

THÈSE

Pour obtenir le grade de

DOCTEUR

Spécialité : Robotique

Préparée à l'université d'Evry - Val d'Essonne, laboratoire IBISC

Présentée et soutenue par

Paul Nadrag

Le

**Faciliter la téléopération d'un robot mobile non-holonome
Application au maintien à domicile des personnes âgées**

JURY

Rapporteur : Gérard Poisson, PRISME, Université d'Orleans

Rapporteur : Guy Bourhis, LASC, Université de Metz

Examineur : Jean-Louis Ferrier, LISA-ISTIA, Université d'Angers

Examineur : Frédéric Vanderhaegen, LAMIH, Université de Valenciennes

Examineur : Philippe Hoppenot, Directeur de thèse, IBISC, Université d'Evry

Sommaire

Introduction

Téléopération avec délais

1. Introduction	9
2. Etude bibliographique	9
2.1. Historique sur la téléopération	9
2.2. Facteurs pouvant influencer la téléopération.....	10
2.3. Délais.....	13
2.4. Téléopération avec délais dans la littérature	13
2.4.1. Délais sur Internet	13
2.4.2. Aides proposées pour des applications typiques	16
3. Aide proposée.....	33
3.1. Proposition	33
3.2. Évaluations	36
3.2.3. Évaluations à IBISC	36
3.2.4. Évaluations à UDC.....	41
3.3. Conclusions sur l'utilité de l'aide évaluée.....	43
3.4. Perspectives	43

Commande en environnement inconnu

1. Description du mode de commande	46
2. Étude bibliographique	48
2.1. Évitements avancés d'obstacles	48
2.2. Génération des plans	53
2.3. Cartographie et localisation simultanées (SLAM)	57
2.4. Systèmes de planification.....	61
3. Implémentation de la commande en environnement inconnu.....	78
3.1. Architecture de commande.....	78
3.2. Résultats	84
3.3. Conclusions	86
3.4. Perspectives	87

Interface homme-machine et changement de mode de commande

1. Interface Homme-Machine.....	89
1.1. État de l'art initial	89
1.1.1. Prise en compte d'éléments de sciences cognitives.....	90
1.1.2. Autres évolutions.....	94
1.2. Approche écologique.....	96
1.2.1. Présentation	96
1.2.2. Application à la téléopération d'un robot mobile	98
1.3. Évaluations des interfaces proposées	105
1.3.1. Protocole.....	105
1.3.2. Résultats	107
1.3.3. Conclusions	111
2. Changement de mode de commande.....	111
2.1. Téléopération avec délais	111
2.2. Commande en environnement inconnu	112
2.3. Automatisation du système homme-machine	113
2.3.1. États d'automatisation d'un système homme-machine	113
2.3.2. Changement du niveau d'automatisation par la machine.....	116
2.3.3. Conséquences du changement du niveau d'automatisation	123
2.3.4. Conséquences d'une automatisation imparfaite	124
2.4. Le cas du système CompanionAble	125
2.5. Proposition	126
2.6. Perspectives	129

Conclusion et perspectives

Annexe

Bibliographie

Introduction

Cette thèse décrit mes travaux sur l'interaction homme-robot distant, au laboratoire IBISC. Le système robotique sur lequel cette thèse est centrée est un système conçu pour aider des personnes âgées vivant seules à leur domicile. La raison pour laquelle un tel public a été choisi est l'évolution démographique européenne : la moyenne d'âge, dans les pays développés, a une tendance toujours croissante. On tend vers des services spécialisés pour le maintien à domicile des personnes âgées en difficulté, la famille proche ne pouvant plus assurer seule l'aide indispensable. C'est une conséquence de la dynamique de la population : les personnes âgées ont tendance à s'éloigner des centres industriels, pendant que l'inverse est vrai pour les jeunes, qui y sont attirés par les opportunités offertes. S'il est plus facile pour un couple de personnes âgées de vivre seules, ceci est moins vrai pour une seule personne, d'autant plus si elle commence à avoir des troubles cognitifs. La solution, si la famille ne souhaite pas les prendre chez elle ou déménager à côté de la personne âgée, est de la confier à des institutions spécialisées. Mais ceci signifie un bouleversement, en fin de vie, des habitudes de la personne âgée, ce qui rend cette fin de vie plus difficile encore. Le but du projet européen CompanionAble, dans lequel j'ai été impliqué pendant la durée de ma thèse, est d'étudier comment améliorer la qualité de vie. Les solutions proposées par ce projet sont un robot mobile intégré dans une maison intelligente. Ce système est censé aider la personne dans des tâches ordinaires, telles que l'ouverture des portes automatiques, la gestion des lumières et des stores (lesquels peuvent être intelligents ou reliés à une télécommande), mais aussi l'aide au maintien des capacités cognitives de la personne par le biais d'exercices de stimulation cognitive. Ces exercices peuvent se présenter sous plusieurs formes, allant d'une présentation de type papier et crayon et jusqu'à une grande interactivité, par l'intermédiaire de l'ordinateur. Ces exercices peuvent être plus ou moins attractifs pour la personne. On espère que la personne âgée verra le robot, à la fois comme un support pour les exercices et comme un partenaire. On peut imaginer un scénario où le robot affichera sur son écran tactile les exercices et pourra utiliser ses capacités vocales (reproduction de la parole et sa reconnaissance) pour avoir une interaction plus naturelle avec la personne, par rapport à l'utilisation du support papier-crayon, ou même d'un logiciel installé sur un ordinateur de bureau, pour l'encourager ou l'informer sur l'état de ses exercices. [Nadrag 2008] montre des idées pour des exercices de stimulation cognitive en utilisant les capacités de déplacement du robot.

