le risque de faux appariement est grand et moins la localisation est fiable. Par ailleurs, si le robot s'approche des bords de l'espace de travail, il y a peu de chances qu'il puisse voir des balises et donc se localiser.

Pour ces raisons, la section suivante utilisera une ouverture de π radians.

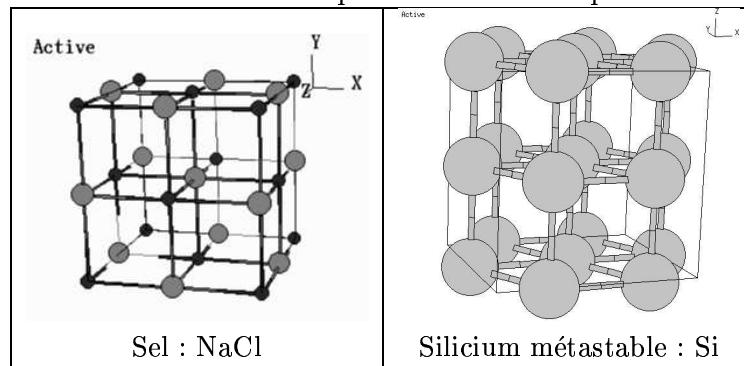
3.3 Étude de maillages de l'espace de travail

Dans cette section, on considérera que le robot dispose d'un capteur d'ouverture angulaire π et de portée p finie. L'objectif recherché est de placer des balises dans un espace sans obstacles et très grand par rapport aux dimensions du robot et à la portée de ses capteurs.

Pour ne pas être gêné par des effets de bord, on considérera que l'espace de travail est infini. Cette situation est difficilement applicable à un robot automobile en environnement urbain, mais elle peut s'appliquer pour un aéronef autonome utilisant des repères au sol, ou bien pour un navire autonome se repérant par rapport à des amers artificiels.

L'idée de base est que, s'il est possible de placer les balises de façon à garantir la localisation dans un polygone simple (rectangle, triangle, hexagone), en pavant l'espace de travail avec ce polygone, la localisation sera possible en tout point de l'espace. Ce principe de maillage de l'espace de travail peut évoquer l'étude de la structure des cristaux parfaits que l'on peut voir en cristallographie. En effet, les cristaux parfaits (sel, silicium...) peuvent être décrits par une maille élémentaire (cf. figure 3.5). Cette maille élémentaire se reproduit ensuite sur tout le volume du cristal par translation.

FIG. 3.5 – Exemples de cristaux simples



La principale différence entre les mailles utilisées en cristallographie et celles dont nous avons besoin pour le problème du placement est la dimension : 3 dimensions pour les mailles cristallines, 2 dimensions pour les mailles de l'espace de travail. Cependant, pour des cristaux simples, l'adaptation de la maille à une dimension inférieure ne pose pas de problème. Considérant que les structures créées par la nature répondent souvent à une recherche d'économie, nous avons essayé d'utiliser les méthodes de la cristallographie pour évaluer des adaptations des mailles naturelles.

L'étude d'une maille cristalline se fait généralement en deux phases : calcul de la dimension de la maille par rapport aux grandeurs caractéristiques du problème (en général, le rayon des atomes) puis calcul de la densité de la maille : le nombre d'atomes par unité de volume. Dans notre cas, la grandeur caractéristique sera la portée du capteur, et la densité le nombre de balises par unité de surface.

3.3.1 Zone de visibilité d'une balise.

La notion de zone de visibilité d'une balise est non seulement fondamentale pour l'étude des maillages de balises mais aussi pour l'étude de l'espace de localisation en section 3.4.

Définition 1 *La zone de visibilité d'une balise B pour une orientation θ , notée $Z_\theta(B)$ est définie comme l'ensemble des points de l'espace de travail où la balise B est visible pour une orientation θ du robot.*

Sensibilité du capteur \mathcal{S} : En notant $\vec{d}_\theta = (\cos(\theta), \sin(\theta))^T$ le vecteur orientation du robot, la sensibilité d'un capteur de portée p et d'ouverture α sera définie par l'équation suivante :

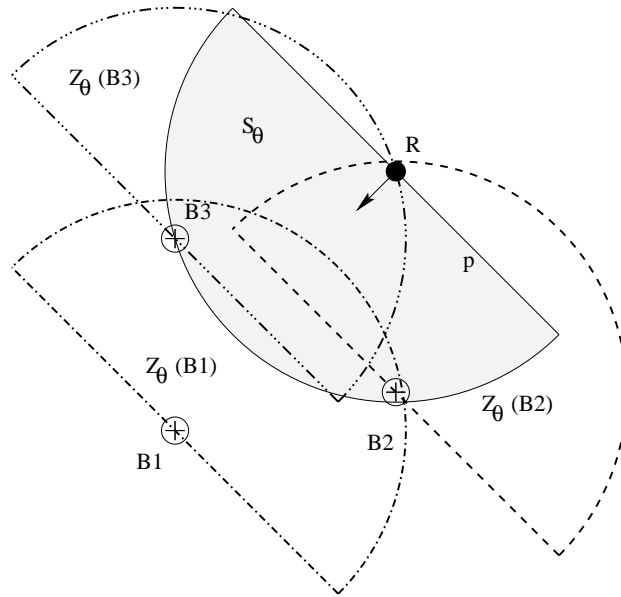
$$\mathcal{S}_\theta^{p,\alpha} = \{\vec{v} / \|\vec{v}\| \leq p \text{ et } |(\vec{d}_\theta, \vec{v})| < \alpha/2\} \quad (3.2)$$

Dans la suite de cette section, on notera $\mathcal{S}_\theta = \mathcal{S}_\theta^{p,\pi/2}$. Si on note R la position du robot et θ son orientation, une balise B est visible de R si $\vec{RB} \in \mathcal{S}_\theta$. Réciproquement, $R \in Z_\theta(B)$ si $-\vec{BR} \in \mathcal{S}_\theta$. On a donc

$$Z_\theta(B) = B \ominus \mathcal{S}_\theta = \{B - \vec{v} / \vec{v} \in \mathcal{S}_\theta\} \quad (3.3)$$

En présence d'obstacles, le calcul de la zone de visibilité est plus compliqué. Il sera évoqué en section 3.4. Les équations 3.2 et 3.3 sont illustrées par la figure 3.6.

FIG. 3.6 – Zones d'influence de 3 balises et sensibilité du capteur



3.3.2 Étude d'une maille carrée

La maille carrée est adaptée de la structure du sel (NaCl). Elle est illustrée sur la figure 3.7.