A part les aspects ludiques ou d'assistance quotidienne fournis par le système, la sécurité de la personne fait également partie de ses objectifs. Un incident, de type chute par exemple, peut être lourd de conséquences pour la personne âgée. Même si elle n'engendre pas de graves complications, elle peut enclencher un processus de déménagement en maison de retraite, très souvent non souhaité par la personne. L'idée serait de voir s'il était possible de détecter ces situations périlleuses, d'avertir une personne externe, qui vérifierait l'état de la personne âgée et pourrait intervenir pour l'aider (envoyer des équipes d'urgence ou un voisin sur place, par exemple). Cela permettrait de maintenir la personne chez elle. Dans cette hypothèse, le robot pourrait s'avérer très utile. Si la personne âgée est encore consciente, quand elle se retrouve dans une situation où elle aurait besoin d'aide, elle pourrait utiliser un système d'appel

d'urgence ou appeler le robot pour lui demander de l'aide. Cette aide pourrait être d'appeler un proche de la personne âgée ou les services d'urgence. Le robot pourrait détecter lui-même les situations périlleuses. En ce cas, il demanderait la confirmation à la personne âgée, pour éviter les fausses alarmes. Si, au bout d'une période fixée, aucune réponse n'est reçue, une procédure établie à l'avance est démarrée. Un exemple possible est que le robot se dirige vers la dernière position connue de la personne et qu'il appelle ensuite le centre de télégilance. Une personne du centre pourra utiliser le robot (en supposant qu'elle n'arrive pas à joindre de façon téléphonique la personne âgée) pour explorer la maison, à la recherche de la personne. Une fois trouvée, son état de santé pourra être vérifié et l'alarme confirmée ou infirmée. Pour des raisons d'intimité, il n'est pas envisageable d'installer assez de caméras dans la maison de la personne pour la couvrir entièrement. Il est possible d'utiliser les capacités de perception et de mouvement du robot à la place. Le principe devra être expliqué auparavant à la personne âgée et son consentement écrit devra être obtenu. Pendant un fonctionnement habituel du système, au cas où une personne autorisée souhaiterait utiliser le robot, le robot doit d'abord obtenir l'accord de la personne âgée. Une façon simple de faire ceci est d'avoir un bouton sur l'interface visuelle du robot. La personne âgée peut être avertie sur la demande de connexion au robot par des moyens auditifs et/ou visuels. Le robot pourrait être utilisé pour faciliter l'interaction avec les proches. Dans l'exemple qui suit, une telle situation est imaginée : la fille de la personne âgée souhaite savoir si sa journée s'est bien passée. Elle peut utiliser la visioconférence pour la contacter. Le robot s'approche de la personne âgée et l'informe que sa fille l'appelle. La personne décide de prendre l'appel sur l'écran du robot. Il est possible qu'elle souhaite changer d'endroit (s'installer sur le canapé, par exemple), ou même de pièce. Elle peut demander au robot de la suivre ou laisser sa fille le contrôler. Pour ce dernier cas, la fille pourra contrôler plus précisément la position du robot, en démarrant une session de téléopération. Pour une situation de ce type (quand il n'y pas d'alarme), il faut que la personne âgée donne son accord pour que la session de téléopération puisse commencer. Un autre cas, cette fois beaucoup plus sérieux, est quand une mésaventure arrive à la personne âgée. Une détection de cette situation alarmante est réalisée par la maison intelligente ou par le robot lui-même. Une confirmation est ensuite demandée à la personne âgée, pour éliminer tout cas de fausse alarme. Bien entendu, le système a été préalablement configuré pour réduire à un minimum les fausses détections. Si, après une certaine période, aucune réponse n'a été reçue de la part de la personne, une alarme peut être déclenchée selon la procédure préalablement établie, soit auprès d'un membre de la famille de la personne âgée, soit auprès d'un centre de surveillance. Le robot pourra, dans ce cas-ci, être téléopéré sans demander l'avis de la personne, parce qu'on a à faire à une situation exceptionnelle. Une fois l'état de santé de la personne précisé (il est possible que le robot ait été utilisé pour la chercher dans sa maison), la suite reste à établir par le téléopérateur : avertir les services de sécurité, s'assurer que la personne est capable de se soigner indépendamment (chute sans conséquences) ou d'appeler une personne de confiance.

Pour notre contexte on a identifié plusieurs spécificités : on a un environnement lointain relativement statique (la personne âgée vit seule avec éventuellement un animal de compagnie), le sol où le robot se déplace est plat et le milieu de communication utilisé est l'Internet public. L'opérateur est soit une personne de la famille, soit un membre du corps

médical. Le problème abordé dans la première partie de la thèse a, à sa source, la nature du moyen de communication employé. Comme on ne peut pas le maîtriser, on s'attend à ce que des délais de communication interviennent et perturbent la liaison entre le site maître (du téléopérateur) et esclave (du robot). Le but a été d'apporter une solution à ce problème, de faire en sorte que l'opérateur puisse utiliser le robot normalement, même en présence de délais. Ceci veut dire qu'il utilisera le robot comme s'il n'y avait pas de délai. En pratique, ceci se traduit par une commande fluide du robot, sans faire d'arrêts pour attendre un rafraichissement de l'interface avec des informations plus récentes (commande de type « move-and –wait »).

L'aide proposée dans la première partie de ma thèse est orientée vers un mode de commande manuel. Dans ce mode, l'opérateur envoie en continu des commandes au robot. Ces commandes sont des couples de vitesses linéaire et angulaire. Dans la deuxième partie de cette thèse on introduit un nouveau mode de commande. Ce mode de commande se base moins sur les habiletés de l'opérateur à piloter le robot et plus sur les capacités d'autonomie du robot. Par rapport au mode de commande utilisé dans la première partie, ce nouveau mode de commande a beaucoup moins besoin de l'intervention de l'opérateur, pour faire des corrections et ajustements de la trajectoire, dans son fonctionnement. Les commandes envoyées par l'opérateur, de façon sporadique, consistent en des couples de coordonnées x et y pour le robot (on utilise l'hypothèse de déplacement dans un monde bidimensionnel). Une fois ces coordonnées reçues, le robot est responsable de trouver un chemin vers sa destination, de le suivre et, si besoin est, d'adapter ce chemin pour ne pas se retrouver bloqué (pour sortir des impasses, par exemple). Par rapport au mode de commande manuel, ce mode de commande a deux avantages : il permet que l'opérateur soit moins concentré sur le pilotage du robot et plus concentré sur l'observation de l'environnement (ou sur une autre activité adjacente à la téléopération) et il peut être utilisé avec un délai qui ferait que, même en utilisant l'aide proposée dans la première partie, la téléopération en mode manuel du robot soit impossible.

La dernière partie de cette thèse montre une façon d'assembler les deux modes de commande du robot décrits auparavant. Une interface de pilotage du robot commune aux deux modes est d'abord proposée. Cette interface permet de piloter le robot en mode manuel ou en mode semi-automatique, sans avoir besoin de la modifier de façon radicale. La différence majeure entre les deux modes est le fait que la position future du robot est présentée d'une façon adaptée. Un autre avantage de cette interface est le fait que le champ de vue limité de la caméra est agrandi avec des informations additionnelles, en provenance du laser, au-delà de son ouverture horizontale. Le fait d'avoir plusieurs modes de commande pose un défi supplémentaire, si on souhaite changer de mode de commande pendant la même session de téléopération. La suite du dernier chapitre est dédiée à analyser les influences du changement du mode de commande sur l'opérateur et le robot. Une proposition est faite concernant la totalité des modes de commande disponibles sur le robot. Le but est d'avoir un minimum de perturbations dans le système et de maintenir les performances optimales.