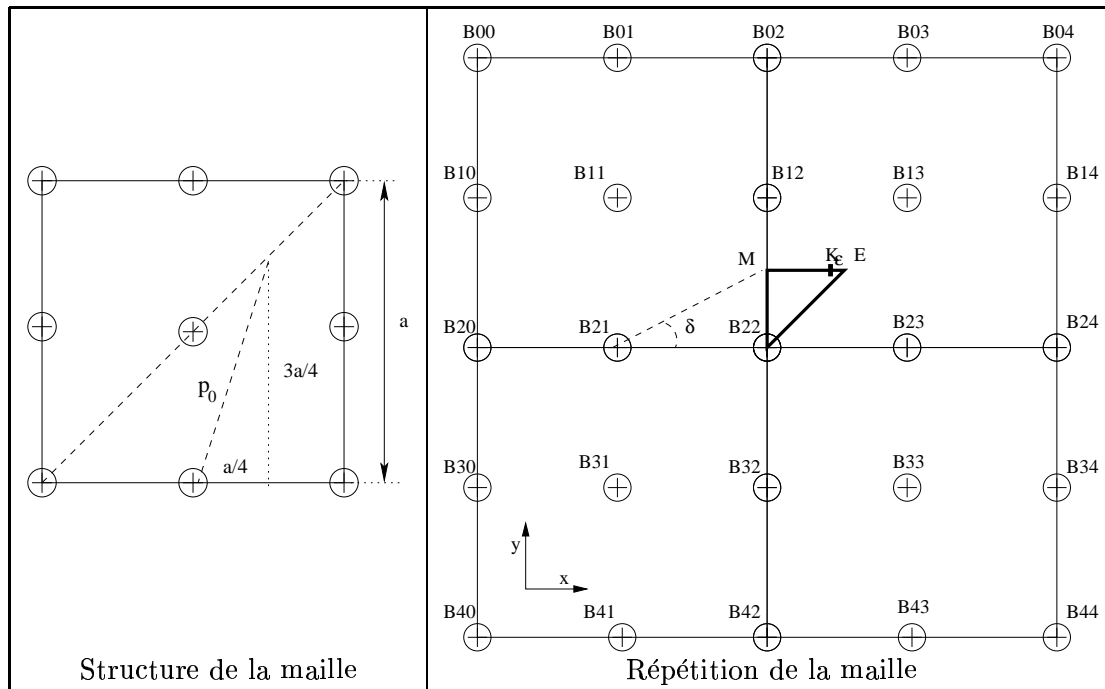
Dimension de la maille : a

Supposons que la maille étudiée soit de côté a (cf. figure 3.7) et cherchons la portée minimale nécessaire pour pouvoir se localiser en tout point du plan (en tout point du plan, au moins deux balises doivent être visibles). Pour des raisons de symétrie, il suffit de montrer que la localisation est possible en tout point du triangle MEB_{22} (en gras).

En premier lieu, montrons que si on a $p = \frac{a\sqrt{10}}{4} = p_0$, alors il est possible de se localiser en tout point de MEB_{22} . Pour cela, on définit

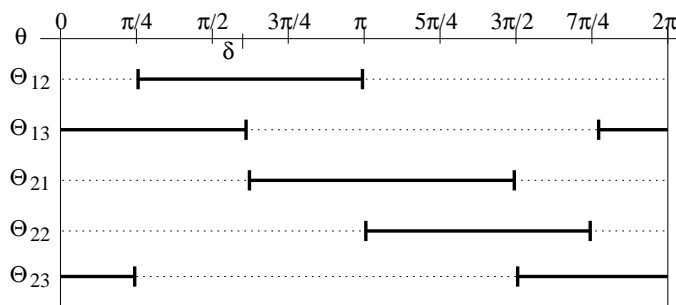
$$\Theta_{ij} = \{\theta / MEB_{22} \subset Z_\theta(B_{ij})\} \quad (3.4)$$

FIG. 3.7 – Maille carrée



qui représente l'ensemble de valeurs de θ pour lesquelles la balise B_{ij} est visible de tout point de MEB_{22} . La figure 3.8 représente Θ pour différentes balises clés. On constate que, quel que soit θ , il y a toujours deux balises visibles de tout point de MEB_{22} : la localisation est donc assurée.

FIG. 3.8 – Couverture de EMB_{22} en fonction de θ



En second lieu, montrons que pour tout $p < p_0$, la localisation n'est pas garantie. En effet, pour une orientation $\theta = 5\pi/4$, quel que soit $p = p_0 - \epsilon$, il existe un point K_ϵ de (ME) proche de E tel que seule B_{22} est visible à partir de K_ϵ .

En conclusion, p_0 est la portée minimale nécessaire. Réciproquement, pour une portée p donnée, une taille de maille $a \leq \frac{4p}{\sqrt{10}}$ permet d'assurer la localisation en tout point de la maille.

Densité de la maille : C

La densité de la maille permet d'évaluer le nombre de balises utilisées par unités de surface. Pour cela, on évalue la surface de la maille : a^2 . Puis on détermine le nombre de balises dans cette maille : 1 balise (au centre) n'est utilisée que dans cette maille, 4 sont partagées entre deux mailles (celles situées sur les arêtes) et 4 sont partagées entre 4 mailles (les sommets). D'où : $\frac{1}{1} + \frac{4}{2} + \frac{4}{4} = 4$ balises sont utilisées. Ce qui donne :

$$C = \frac{4}{a^2} = \frac{5}{2p^2} \quad (3.5)$$

Finalement, on obtient les données numériques suivantes :

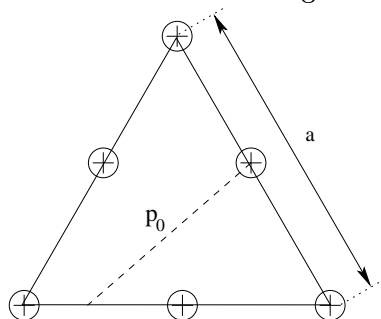
- Pour une portée de 20 mètres, on a $C = 6,25 \cdot 10^{-3} m^{-2}$.
- Une balise couvre $C^{-1} = 160 m^2$
- Pour couvrir $10000 m^2$, il faut plus de 62,5 balises, i.e. 63 balises.

3.3.3 Étude d'une maille triangulaire

Comme on a étudié une maille cubique, on peut étudier une maille triangulaire (figure 3.9). On obtient les données suivantes :

- Taille de la maille : $a = \frac{4p}{\sqrt{10}}$.
- Densité : $C = \frac{5}{p^2\sqrt{3}}$.
- Pour une portée de 20 mètres, on a $C = 7,21 \cdot 10^{-3} m^{-2}$.
- Une balise couvre $C^{-1} = 139 m^2$
- Pour couvrir $10000 m^2$, il faut plus de 72,1 balises, i.e. 73 balises.

FIG. 3.9 – Maille triangulaire



3.3.4 Commentaires

Après cette étude, on constate que les méthodes de la cristallographie permettent d'étudier a posteriori une solution au problème du placement dans un espace très large. Par exemple, la maille carrée permet d'obtenir un placement des balises plus efficace que la maille triangulaire : les balises sont donc mieux utilisées. L'intérêt de cette méthode est la mise en évidence de la nécessité de ne pas placer les balises au hasard, car il existe des placements qui permettent de couvrir plus d'espace avec un même nombre de balises.

Cependant, il faut remarquer que cette technique de placement demande des circonstances d'applications très particulières (pas d'obstacles, espace très grand) et que même

si on peut toujours assurer la visibilité de deux balises, les mailles sont tellement riches en symétries qu'il n'est pas possible de se localiser sans estimation préalable de la position. A l'initialisation du module de localisation, une intervention humaine est donc indispensable.

3.4 Solution numérique pour un capteur réel

Les sections 3.2 et 3.3 nous ont montrée qu'avec un capteur d'ouverture inférieure à π , il existera toujours des parties de l'espace de travail où la localisation sera impossible. Elles ont aussi mis en évidence la possibilité d'améliorer les capacités de localisation en choisissant correctement le placement des balises. Le but de cette section est d'utiliser des techniques d'optimisation numérique pour déterminer une solution optimale au problème du placement.

Le robot sera assimilé à un point $(x, y) \in \mathbb{R}^2$ doté d'une orientation θ . L'espace de travail, noté $\mathcal{W} \subset \mathbb{R}^2$ sera supposé polygonal avec éventuellement des trous polygonaux : les obstacles. Le robot disposera d'un capteur télémétrique de portée p et d'ouverture angulaire α quelconque. Pour simplifier l'exposé, on supposera que le robot a besoin d'au moins deux balises pour se localiser, cependant, on pourrait tout à fait se passer de cette hypothèse.