Téléopération avec délais

1. Introduction

La première partie de cette thèse traite la téléopération d'un robot mobile en présence de délais. Pour que le système de maintien à domicile des personnes âgées reste accessible pour un public le plus large que possible, on a décidé d'utiliser l'Internet pour assurer la communication entre l'opérateur et le robot. Ceci signifie qu'un milieu de communication qu'on ne contrôle pas est utilisé. On s'attend donc à avoir des délais qui pourraient perturber l'opérateur s'ils ne sont pas traités par le système.

Une étude bibliographique sur cet aspect est faite en début de ce chapitre. On a voulu établir quels sont les facteurs qui pourraient perturber une session de téléopération. Ensuite, on a étudié comment la problématique des délais est prise en compte dans la littérature. Comme on envisage l'utilisation de l'Internet, on a examiné s'il serait possible de déterminer des caractéristiques pour les délais qui apparaissent pour ce type de réseau. On a également examiné les aides existantes. La partie suivante de ce chapitre présente l'aide qu'on propose. Des résultats issus de deux campagnes d'évaluations sont ensuite analysés.

2. Etude bibliographique

L'étymologie du mot « téléopération » a son origine dans le mot grec « tête », qui signifie loin, à distance, et le latin « operor », qui signifie travailler, œuvrer, accomplir. Dans la suite de cette thèse, la notion de téléopération désigne une interaction entre l'opérateur et le robot. L'opérateur envoie une consigne et le robot l'exécute. Cette consigne peut prendre plusieurs formes : une destination à atteindre, un plan (suite d'actions) à exécuter, une vitesse à atteindre, un comportement à adopter etc. Une commande de niveau bas signifie que la responsabilité du contrôle du robot repose principalement sur l'opérateur (c'est à lui d'être sûr que le robot ne causera pas des dégâts dans l'environnement lointain). Une commande de niveau élevé implique un degré élevé d'implication de la part du robot dans la réalisation de la tâche. Le robot utilise ses propres capacités pour réaliser le souhait de l'opérateur. La notion de télécommande sera utilisée en relation avec des commandes de bas niveau, transmises en continu (les ordres sont envoyés à une fréquence de quelques dizaines de hertz). La présence de l'opérateur dans la boucle de commande est requise à tout instant de la session de téléopération.

2.1. Historique sur la téléopération

Un des premiers objets téléopérés a été la torpille de Brennan (1877). Elle offrait la possibilité à un opérateur de contrôler, de façon pratique, sa direction, jusqu'à une distance de 1,8 km, avec une vitesse pouvant atteindre 27 nœuds (environ 50 km/h). Les bases de ce système étaient d'origine mécanique, les commandes de l'opérateur étant transmises par l'intermédiaire des fils. Les origines de la téléopération moderne (qui utilise un milieu de transmission électromagnétique) peuvent être considérées comme datant de la fin du 19^{ème} siècle, en étroite liaison avec le développement des communications radio. Dans son brevet n° 613809 du 8 novembre 1898, Nikola Tesla décrit la télécommande (sans fils) d'un bateau miniature. À cette époque (début du vingtième siècle), l'opérateur était en contact visuel direct

avec la partie esclave du système. Actuellement, on a la possibilité d'avoir des distances beaucoup plus importantes entre les deux sites (les sondes spatiales Voyager 1 et 2 se trouvent à environ 15 milliard kilomètres de la Terre).

Au début du 21^{ème} siècle, la téléopération trouve ses applications pratiques le plus souvent dans les tâches dites DDD (Dull, Dangerous and Dirty – monotones, dangereuses et sales). La robotique spatiale se remarque par les moyens mis en jeu depuis un bon nombre d'années. Le robot Lunokhod 1 a été déposé sur la Lune en 1970 et télé-opéré depuis la Terre pendant 322 jours. On peut également rappeler les rovers martiens Spirit et Opportunity [Rover 2011]. D'autres domaines dangereux sont les réparations dans des milieux hostiles (l'espace, les profondeurs des océans, les centrales nucléaires), des opérations de déminage, des opérations de recherche et sauvetage (à l'intérieur des immeubles en ruines à la suite d'un tremblement de terre ou une explosion) [Burke 2004], dans le domaine médical, pour faire des échographies (le système OTELO [Courreges 2008]) ou des interventions chirurgicales (les systèmes ZEUS [ZEUS 2000] et da Vinci [Vinci 2010]).

La tendance générale est, à cause de différentes contraintes, de s'éloigner d'une téléopération pure (basée sur une philosophie maître-esclave, où la partie esclave n'est munie d'aucune intelligence et exécute « à la lettre » chaque ordre reçu) et de se diriger vers un système coopératif homme-machine, où la machine est munie d'un certain niveau d'intelligence qui lui permet de partager des tâches avec l'opérateur humain.

L'exemple le plus connu par le grand public est celui des drones aériens de type MQ-1 Predator, utilisés par l'armée des États-Unis. La partie humaine consiste dans une équipe d'opérateurs, chacun avec des tâches précises [NATO 2007]. La machine peut, par exemple, déterminer et suivre un chemin vers une destination dont les coordonnées lui ont été indiquées par la partie humaine ou accomplir toute seule un atterrissage ou un décollage.

Dans le domaine du nucléaire, des robots, munis [Cybernetix 2011] ou non [Hanford 2011] de bras manipulateurs sont utilisés pour des opérations variant des inspections au démantèlement nucléaire.

Comme on peut le constater, les systèmes décrits auparavant ont un cadre d'application contraignant. Soit il s'agit d'un canal de communication dédié (filaire ou pas), soit le rayon est limité. La problématique du délai soit n'est pas prise en compte, parce que les conditions d'utilisation ne sont pas favorables à son apparition, soit elle est intégrée dans la façon dont le robot est contrôlé (envoi des plans de mouvements à la place de commandes discrètes), si les délais sont trop importants.