3.4.1 Visibilité d'une balise : \mathcal{V}

Il est possible d'étendre la notion de zone de visibilité définie en 3.3.1 de façon à prendre en compte les limites de l'environnement et les obstacles. On désignera par polygone de visibilité $(\mathcal{P}(B))$ d'une balise B l'ensemble des points de l'espace de travail d'où cette balise est visible⁴. Dans [GMR95] et [AGTZ98] par exemple, sont présentées des méthodes permettant de calculer le polygone de visibilité d'un point en $O(\log(n))$ où n est le nombre de sommet de l'espace de travail. Ces méthodes ne prennent pas en compte le cas d'un capteur de portée finie et d'ouverture angulaire limitée. Cependant, si on veut calculer la zone de visibilité pour une orientation θ donnée, on connaît la zone d'influence de chaque balise. La visibilité \mathcal{V}_θ d'une balise B sera donc l'intersection de ces deux objets :

$$\mathcal{V}_\theta(B) = \mathcal{P}(B) \cap Z_\theta(B) \quad (3.6)$$

3.4.2 Espace de localisation

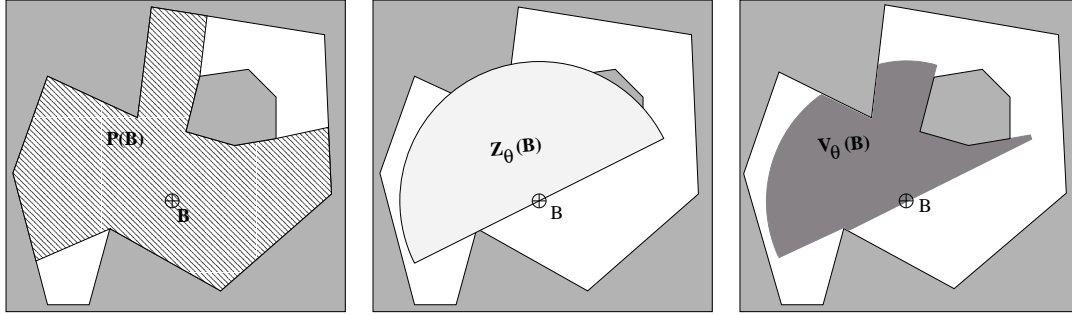
Définition 2 *Supposons qu'il y ait n balises $\{B_i\}_{i=1..n}$ réparties dans l'espace de travail. L'espace de localisation \mathcal{L} d'un robot R est définie par l'équation suivante :*

$$\mathcal{L} = \{(x, y, \theta) / \exists i, j, (i \neq j) \text{ et } ((x, y) \in \mathcal{V}_\theta(B_i)) \text{ et } ((x, y) \in \mathcal{V}_\theta(B_j))\} \quad (3.7)$$

\mathcal{L} correspond donc à l'ensemble des points (x, y) où, pour une orientation θ , on peut voir au moins deux balises. Si on veut considérer une localisation demandant plus ou moins de balises, il suffit d'adapter l'équation 3.7.

⁴Une balise est visible d'un point de l'espace de travail si le segment joignant ce point et la balise reste dans l'espace de travail.

FIG. 3.10 – Construction de la visibilité de B



3.4.3 Construction de l'espace de localisation

Il est possible d'obtenir assez simplement une approximation d'espace de localisation. Pour cela, on le discrétise suivant les valeurs de θ : pour une orientation θ donnée, il est en effet possible de déterminer exactement l'espace de localisation en construisant une partition de l'espace de travail en cellules d'influence, notée $\mathcal{P}^\theta = \{\mathcal{I}_i^\theta\}$. Une telle partition doit vérifier les propriétés suivantes :

- $\mathcal{W} = \bigcup_i \mathcal{I}_i^\theta$.
- L'ensemble des balises visibles (éventuellement vide) doit être le même en tout point de \mathcal{I}_i^θ .
- $i \neq j \implies \text{Aire}(\mathcal{I}_i^\theta \cap \mathcal{I}_j^\theta) = 0$.
- En notant V_i l'ensemble des balises visibles en \mathcal{I}_i^θ , pour toute cellule \mathcal{I}_j^θ voisine⁵ de \mathcal{I}_i^θ , $V_i \neq V_j$.

En pratique, on construit en même temps la partition en cellules d'influence et l'ensemble des balises visibles dans chaque cellule. L'algorithme de la figure 3.11 donne un moyen d'obtenir cette partition. On notera $d\theta$ le pas de la discrétisation suivant θ .

FIG. 3.11 – Construction de la partition en cellules d'influence

On notera $\mathcal{P}_k^\theta = \{(\mathcal{I}_{k,i}^\theta, V_{k,i})\}_{i=1 \dots n_k}$ l'état de la partition pour une orientation θ à l'itération k .

Au début, dans tout l'espace de travail, aucune balise n'est visible.

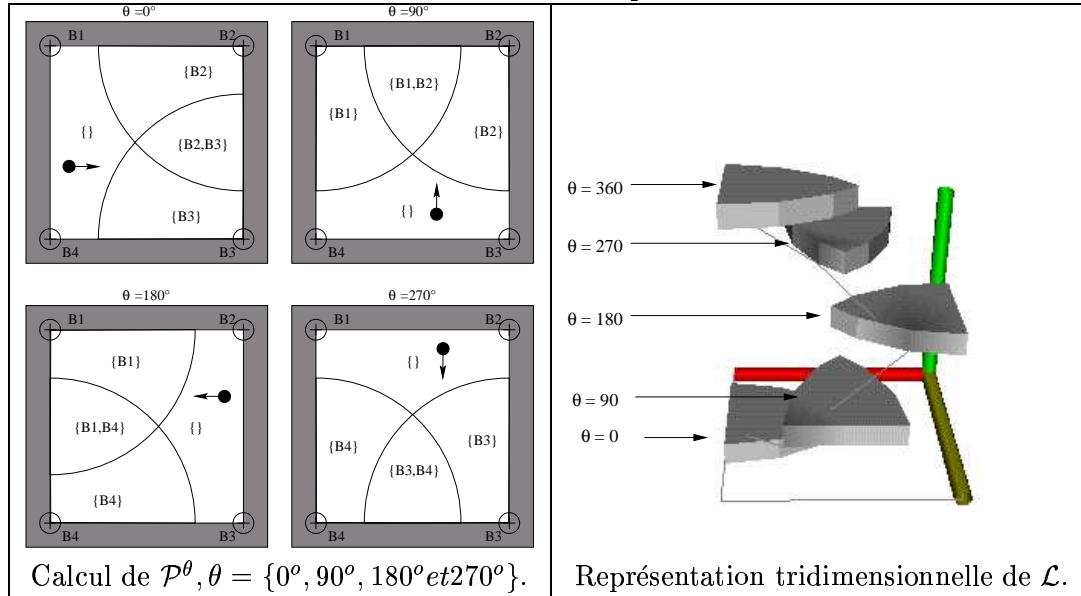
- 1 $\mathcal{P}_0^\theta \leftarrow \{(\mathcal{W}, \emptyset)\}$
- 2 Pour toute balise B_k ($k = 1 \dots n$),
Raffinement de la partition
- 3 $\mathcal{P}_k^\theta \leftarrow$
- 4 $\left\{ (\mathcal{I}_{k-1,i}^\theta \setminus \mathcal{V}_\theta(B_k), V_{k-1,i}) \mid i = 1 \dots n_{k-1} \right\} \cup$
- 5 $\left\{ (\mathcal{I}_{k-1,i}^\theta \cap \mathcal{V}_\theta(B_k), V_{k-1,i} \cup \{B_k\}) \mid i = 1 \dots n_{k-1} \right\}$
- 6 Suppression des ensembles vides.
 $\mathcal{P}_k^\theta \leftarrow \mathcal{P}_k^\theta \setminus \{(I, V) \mid I = \emptyset\}$

En extrayant de \mathcal{P}^θ les cellules associées à deux balises ou plus, on obtient le sous-ensemble de l'espace de travail où le robot peut se localiser avec une orientation θ , noté

⁵Deux cellules seront dites voisines si leur union est connexe.

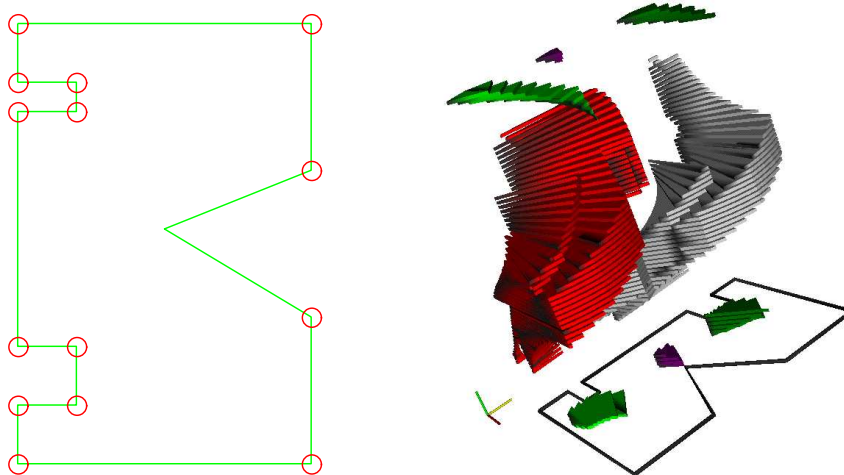
\mathcal{L}_θ . Si on calcule \mathcal{L}_θ pour suffisamment de valeurs de $\theta \in [0, 2\pi]$ ($d\theta$ suffisamment petit), on obtient alors une bonne approximation de \mathcal{L} . La figure 3.12 illustre une approximation de \mathcal{L} avec un pas de 90° selon θ dans un espace de travail très simple. A partir du calcul de \mathcal{L}_θ pour différentes valeurs de θ , il est aussi possible de construire une représentation tridimensionnelle de \mathcal{L} (à droite sur la figure 3.12).

FIG. 3.12 – Construction de l'espace de localisation



En utilisant un pas de 5° , une ouverture de 60° et une portée faible dans un environnement plus complexe, on obtient l'espace de localisation représenté dans la figure 3.13. Dans les figures 3.12 et 3.13, l'axe vertical (en vert) représente θ et l'espace de travail est affiché autour de la couche $\theta = 0$.

FIG. 3.13 – Espace de localisation complexe



3.4.4 Optimisation de l'espace de localisation

Intuitivement, l'espace de localisation représente le sous-ensemble de l'espace de travail dans lequel le robot peut évoluer en maintenant une localisation fiable. Il y a donc trois paramètres dont il faut tenir compte dans l'optimisation :

- L'espace de localisation doit avoir un volume maximum, ce qui signifie que l'on veut pouvoir aller n'importe où avec n'importe quelle orientation dans l'espace de travail.
- L'espace de localisation doit être connexe. En effet, s'il existe plusieurs composantes connexes, alors dans chacune d'elles, la localisation sera fiable, mais il sera impossible de passer de l'une à l'autre en maintenant cette fiabilité. Sur la figure 3.13, les composantes connexes ont chacune une couleur différente.
- Enfin, pour limiter l'influence des erreurs de mesure (cf. section 2.4), les balises ne doivent pas être trop proches les unes des autres.

Évaluation du volume de \mathcal{L} : On obtient une approximation du volume de l'espace de localisation en utilisant l'équation 3.8 :

$$\text{Volume}(\mathcal{L}) \approx \sum_{i=0}^{i < 2\pi/d\theta} \text{Aire}(\mathcal{L}_i d\theta) d\theta \quad (3.8)$$

Plus $d\theta$ est petit, plus cette approximation sera proche du volume exact (il s'agit d'un calcul d'intégrale par la méthode des rectangles).

Évaluation de la connexité de \mathcal{L} : Pour évaluer la connexité de \mathcal{L} , on utilise l'équation 3.9 :

$$\mathcal{L} \approx \bigcup_{\theta} \mathcal{L}_{\theta} = \bigcup_{\theta} \left(\bigcup_i \mathcal{I}_i^{\theta} \right) \quad (3.9)$$

Cette équation traduit le fait que \mathcal{L} est une union de cellules d'influence (cf. section 3.4.3). Pour déterminer les parties connexes, on construit un graphe \mathcal{G} dont les noeuds sont les cellules d'influences. Pour l'ajout des arcs, deux cas sont à considérer :

- On ajoute un arc entre deux cellules appartenant à un même \mathcal{L}_{θ} si elles sont contiguës.
- On ajoute un arc entre une cellule de \mathcal{L}_{θ} et une de $\mathcal{L}_{\theta+d\theta}$ si leurs projections dans le plan $\theta = 0$ ont une intersection non vide. Intuitivement, cela signifie que les deux cellules sont l'une au-dessus de l'autre.
- Seuls les arcs précédents sont ajoutés dans le graphe.

Un tel graphe sera appelé **graphe de localisation**. Sur la figure 3.12, les arcs entre les cellules sont dessinés sous la forme de fines lignes grises : le graphe est connexe.

Le nombre de parties connexes dans \mathcal{L} est alors le nombre de parties connexes dans \mathcal{G} , ce qui peut être déterminé en temps proportionnel au nombre de noeuds dans \mathcal{G} ⁶.

Minimisation des erreurs de mesure : Pour maximiser la fiabilité des erreurs de mesures, il faudrait garantir que deux balises sont toujours visibles avec un angle relatif proche de $\pi/2$. Cependant, si on considère deux balises quelconques, lorsque la position

⁶La recherche des parties connexes est équivalente à une exploration du graphe.

du robot approche de la droite passant par les balises, l'angle relatif tend vers 0. Il faudrait donc intégrer la minimisation de l'influence des erreurs de mesure dans le processus d'optimisation. Une première façon de le faire consiste à définir une borne inférieure d_{lim} à la distance entre deux balises. Il serait intéressant de développer un critère moins grossier, mais faute de temps, ceci n'a pas pu être fait dans le cadre de cette étude.

Fonction de coût : L'optimisation de l'espace de travail peut maintenant se faire en utilisant des techniques d'optimisation numérique telles que les algorithmes génétiques⁷. Pour ces algorithmes, il suffit de définir une fonction d'évaluation qui associe un score à une distribution des balises dans l'espace de travail. Pour des facilités d'implémentation, nous avons choisi de ne chercher à placer des balises que sur les bords de l'espace de travail. Ceci se justifie aussi par le fait que dans la plupart des cas, on ne désire pas encombrer l'espace de travail en y ajoutant des balises. Cependant, le raisonnement utilisé pour l'optimisation est indépendant de ce choix.