2.2. Facteurs pouvant influencer la téléopération

Le moyen le plus commun de se connecter à l'Internet est, en Europe, par l'intermédiaire d'une liaison de type ADSL, fibre optique ou câble (type télévision). Comme il s'agit d'une connexion qui n'est pas dédiée et que l'opérateur peut se trouver à une distance importante du robot, on examinera les facteurs susceptibles de perturber la télécommande du robot. Il faut garder à l'esprit le contexte général, qui est d'avoir le robot dans une habitation ou à l'intérieur d'un bâtiment.

Pour les raisons présentées ci-dessus, on a fait le choix de concentrer nos investigations sur les possibilités offertes par l'Internet publique. Il est de plus en plus répandu, avec une couverture de 68,9% pour la France [Internetworldstats 2011] et pouvant atteindre 81,1% pour des pays comme la République de Corée [Internetworldstats 2011]. Pour l'instant, les fournisseurs d'accès Internet sont agnostiques¹ par rapport au contenu des données qu'ils transportent. Même s'il serait souhaitable de pouvoir modifier la priorité de certaines données (visioconférence ou jeux en réseau) par rapport aux autres (transferts de fichiers, vidéos en ligne, chargements de pages web) qui n'ont pas les mêmes contraintes de temps réel, ceci n'est pas encore possible avec l'architecture couramment utilisée. Il est possible qu'un changement ait lieu avec le passage à IPv6 (les adresses IPv4 étant épuisées), où il sera possible de gérer les priorités de différents flux de données.

Le principal problème qui sera rencontré est le délai. Ce délai est principalement dû à deux causes : la distance (physique) entre les deux parties et la congestion du réseau.

Il n'est pas possible de réduire le délai dû à la distance entre les deux postes (et au temps de traitement lié au passage à travers différents routeurs). Il est possible de faire des mesures avant de commencer la session de téléopération pour avoir une idée approximative sur la valeur du délai, mais il est possible que, pendant la même session, des modifications interviennent qui font que le chemin jusqu'au moment du transfert des données change.

Les problèmes posés par la congestion du réseau peuvent générer des nuisances légères mais aussi importantes. Il est possible d'observer un pic dans le délai moyen ou d'avoir un temps trop long entre les deux parties, qui engendrera l'impossibilité de la session de téléopération (time out). D'autres problèmes comme l'arrivée désordonnée des paquets de données ou la perte ponctuelle de paquets existent aussi.

Il est possible d'avoir des prévisions sur l'évolution de la congestion (on peut supposer, par exemple, qu'elle sera plus importante le matin, quand les gens arrivent au travail, et le soir, quand ils rentrent à la maison), mais on ne peut pas avoir des indications avec un grand degré de précision. Il est possible pour certains paquets d'être dirigés par des chemins plus longs (en traversant d'autres continents, par exemple), mais ces liaisons ont une capacité limitée, qui est vite saturée si trop de paquets y sont redirigés.

Un troisième facteur influençant, à une moindre mesure que la qualité du réseau, est le matériel utilisé pour le poste de commande et le robot. Une étude empirique [behardware 2006] a trouvé que le temps écoulé entre le moment où l'opérateur appuie sur un bouton et l'affichage de cet événement à l'écran est, avec une un système de test légèrement en dessus de la moyenne, de 65 ms. Avec une configuration pas du tout adaptée, il peut atteindre 110 ms ! Cette valeur est interne au système homme - interface de contrôle, le délai entre le site de l'opérateur et celui du robot vient se rajouter dessus. Ceci peut faire qu'une situation acceptable se transforme dans une expérience frustrante pour l'opérateur.

¹ (Informatique) Qui peut s'adapter aux composants avec lesquels il interagit. Exemple: Le « socket » (lien, connexion) de la couche de communication est agnostique en ce qui concerne les mécanismes de transports sous-jacents - il est « transport-agnostique ».

Pour des tâches qui demandent des réponses très rapides, des études ont trouvé que les performances de l'opérateur humain commencent à diminuer si le délai dépasse 180 ms [Armitage 2003] ou 250 ms [Henderson 2001]. Ceci ne veut pas dire que l'opérateur ne pourra pas accomplir ses tâches, mais uniquement que ses performances seront dégradées, en tendant vers zéro. [Simpson 2007] a trouvé que, même avec un délai de 1,5 s les opérateurs étaient capables d'accomplir leurs tâches (optimisation des paramètres d'une poutre).

Le délai maximal qui n'empêchera pas le fonctionnement normal du système téléopéré n'est pas fixe. Il dépend d'une multitude de facteurs. La tâche à accomplir est un élément important. Avec le même robot, les contraintes seront différentes s'il faut naviguer dans un environnement ou chercher un certain objet. Les caractéristiques du robot et de l'environnement influenceront aussi sa valeur. Un robot capable de mouvements rapides dans un environnement où les obstacles sont près les uns des autres n'aura pas les mêmes contraintes qu'un robot lent, dans un environnement où les obstacles sont espacés. Un facteur non négligeable est l'opérateur humain lui-même et ses capacités. S'il a une expérience suffisante du système et de son environnement, il est capable de se former un modèle mental et de l'utiliser pour anticiper l'état futur du robot, obtenant ainsi une marge plus large pour le délai. Si le délai dépasse cette valeur limite, la réaction la plus fréquente des opérateurs est d'adopter le comportement nommé « move-and-wait » (avance et attends) [Vanijjirattikhan 2008]. L'opérateur fera avancer le robot sur une petite distance pour laquelle il est sûr de lui et attendra le rafraîchissement des données provenant du robot. C'est uniquement après avoir reçu ce rafraîchissement (qui lui confirmera ou pas son hypothèse sur la nouvelle position du robot dans l'environnement) qu'il enverra une nouvelle commande au robot.

Ces problèmes, que l'Internet pourrait poser à l'opérateur, et la possible importance de la tâche montrent l'intérêt de diminuer leur influence. Pour des tâches de positionnement du robot (en vue d'une visioconférence, par exemple), la rapidité avec laquelle la tâche est accomplie n'est pas primordiale. Si la session de télécommande est initiée à la suite d'une alarme générée par la chute de la personne, la rapidité avec laquelle la tâche est accomplie est extrêmement importante, la vie de la personne âgée pouvant en dépendre. Le confort de l'opérateur est aussi affecté par ces facteurs, celui-ci pouvant être gêné ou même empêché dans sa tâche. Aider l'opérateur n'est pas indispensable pour qu'il réussisse sa tâche (avec plus ou moins de peine), mais en le faisant, son comportement se rapproche d'un comportement similaire à un comportement en absence de délais et il est plus facile également d'assurer la sécurité de la personne âgée.