La fonction de coût utilisée est la fonction f suivante :

$$\begin{aligned} d_{\min} &= \text{plus petite distance entre deux balises} \\ \mathcal{K} &= \begin{cases} (d_{\min}/d_{\text{lim}})^2 & \text{si } d_{\min} \leq d_{\text{lim}} \\ 1 & \text{sinon} \end{cases} \\ f &= \mathcal{K} \frac{\text{Volume}(\mathcal{L})}{\text{Nombre de parties connexes dans } \mathcal{L}} \end{aligned}$$

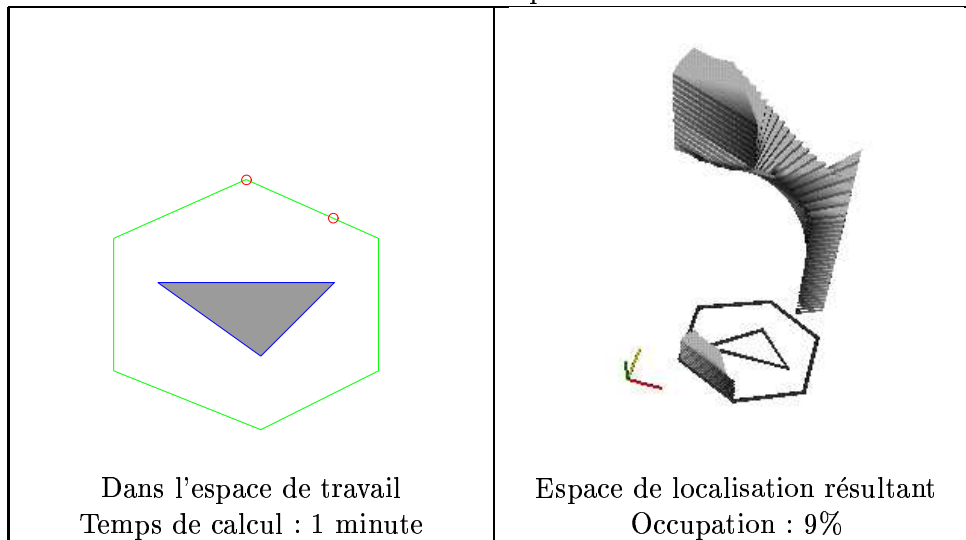
La maximisation de cette fonction de coût correspond à la maximisation du volume moyen d'une partie connexe de \mathcal{L} en cherchant à maintenir une distance minimale entre les balises. La fonction de coût sera donc maximum si une partie connexe unique occupe tout l'espace. Pour assurer une distance minimale entre les balises, nous avons choisi une décroissance quadratique de \mathcal{K} lorsqu'il existe des balises trop proches. De plus, le fait de diviser par le nombre de parties connexes correspond à accorder une très forte importance à la connexité de l'espace de localisation. Cette importance se justifie lorsqu'on recherche des capacités de localisation fiables à tout instant. Si on accepte qu'il y ait des moments où la localisation est impossible, alors on peut envisager une fonction de coût donnant plus d'importance au volume de l'espace de localisation.

3.4.5 Résultats de l'optimisation

Pour illustrer le fonctionnement de la méthode, nous illustrons le placement optimal de 2, 6 et 20 balises dans un environnement hexagonal (figures 3.14, 3.15 et 3.16). Pour ces tests, nous avons choisi une portée égale à la largeur de l'hexagone et une ouverture angulaire de 120 degrés. Avec 20 balises, l'espace de localisation remplit presque tout l'espace disponible. Seules les orientations qui placent le robot dans la situation de la figure 3.2 génèrent encore des zones où il est impossible de se localiser. Cependant, quoi qu'on fasse, de telles zones existeront toujours (cf. section 3.2). Finalement, le choix du nombre de balises à utiliser dépendra de l'application envisagée.

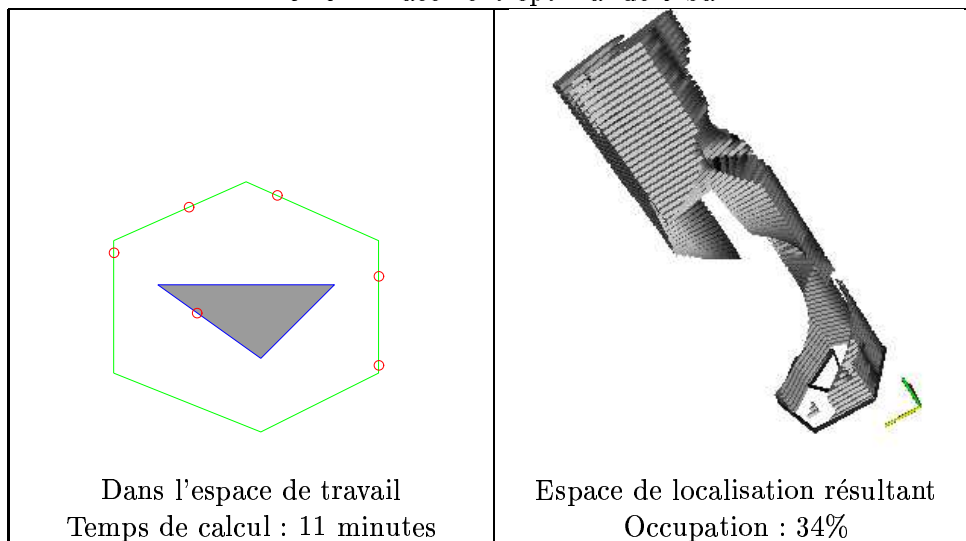
⁷Une librairie d'algorithmes génétiques bien documentée et facile à prendre en main est disponible à l'adresse <http://lancet.mit.edu/ga>.

FIG. 3.14 – Placement optimal de 2 balises



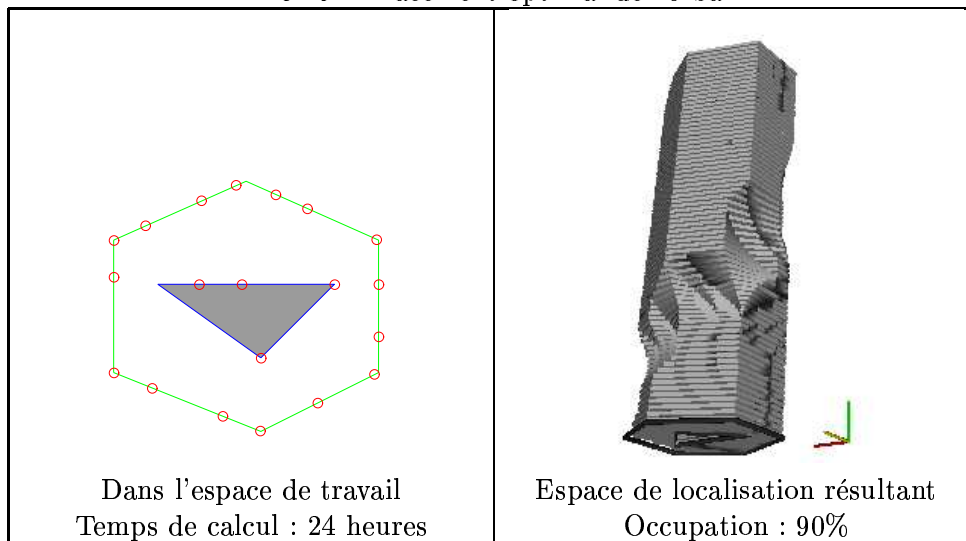
Optimisation : On peut aussi remarquer le coût très important de l'optimisation. Cependant, on pourrait réduire considérablement le temps de calcul en optimisant les méthodes de calcul du polygone de visibilité (méthode décrite dans [GMR95] par exemple). De cette façon, il est probable que l'on puisse économiser de l'ordre de 20% du temps de calcul. Faute de temps, ces optimisations n'ont pas été testées.

FIG. 3.15 – Placement optimal de 6 balises



Nombre de balises à placer	Occupation (%)	Temps de calcul
2	9	1min
4	19	3min
6	34	11min
8	55	1h10min
12	74	40min
16	83	8h
20	90	24h

FIG. 3.16 – Placement optimal de 20 balises



Parallélisation : Une autre façon de réduire le temps de calcul nécessaire serait de paralléliser le code. Cette parallélisation pourrait être très efficace pour les raisons suivantes :

- Les algorithmes génétiques sont fondés sur l'évaluation d'individus indépendants et sont donc aisément parallélisables.
- Le calcul de \mathcal{L} demande au préalable de construire les polygones de visibilité de chaque balise. Ces polygones sont indépendants.
- Chaque calcul de \mathcal{L}_θ est indépendant des autres.

Seule l'évaluation de la connexité demanderait a priori une exécution séquentielle. On peut supposer que les gains issus de la parallélisation seraient considérables. Cependant, il faut noter que la parallélisation n'est pas indispensable car le calcul du placement des balises n'est effectué qu'une fois lors de la conception de l'environnement, moment où, a priori, on n'est pas soumis aux problèmes de temps réel.

Comparaison avec la méthode analytique (3.2) : En raison du temps nécessaire pour effectuer l'optimisation d'un espace de travail, nous avons choisi de comparer les résultats numériques et les résultats issus de la section 3.2 sur un exemple simple : un environnement carré, sans obstacles. Nous avons choisi une zone interdite large d'un cinquième du côté du carré et une ouverture de 90° . Les résultats sont présentés dans la

FIG. 3.17 – Comparaison des méthodes analytique et numérique

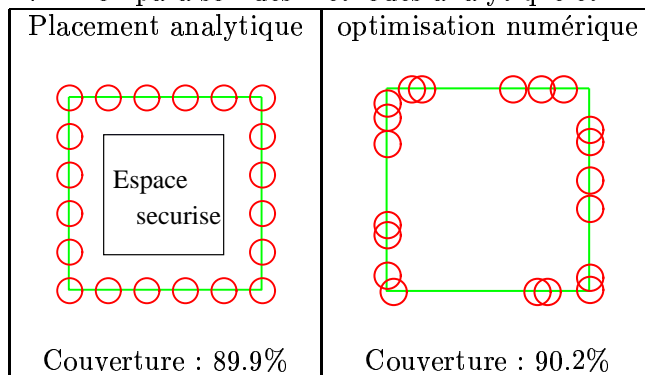


figure 3.17.

On constate que dans ce cas simple, la méthode numérique donne des résultats aussi bons que le placement systématique. Dans ce cas, il est clair qu'il n'y a donc pas d'intérêt à utiliser plusieurs heures de calcul pour obtenir un résultat équivalent. Cependant, l'avantage de la méthode numérique réside surtout dans sa généralité : quel que soit le capteur (portée limitée, ouverture plus ou moins grande), il sera possible de trouver un placement des amers optimal sans modifier la méthode.

Par ailleurs, les méthodes présentées dans cette section permettent non seulement d'optimiser le placement des amers dans un espace de travail, mais aussi d'évaluer a posteriori les capacités de localisation offertes par un placement particulier. Il s'agit donc d'un outils très utile lorsqu'on doit concevoir un environnement maximisant les capacités de localisation pour des robots autonomes

3.5 Conclusion sur l'organisation de l'espace de travail

Dans ce chapitre, nous avons abordé le problème du placement optimal d'amers artificiels dans un espace de travail polygonal. Après avoir évalué deux placements systématiques qui ont permis de mettre en évidence l'existence de zone de non-localisation dans le cas général et l'intérêt d'un placement judicieux des amers, nous avons développé une méthode d'optimisation numérique. Cette méthode est fondée sur les notions d'espace de localisation et de graphe de localisation. Elle s'est révélée efficace pour garantir une localisation fiable dans une partie connexe remplissant au mieux l'espace des configurations du robot. De plus, même si la construction de l'espace de localisation a été présentée ici dans le cas d'un robot nécessitant deux amers pour se repérer, la méthode est généralisable immédiatement à d'autres types de localisation : triangulation à partir du relevé d'angles relatifs entre trois amers, localisation à partir d'un seul amer, etc... De même, elle est généralisable à des capteurs d'ouverture angulaire et de portée quelconques.

L'optimisation numérique du placement des balises pourrait être encore améliorée. Tout d'abord, il est sans doute possible de réduire le temps de calcul nécessaire. Ensuite, il serait sans doute très intéressant de définir une fonction de coût permettant de mieux prendre en compte l'influence des erreurs de mesure sur la qualité de la localisation.

Finalement, à partir de la notion de graphe de localisation, il est assez naturel de

s'interroger sur les applications d'algorithmes de recherche du plus court chemin. Dans le chapitre suivant, nous envisagerons donc d'utiliser le graphe de localisation pour construire un planificateur de trajectoire.

Chapitre 4

Ouverture sur la planification

4.1 Définition du problème

Après les chapitres sur la localisation et l'organisation de l'espace de travail, ce chapitre est destiné à une première étude de l'application de la notion d'espace de localisation à la planification de trajectoire. Contrairement aux autres chapitres, les idées présentées ici sont principalement des conjectures qui n'ont pu être développées complètement faute de temps.

Planification : Le problème considéré dans ce chapitre concerne un robot mobile autonome et holonome¹. On désigne par “planification de trajectoire” la construction d'une commande amenant le système d'une configuration (position et orientation) initiale à une configuration finale. Ce problème est un aspect fondamental de la robotique. Parmi les questions liées à la planification, on peut citer la gestion des obstacles de l'espace de travail, l'optimisation du chemin parcouru, la prise en compte de contraintes cinématiques particulières (robot de type voiture par exemple) et la planification orientée capteur. C'est cette dernière question qui va nous intéresser ici. On parle de planification orientée capteur lorsque les trajectoires calculées intègrent une contrainte sur les capteurs : par exemple, on peut vouloir garantir qu'en tout point de la trajectoire, il sera possible d'utiliser un capteur particulier pour se localiser. La notion d'espace de localisation semble offrir de telles perspectives.

Espace de localisation : En cherchant à optimiser le placement d'amers dans un espace de travail (section 3.4), nous avons introduit les notions d'espace de localisation \mathcal{L} et de graphe de localisation \mathcal{G} . Supposons que l'on ait à déterminer la trajectoire d'un robot entre une configuration initiale $q_i = (x_i, y_i, \theta_i)$ et une configuration finale $q_f = (x_f, y_f, \theta_f)$ en maintenant sur toute la trajectoire les capacités de localisation. Par définition, une telle trajectoire doit être entièrement contenue dans l'espace de localisation. Si q_i et q_f ne sont pas dans la même partie connexe de \mathcal{L} ou si l'une des deux n'est pas dans \mathcal{L} , il ne sera pas possible de trouver une trajectoire répondant aux contraintes. Dans le cas contraire, on sait que la trajectoire existe, il suffit donc de la trouver.