2.3. Délais

Pour savoir quelle assistance accorder à l'opérateur, il faut savoir le délai existant entre le robot et le site maître, pour l'adapter en fonction de celui-ci. En ce qui concerne la valeur du délai, à la suite des différentes références bibliographiques trouvées, on a décidé de le classer en trois groupes :

1. Imperceptible
2. Moyen
3. Grand

On a considéré que, si le délai de communication était inférieur à 150 ms, alors il ne serait pas perçu comme dérangeant par l'opérateur. Cette limite peut varier en fonction du contexte (la tâche à réaliser) et des capacités de l'opérateur. À titre d'exemple, [Armitage 2003] trouve une valeur minimale de 180 ms pour le cas des jeux vidéo (jeu de tir à la première personne). Les spécifications du notre robot (vitesse linéaire maximale de 1,2 m/s et angulaire de 0,5 rad/s) et la tâche à effectuer (navigation dans une résidence et recherche d'une personne) induisent le fait qu'on n'a pas besoin de réactions instantanées de la part de l'opérateur (mais il doit se montrer assez réactif, quand même).

Le délai moyen a été fixé comme la zone entre 300 ms et 2 s. Pour cette zone, les opérateurs sont obligés d'adopter le comportement « move-and-wait ». Une aide a été proposée pour le cas d'un délai a une valeur se situant en cette zone, qui a comme but d'éliminer le comportement « move-and-wait », afin que le contrôle du robot se déroule de façon continue. Cette aide sera décrite de façon très détaillé dans le chapitre 3.

Pour le troisième groupe, le délai a une valeur supérieure à deux secondes et il n'est plus possible de piloter le robot en continu. Soit on choisit un autre mode de commande que la télécommande manuelle, soit on considère la connexion comme étant interrompue.

2.4. Téléopération avec délais dans la littérature

Le premier problème que j'ai abordé pendant ma thèse a été la téléopération avec délais. Pour rappel, notre cas est assez spécifique : on a un robot mobile, non-holonome, à contrôler à l'intérieur d'une maison où une personne âgée habite seule. Pour des raisons pratiques (coût, existence du réseau, etc.), on a choisi d'utiliser l'Internet comme milieu de communication entre le robot et l'opérateur.

La partie suivante (2.4.1) fait l'état des délais qu'on s'attend à retrouver sur Internet, leur forme générale et se penche aussi sur la synchronisation précise des horloges des ordinateurs. Les parties suivantes (2.4.2) décrivent des aides utilisées pour combattre les effets des délais sur la téléopération. À la fin de cette section on montre l'utilisation des fenêtres prédictives et décrit la réalisation de notre fenêtre, utilisée pendant les évaluations décrites dans le chapitre suivant (3.2).

2.4.1. Délais sur Internet

Une étude visant l'acquisition des informations concernant les caractéristiques de l'Internet a été menée en début de ma thèse, pour déterminer la possibilité d'identifier des

caractéristiques. Des informations concernant la forme de délais ou leur modélisation conduiraient vers la réduction des voies de recherche pour savoir comment les limiter leurs effets.

Des séries de mesures de paquets de données ont été analysées par [Iwama 2004]. Les auteurs ont voulu caractériser le délai unidirectionnel. Pour assurer la précision du système, un équipement très sensible a été utilisé (des horloges à césium). Les paquets de données ont été aussi modifiés après leur création, pour s'assurer que le moment de l'émission de la trame est bon (il pourrait être faussé à cause de la dérive de l'horloge interne de l'ordinateur qui a créé le paquet). La précision ainsi obtenue est de l'ordre de dizaine de nanosecondes. Les résultats obtenus montrent une dépendance linéaire du délai de la taille des paquets (avec une pente d'environ 5,3 Mbps pour une liaison de type 10 Mbps agglomérée et 190 Mbps pour une liaison de type 10 Gbps qui n'est pas sollicitée par le trafic).

Des mesures empiriques du délai Internet ont été faite dans [Garcia 2003]. Deux exemples sont reproduits sur la Figure 1 et la Figure 2.

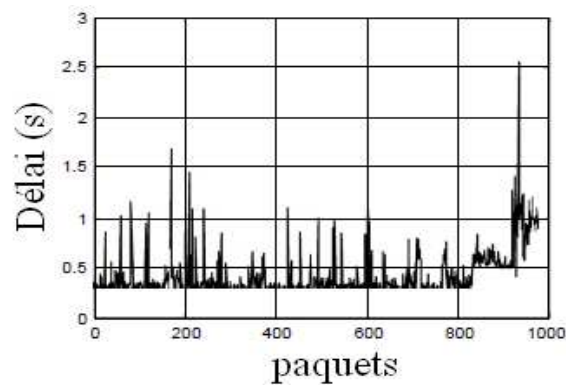


Figure 1 Délais samedi après-midi [Garcia 2003]

Dans la Figure 1, les délais ont été mesurés pendant une période de trafic bas, entre les villes de San Juan (Argentine) et Madrid.. On observe un délai constant, qui a une valeur de 250 ms environ, et des pics, qui apparaissent avec une fréquence variable. La fréquence de réception des paquets de données n'est pas précisée.

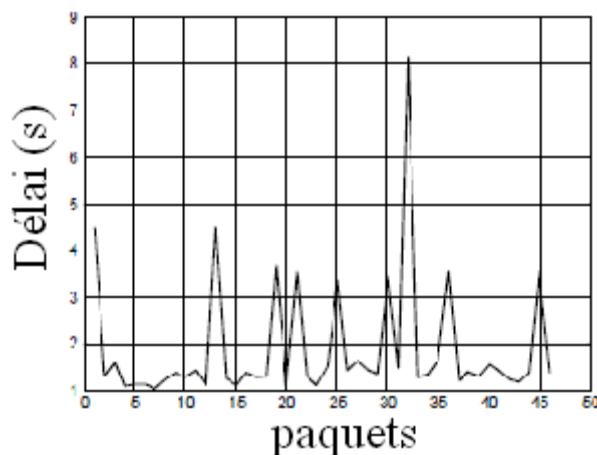


Figure 2 Délais lundi matin [Garcia 2003]

Dans la Figure 2, la période choisie pour effectuer les enregistrements est une période de pointe (lundi matin), quand le trafic est important (les villes impliquées étaient les mêmes). La cause de ceci est la reprise du travail par les gens. Cette fois-ci, la partie constante du délai dépasse une seconde (1300 ms environ), et le caractère variable des pics est toujours présent. La fréquence de réception des paquets de données n'est pas précisée.

Une autre série de mesures est décrite dans [Moon 2000]. Un exemple est reproduit dans la Figure 3. Une remarque est également faite sur la proportion de paquets qui sont arrivés en désordre à leur destination. Faisant référence à une autre étude [Paxson 1997], qui a trouvé que moins d'un pour cent des paquets arrivaient en désordre, les données examinées ne témoignent que de moins de 0,1% de paquets qui sont arrivés en désordre.

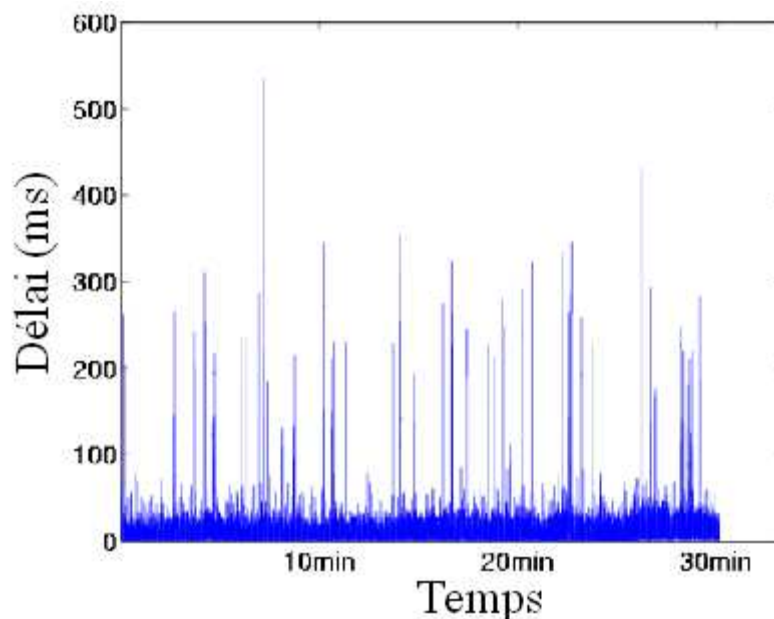


Figure 3 Délais typiques

Une observation commune sur l'ensemble de ces papiers est que les données provenant de la capture de paquets ont la caractéristique d'avoir une partie constante, comprise entre 40 ms et 1,2 s, en fonction de contexte, qui est occasionnellement perturbée par des pics.

Si on se concentre sur la partie constante du délai, il est possible d'envisager d'avoir sa valeur en temps réel, à condition que les horloges internes du site maître et du site esclave soient synchronisées. Une solution peut être l'utilisation du NTP (Network Time Protocol). Il s'agit d'un protocole largement répandu, utilisé pour la synchronisation des horloges des ordinateurs connectés à Internet. L'idée centrale est de prendre quelques mesures concernant le délai présent entre l'ordinateur souhaitant se synchroniser et le serveur de temps (une machine reliée à une horloge très précise, par exemple de type atomique). Les questions qui se sont posées sont sur la déviation postérieure de la synchronisation et quelle devrait être la fréquence de resynchronisation.

De [Mills 1989] on apprend que, pour NTP v2, la précision de la synchronisation est de l'ordre de 10 ms. Le réseau utilisé est un réseau filaire.

Comme le robot doit pouvoir se déplacer, il devra utiliser une connexion sans fil à un point d'accès Internet. Une telle connexion est utilisée aussi par [Elson 2002]. Cette fois, la version 4 de NTP est examinée. La conclusion du papier est que la précision de la synchronisation est affectée par le trafic existant sur le réseau. Si le trafic est minimal, la précision est très bonne, avec une variation de 50 microsecondes. Si le trafic est important (6,5 Mbps dans un réseau ayant une capacité théorique de 11 Mbps – un réseau de type 802.11), alors la précision sera moins bonne, la variation étant de deux millisecondes.

Une étude plus élargie est décrite dans [Mills 1997]. Elle a porté sur 180.000 machines dans le monde. On a des observations sur la synchronisation des horloges et aussi sur le délai entre la machine de test et les autres. Les résultats témoignent du fait que la majorité des PC ont une marge de synchronisation de 21 ms, pendant que la moyenne sur l'ensemble du test est de 29 ms. Concernant le délai, le délai moyen a été de 186 ms, et le délai médian est de 118 ms. Le délai maximal constaté a été de 1,9 secondes. L'importante valeur maximale a été attribuée, par les auteurs, à la congestion du réseau. La forme du délai était, généralement, symétrique. L'asymétrie n'a jamais dépassé 10% du délai total, même pour des liaisons transatlantiques.

Des tentatives de créer un modèle pour décrire la dynamique des délais sont décrites en [Morita 2002]. Le modèle utilisé est de type auto régressif exogène. Pour des cas où il n'y a pas un fort trafic sur le réseau, le modèle proposé donne des bons résultats. Par contre, le modèle n'est pas capable de prédire le comportement du système quand les congestions deviennent fréquentes.

La conclusion qu'on peut tirer de cette rapide étude bibliographique sont que le délai sur Internet est principalement dû à la composante physique (la distance entre le poste maître et le poste esclave). Cette composante est à peu près constante dans le temps. Il est possible de rencontrer des pics de délai, principalement dus à la congestion du réseau (trafic proche de sa capacité maximale). Si on n'a pas une connaissance globale du réseau (architecture et clients connectés), il est impossible d'anticiper les pics avec précision (il est quand même possible d'identifier, suite à des prospections de longue durée, des périodes plus susceptibles d'avoir un volume de trafic plus important que d'autres).

NTP (protocole pour le temps réseau) peut être utilisé pour synchroniser les horloges des postes maîtres (de l'opérateur) et esclaves (du robot), avec une marge de 20 ms, largement suffisante pour notre cas (la téléopération manuelle non assistée peut s'effectuer de façon continue pour des délais jusqu'à environ 300 ms).