¹robot pouvant suivre n'importe quelle courbe de l'espace des configurations

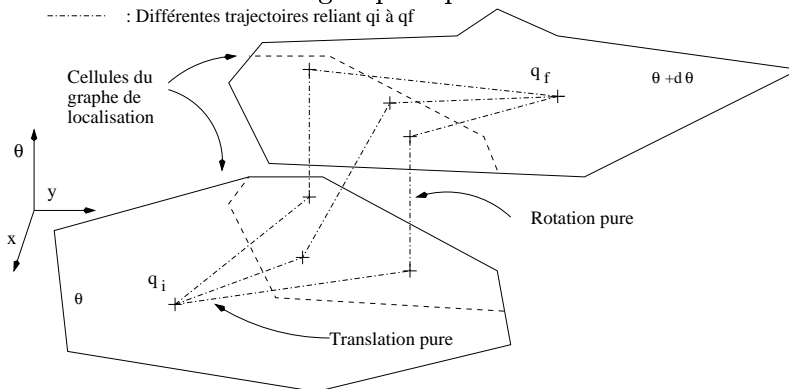
Graphe de localisation : L'idée de base que nous allons utiliser est que le graphe de localisation offre une représentation structurée de l'espace des configurations permettant de traiter le problème de la planification. En effet, lorsqu'on dispose d'un graphe dont les arêtes sont valuées, il existe des algorithmes pour trouver le plus court chemin entre deux noeuds du graphe (Dijkstra...). Chaque noeud de \mathcal{G} représente une cellule de l'espace des configurations dans laquelle la localisation est assurée. On peut donc trouver les cellules \mathcal{L}_i et \mathcal{L}_f contenant les configurations q_i et q_f et déterminer le plus court chemin entre ces deux cellules dans \mathcal{G} .

4.2 Problèmes soulevés par l'utilisation du graphe de localisation

Valuation des arcs : Les arcs de \mathcal{G} relient des ensembles de configuration de $\mathbb{R}^2 \times SO^1$ (l'ensemble des orientations du robot), il faut donc définir une distance dans cet espace. Une telle distance doit permettre de mettre en relation les déplacements en rotation et les déplacements en translation. Selon le robot, les rotations ne représenteront pas la même difficulté. Par exemple, pour un robot dont l'orientation correspond à l'orientation d'une caméra mobile, les rotations ne posent pas de réel problème. Pour un robot de type voiture, une rotation pure (x et y restant constants) n'est pas réalisable. Le choix de la distance est donc déterminant. De ce choix vont dépendre les trajectoires générées par le planificateur et leur faisabilité.

Passage du graphe à l'espace de localisation : Une fois la notion de distance définie, il est aisé de trouver un itinéraire de la cellule initiale à la cellule finale dans le graphe de localisation. Cet itinéraire donne un ensemble de configurations de passage. Pour calculer les chemins entre les points de passage, il est nécessaire de définir précisément les stratégies de passage d'une cellule à une autre, en particulier dans le cas d'un changement d'orientation. De même que pour le choix de la distance dans \mathcal{L} , il peut exister des spécificités qui rendent certaines stratégies préférables.

FIG. 4.1 – Différentes stratégies pour passer d'une cellule à une autre



Gestion de la non connexité : On suppose maintenant qu'il est possible d'utiliser le graphe de localisation pour planifier un chemin d'une configuration à une autre. Cependant, pour qu'un tel chemin existe, il faut que les deux configurations appartiennent à la même composante connexe de \mathcal{G} . Toutefois, il peut être nécessaire de passer d'une composante connexe à une autre. Cela signifie qu'il faut choisir une trajectoire partant d'un point de la première composante connexe et arrivant à un point de la seconde mais qui va être exécutée sans localisation. Plusieurs paramètres peuvent être pris en compte dans le choix de cette trajectoire : on peut chercher à minimiser la distance parcourue hors de l'espace de localisation, sachant que cette distance ne pourra pas être inférieure à la distance de Hausdorff entre les deux parties connexes. On peut aussi utiliser une modélisation de l'erreur accumulée sur l'estimation de la position en fonction du mouvement effectué, pour essayer de trouver la trajectoire qui minimise le risque de ne pas atteindre la partie connexe destination.

4.3 Conclusion sur la planification

Les notions d'espace de localisation et de graphe de localisation semblent promettre des propriétés utiles et intéressantes pour la planification. En effet, pour la planification, l'espace de localisation est plus riche et mieux structuré que l'espace de travail. De plus, par rapport à une planification dans l'espace des configurations seul, la planification dans l'espace de localisation permet de garantir des capacités de localisation qui peuvent rendre l'exécution d'une trajectoire plus sûre.

Il serait donc intéressant de pousser plus en avant l'étude de ces notions.

Chapitre 5

Conclusion

Cette étude nous a permis de faire un tour d'horizon du problème de la localisation. En premier lieu, nous avons vu comment utiliser des amers artificiels et un télémètre laser pour construire un procédé de localisation fiable, robuste et précis. L'utilisation de balises circulaires dans un environnement très mobile nous a amené à définir une méthode de traitement des données laser spécifique, fondée sur un automate de Mealy. Le choix d'utiliser des balises non différenciables a demandé la mise en place de techniques de mise en correspondance issues de la recherche en vision par ordinateur et de la théorie des graphes. Enfin, pour intégrer les données provenant de nos divers capteurs, il a fallu utiliser le classique filtre de Kalman en l'adaptant pour tenir compte des problèmes de latence spécifiques à notre plateforme.

En second lieu, nous avons abordé un problème qui n'a pas encore suscité un grand intérêt dans la littérature : le placement optimal des amers dans l'environnement. La mise en oeuvre de solutions à ce problème permet de systématiser le placement intuitif utilisé dans toutes les expérimentations sur la localisation. Nous avons d'abord recherché une solution analytique dans le cas d'un capteur de portée infinie et d'ouverture angulaire inférieure à π . Ensuite, nous nous sommes inspirés des cristaux naturels pour étudier le placement des balises dans un environnement infini sans obstacles. Finalement, nous avons utilisé des méthodes d'optimisation numérique pour traiter le cas d'un capteur réel dans un environnement réel. En plaçant les amers de façon optimale, on garantit la possibilité d'acquérir des données extéroceptives avec une fréquence élevée, ce qui rend la localisation encore plus fiable, robuste et précise. Par ailleurs, pour tenir compte du placement optimal des balises dans l'espace de travail, il est apparu assez naturel d'envisager un planificateur de trajectoire capable d'utiliser ces informations pour générer des trajectoires plus fiables.

Un tel planificateur fait partie des sujets que nous souhaitons étudier dans un avenir proche. Il pourrait aussi être intéressant d'essayer d'optimiser le placement des balises en cherchant à minimiser l'influence des erreurs de mesure. Enfin, la précision obtenue par le processus de localisation permet d'envisager sereinement une construction de l'environnement au fur et à mesure des déplacements du robot : dès qu'une balise inconnue est détectée, sa position est mémorisée, ce qui étend la partie de l'espace de travail dans lequel le robot peut évoluer. Il faudra alors mettre en place des mécanismes de gestion et de correction des erreurs à rapprocher du travail de [Cro89].

Finalement, les objectifs du travail présenté ici ont été atteints : les Cycabs disposent maintenant d'une architecture de localisation complète pouvant être facilement intégrée

dans d'autres expérimentations : manoeuvres automatiques, assistance à la conduite,... Il s'agit donc d'un premier succès. Par ailleurs, nous considérons que la définition des notions d'espace de localisation et de graphe de localisation constitue un résultat plus fondamental, riche en perspectives intéressantes.

Bibliographie

- [AGTZ98] Aronov, Guibas, Teichmann, and Zhang. Visibility queries in simple polygons and applications. In *ISAAC : 9th International Symposium on Algorithms and Computation (formerly SIGAL International Symposium on Algorithms), Organized by Special Interest Group on Algorithms (SIGAL) of the Information Processing Society of Japan (IPSJ) and the Technical Group on Theoretical Foundation of Computing of the Institute of Electronics, Information and Communication Engineers (IEICE)*, 1998.
- [AMdIES98] J.M. Armingol, L. Moreno, A. de la Escalera, and M.A. Salichs. Landmark perception planning for a mobile robot localization. In *Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation*, 1998.
- [AO01] F.M. Antoniali and G. Oriolo. Robot localization in nonsmooth environments : experiments with a new filtering technique. *Submitted at the 2001 IEEE Int. Conf. on Robotics and Automation*, 2001.
- [ATS00] Arras, Tomatis, and Siegwart. Multisensor on-the-fly localization using laser and vision. In *IEEE/RSJ*, 2000.
- [BD96] A.J. Briggs and B.R. Donald. Robust geometric algorithms for sensor planning. In *Proc. of the second international workshop on algorithmic foundations of robotics*, Toulouse, France, 1996.
- [BF96] J. Borenstein and L. Feng. Gyrodometry : a new method for combining data from gyros in mobile robots. In *Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation*, 1996.
- [BFHS96] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *AAAI/IAAI, Vol. 2*, pages 896–901, 1996.
- [BG94] Betke and Gurvits. Mobile robot localization using landmarks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 135–142, 1994.
- [BNRDW00] T. Bailey, E.M. Nebot, J.K. Rosenblatt, and H.F. Durrant-Whyte. Data association for mobile robot navigation : a graph theoretic approach. In *Proc. of the 2000 IEEE Int. Conf. on Robotics and Automation*, 2000.
- [BSB⁺00] BriggsA.J., D. Scharstein, D. Braziunas, C. Dima, and P. Wall. Mobile robot navigation using self-similar landmarks. In *Proc. of the 2000 IEEE Int. Conf. on Robotics and Automation*, 2000.

- [BSS96] D. Boley, E. Steinmetz, and K. Sutherland. Robot localization from landmarks using recursive total least squares. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, 1996.
- [BSTL95] C. Becker, J. Salas, K. Tokusei, and J.C. Latombe. Reliable navigation using landmarks. In *Proc. of the 1995 IEEE Int. Conf. on Robotics and Automation*, 1995.
- [CB96] V. Caglioti and G. Borghi. Updating position estimate of mobile robots in curvilinear environments via range measures, 1996.
- [Cro89] J.L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Proc. of the 1989 IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, 1989.
- [CS] L. Charbonnier and O. Strauss. A suitable approximation for laser range finder data.
- [DS98] Dubrawski and Siematkowska. A method for tracking pose of a mobile robot equipped with a scanning laser range finder. In *Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation*, 1998.
- [DS00] G. Dedeoglu and G. Sukhatme. Landmark-based matching algorithm for cooperative mapping by autonomous robots, 2000.
- [GMR95] Guibas, Motwani, and Raghavan. The robot localization problem. In Goldberg, Halperin, Latombe, and Wilson, editors, *Algorithmic Foundations of Robotics, The 1994 Workshop on the Algorithmic Foundations of Robotics*, A. K. Peters, 1995.
- [GNB00] Guivant, Nebot, and Baiker. Autonomous navigation and map building using laser range sensor in outdoor application. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages pp 3817–3822, 2000.
- [Gud00] J. Gudmundsson. Covering polygons¹, 2000.
- [HBBC96] P. Herbért, S. Betgé-Brezetz, and R. Chatila. Probabilistic map learning : Necessity and difficulties. In Leo Dorst Michiel van Lambalgen and Frans Voorbraak, editors, *Reasoning with Uncertainty in Robotics*, pages 307–321. Springer, 1996.
- [Lam96] A. Lambert. *Planification de Tâches-Robots par une Approche Possibiliste*, chap. 3. PhD thesis, Université de technologie de Compiègne, 1996.
- [LHSL00] F. Large, J. Hermosillo, S. Sekhavat, and C. Laugier. Using artificial neural networks to improve sensor-based maneuvers for a car-like vehicle. In *Proc. of the 2000 Int. Conf. on Intelligent Autonomous System*, 2000.
- [LL92] A. Lazanas and J.-C. Latombe. Landmark-based robot navigation. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pages 816–822, San Jose, California, 1992. AAAI Press.
- [LM94] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *CVPR94*, pages 935–938, 1994.

¹http://www.dna.lth.se/home/Joakim_Gudmundsson/Indexkatalog/covering.html

- [LSL98] J.P. Laumond, S. Sekhavat, and F. Lamiroux. *Guidelines in nonholonomic motion planning for mobile robots*, chapter 1. Springer, 1998.
- [MA98] C. Madsen and C. Andersen. Optimal landmark selection for triangulation of robot position. *Journal of Robotics and Autonomous Systems*, 13(4) :pp277–292, 1998.
- [Ols97] C.F. Olson. Mobile robot self-localization by iconic matching of range maps. In *Proceedings of the International Conference on Advanced Robotics*, pages 447–452, 1997.
- [Ols98] C.F. Olson. A probabilistic formulation for hausdorff matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 150–156, 1998.
- [RB97] S. Roumeliotis and G. Bekey. An extended kalman filter for frequent local and infrequent global sensor data fusion, 1997.
- [RGI96] Russell, Greiner, and Isukapalli. Learning to select useful landmarks. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 26(3), 1996.
- [Ruc95] W.J. Rucklidge. Locating objects using the hausdorff distance. In *Proc. of the 1995 IEEE Int. Conf. on Computer Vision*, 1995.
- [Sch96] C. Schmidt. *Appariement d’images pas invariants locaux de niveaux de gris. Application à l’indexation d’une base d’objets*, chap. 3 et 4. PhD thesis, INPG, 1996.
- [SH00] S. Sekhavat and J. Hermosillo. The cycab robot : a differentially flat system, 2000.
- [Ski97] S.S. Skiena. *The Algorithm Design Manual*². Springer-Verlag, 1997.
- [SML96] W.J. Schroeder, K.M. Martin, and W.E. Lorensen. The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In Roni Yagel and Gregory M. Nielson, editors, *IEEE Visualization ’96*, pages 93–100, 1996.
- [TCT98] V.S. Tsonis, K.V. Chandrinou, and P.E. Trahanias. Landmark-based navigation using projective invariants. In *Proc. of the 1998 IEEE Int. Conf. on Intelligent Robots and Systems*, 1998.
- [TOLA95] K. Tashiro, J. Ota, Y.C. Lin, and T. Arai. Design of the optimal arrangement of artificial landmarks. In *Proc. of the 1995 IEEE Int. Conf. on Robotics and Automation*, 1995.
- [Wal97] F. Wallner. *Estimation de la position d’un robot mobile par l’utilisation des composantes principales des données télémétriques laser*, chap. 3. PhD thesis, INPG, 1997.
- [Zan99] F. Zanichelli. Topological maps and robust localization for autonomous navigation. In *Online Proceedings of the Workshop on Adaptive Spatial Representations of Dynamic Environments at the Sixteenth International Joint Conference on Artificial Intelligence*, 1999.

²<http://www.tcs.mu-luebeck.de/AlgoDesignMan/BOOK/BOOK/BOOK.HTM>