2.4.2. Aides proposées pour des applications typiques

Dans les parties suivantes, différentes modalités pour aider l'opérateur à accomplir sa tâche seront présentées. Dans la section 2.4.2.1 la problématique de la téléopération haptique est analysée. Comme elle est la plus sensible aux délais, on a commencé avec elle, pour examiner des situations dans lesquelles offrir de l'aide est très important. La problématique est abordée sous une autre optique dans 2.4.2.2, où le robot et l'opérateur forment une équipe, et s'entraident. Différentes modalités pour enrichir le retour du robot sont présentées dans les sections 2.4.2.3 et 2.4.2.4. L'utilisation de la réalité virtuelle ou augmentée permet de faire

une prédiction de la scène distante et un affichage qui aidera l'utilisateur dans sa tâche, avec un retour instantané de ses commandes.

2.4.2.1. Téléopération haptique

Parmi les applications de la téléopération on peut citer les tâches de palpation. Un bras est généralement attaché à une base rigide. Il est utilisé pour « toucher » l'environnement et retourner cette information à l'opérateur. Pour des applications demandant un contact avec l'environnement (comme le domaine médical), il est indispensable d'avoir une idée précise de la force qu'on applique sur l'environnement.

Une solution possible à ce problème a été apportée par Günter Niemeyer dans sa thèse [Niemeyer 1996]. L'utilisation des variables d'ondes y est exposée. Il décrit la possibilité de compenser le délai par une augmentation de la résistance que le dispositif haptique présente à l'opérateur. D'une façon pratique, sa conclusion est que, en utilisant sa méthode, les délais ne sont plus compensables au-delà de quelques secondes. Il estime que cette méthode est parfaitement adaptée à un délai de 0,5 s. Même si cette valeur peut paraître petite par rapport au délai maximal qu'on envisage (deux secondes), il faut souligner le fait que, par rapport au retour visuel, le retour haptique est beaucoup plus sensible au délai. L'information retournée par la vision est une information complexe, de haut niveau, qui nécessite beaucoup de traitement par le système nerveux (dans le cortex visuel) avant de pouvoir être comprise par l'opérateur. En ce qui concerne l'information haptique, son traitement s'effectue de façon plus rapide. De ce fait, l'opérateur humain est plus sensible au décalage entre sa commande et la réponse attendue dans ce mode qu'en mode d'utilisation purement basé sur la vidéo. De façon générale, le retour haptique supporte un délai de 0,1 s [Niemeyer 1996], et le retour vidéo a son seuil à 0,3 s ([Armitage 2003], [Henderson 2001]). Cependant, il ne faut pas garder ces valeurs comme des valeurs absolues, mais comme des indicateurs, étant donné qu'elles sont liées aux personnes examinées et aux tâches à effectuer.

[Arioui 2003] analyse le problème du rendu haptique dans un environnement virtuel. La tâche est de toucher un mur. Le contrôleur proposé peut être utilisé avec un délai constant et un délai variable. Seule la valeur du délai dans un sens (du maître à l'esclave) est supposée connue. Des tests avec un délai constant (maître-esclave = 1s et esclave-maître = 0,5s) et un délai variable (entre 0,5 et 1,5 s) sont présentés. Par rapport à la méthode basée sur les variables d'onde, cette méthode reproduit mieux la valeur de la force. Son désavantage réside dans l'écart qui peut apparaître entre la position du bras maître et celle du bras esclave.

Dans [Yokokohji 2001] une expérience plus extrême a été menée. Le délai a été fixé à 6 s. Le poste maître était sur la Terre et le poste esclave (le bras robotique) se trouvait à l'intérieur du satellite ETS-VII. Le but de l'expérience n'était pas de vérifier si le contrôle s'effectuerait comme s'il n'y avait pas de délai, mais d'examiner l'apport de l'aide (même imparfaite), basée sur un régulateur bilatéral proportionnel-dérivé. Trois conditions ont été examinées : retour de l'information haptique et visuelle sur l'effort généré par la partie esclave sur son environnement, retour de l'information haptique uniquement ou retour de l'information visuelle exclusivement. La conclusion de l'article est que le retour haptique, seul ou en combinaison avec une indication visuelle concernant la valeur de la force, aide à augmenter la

performance de la tâche (reconnaitre la forme d'un objet ou introduire une cheville dans un trou).

La même expérience (tâche à réaliser) est reprise en [Yoon 2001]. Cette fois, on n'a plus un retour vidéo direct, mais une représentation en réalité virtuelle de la scène spatiale. Une représentation opaque est utilisée pour indiquer la dernière position connue du bras (pour laquelle des données de télémétrie ont été reçues) et la prédiction de la position du bras est réalisée en utilisant une représentation en fils de fer. La composante graphique pour cette expérience est, par rapport à la précédente, beaucoup plus complexe. Cette fois, des erreurs de ± 10 mm dans le positionnement du bout du bras sont introduites à l'affichage. L'opérateur est informé de l'existence de ces erreurs, mais il n'est pas informé de leur valeur. Le suivi de la forme d'un objet n'est pas perturbé par les erreurs. L'introduction d'une cheville dans un trou devient beaucoup plus compliquée, mais les opérateurs sont capables de la réussir également. La conclusion des auteurs est que, même en présence d'erreurs de modélisation, l'opérateur est capable (avec quelques difficultés) de réaliser des tâches comme le suivi de contours ou l'introduction d'une cheville dans un trou.

Un système d'échographie est décrit dans [Courreges 2008]. La liaison entre le site maître et le site esclave est soit de type ISDN, soit de type satellite. Le système fonctionne en boucle ouverte entre le site maître et le site esclave. Sur le site esclave, il fonctionne en boucle fermée. Cette approche permet de résoudre le problème de l'instabilité, due aux délais. Des tests réalisés en Espagne et en France sur différents types de problème de santé ont mis en évidence les points forts et faibles du système. L'identification des lésions a été particulièrement difficile. En moyenne, le système a eu à 80% la performance d'un système d'échographie où l'opérateur se trouve à côté du patient.

[Hirche 2003] présente le résultat des travaux dont le but est de se diriger vers la téléprésence². Le bras robotique utilisé a sept degrés de liberté. Les expériences se sont concentrées sur des tâches de contact, dans une seule direction (avant/arrière). Le délai initial était de 10 ms, ensuite il variait entre cette valeur et 100 ms. L'architecture de contrôle se base sur une approche combinant la vitesse avec la position et la force. Elle a été comparée à une architecture basée uniquement sur la vitesse et la force. L'architecture de contrôle proposée permet d'appliquer des forces importantes sur l'environnement. Cette architecture montre une certaine rigidité dans le poste de commande. Le suivi de la position du bras robotique est meilleur que celui obtenu avec l'architecture de contrôle basé uniquement sur la vitesse et la force.

Des études ont été aussi menées dans le domaine du travail télé-collaboratif [Chong 2003]. L'environnement du site-esclave est un environnement industriel. La tâche que les téléopérateurs doivent accomplir est de dévisser et retirer un couvercle de conduite. Le couvercle est trop lourd pour qu'un seul robot puisse le manipuler. Les postes de commande utilisés par les opérateurs sont complexes : quatre images caméra sont affichées simultanément, ainsi qu'une représentation en réalité virtuelle de la scène. Le dispositif de contrôle peut être déplacé sur

² Technique qui permet à une personne d'avoir l'impression d'être présente, de donner l'impression d'être présente, ou d'avoir un effet à un endroit autre que son emplacement réel.

six axes. Six caméras (fixes) sont présentes sur du côté des bras robotiques, situés à 90 km du site de commande. Sur ses écrans, l'opérateur peut changer de caméra. Le délai constaté a été de 66 ms pour les données de commande et 1 s pour la vidéo.

Afin d'aider l'opérateur dans sa tâche, en plus de la représentation en réalité virtuelle de la scène (qui se fait sans délai), des champs de forces virtuels ont été mis en place pour éviter que les deux bras robotiques entrent en contact. Des avertissements sonores sont aussi utilisés pour alerter l'opérateur de l'existence d'une situation dangereuse. En utilisant uniquement le retour des caméras, le temps moyen pour la réalisation de la tâche était de 5 minutes. En utilisant l'affichage prédictif, le temps moyen pour la réalisation de la tâche diminuait à 4 minutes.

2.4.2.2. Répartition de tâches

Une façon de contourner le problème de délais est de séparer la tâche de navigation en deux sous-tâches : planification et exécution. L'opérateur sera, en ce cas, responsable de la partie planification, pendant que le robot aura la responsabilité de la partie exécution (implémentation de la planification de l'opérateur). [Simmons 2002] décrit un robot commandé par Internet, qui fonctionne de cette façon. L'opérateur peut lui indiquer la pièce (dans un bâtiment) où il faut qu'il aille et une action à exécuter (un enregistrement vocal à reproduire). Le robot planifie le chemin et le suit en évitant d'obstacles.

Dans [Wang 2005] la même idée est utilisée. Deux modes de commande sont proposés à l'utilisateur : un mode manuel (contrôle direct) et un mode avancé (points à atteindre ou comportements à adopter – errer dans l'environnement, par exemple). La communication entre le robot et l'opérateur humain est effectuée via Internet. L'environnement du robot est l'intérieur d'une maison. Il peut être connu à l'avance ou pas. Si le robot se retrouve dans l'impossibilité de continuer d'avancer (la consigne reçue est d'aller dans une autre pièce et la porte d'accès entre les deux, le seul chemin disponible, est fermée) il peut demander l'aide de l'opérateur afin de pouvoir réaliser la tâche qui lui a été assignée. Les auteurs soulignent une faiblesse des modes de commande avancés : la possibilité que l'opérateur commence à s'ennuyer est liée au temps passé à piloter le robot dans le mode de commande avancé, surtout si le robot met beaucoup de temps à exécuter l'ordre (soit à cause d'une grande distance à parcourir, soit à cause des ressources insuffisantes pour la planification et le suivi de la trajectoire).

Dans [Zigoris 2003] l'opérateur n'est pas chargé de commander un seul robot, mais une équipe entière (six robots). La possibilité d'un contrôle complètement manuel existe, mais l'opérateur a aussi la possibilité d'envoyer des ordres avancés à ses robots. Les délais ne sont pas grands, mais les auteurs observent que, à cause du rythme du jeu, leur valeur était suffisante pour affecter le score final. Ces ordres sont de type « protéger une zone », « intercepter un robot adverse », etc. A cause de la taille réduite des robots et de leur vitesse (2 m/s environ), la télécommande manuelle n'est pas souvent utilisée, en faveur du mode de commande de haut niveau. L'interface graphique est également adaptée au contexte, rappelant plutôt des jeux de stratégie en temps réel.

La même démarche est décrite dans [Vanijirattikhan 2008]. Les auteurs partent de l'observation que la téléopération basique, en présence de délais importants, induit un comportement de type « move and wait », ceux qu'ils veulent éviter. La solution qu'ils avancent est de faire appel à la « téléprogrammation » (teleprogramming, en Anglais). Ceci revient à envoyer des ordres de haut niveau au robot, qui sera ensuite capable de les interpréter et de les implémenter (mettre en application de façon effective). Un exemple d'ordre est une destination à atteindre, spécifiée par ses coordonnées cartésiennes.

[Munasinghe 2006] décrit une séparation des tâches : l'opérateur contrôle une partie du système, le robot en contrôle une autre. Le robot qui sera manipulé est de type bras, situé sur une base fixe. Le but du système est de frapper une balle de golf avec une crosse, située au bout du bras robotique. Ce n'est pas pour des applications ludiques, mais uniquement pour tester la faisabilité du système. Le retour visuel de l'opérateur est assuré par une caméra fixe, située dans un coin de la pièce où se trouve le système. Ainsi, il peut arriver que dans certaines positions la balle ne soit pas visible. L'utilisateur contrôle, non pas de façon continue, mais pas à pas, le déplacement (positionnement) du bras. Simultanément, le bout du bras (la position de la crosse) est contrôlé par le site esclave (le robot), afin qu'il soit toujours tourné vers la balle. L'amplitude des mouvements permis à l'opérateur est modifiée en fonction du délai (plus le délai est petit, plus la distance maximale continue sur laquelle l'opérateur peut contrôler le bras est augmentée). Pour contrecarrer le manque d'information dû à l'affichage à l'écran (2D – un affaiblissement des informations sur la profondeur), un pointeur laser est monté sur le bras robotique, à côté de la crosse. Le laser est orienté perpendiculairement au sol, afin d'indiquer la projection du bras dans son plan. Ainsi, l'opérateur aura une information supplémentaire sur la position du bras par rapport à la balle, quand il voudra la frapper.

2.4.2.3. Utilisation de la réalité virtuelle sur l'interface de contrôle

Le robot utilisé dans [Monferrer 2002] est un robot sous-marin. Le but de l'interface est de permettre l'accès simultané de plusieurs personnes au robot. Le robot est muni de deux bras. Il est possible de permettre un contrôle simultané du robot (manipulation avec les bras et déplacement du robot, par exemple). Les avantages de ce type d'interface, pour les auteurs, sont les suivants : possibilité d'utiliser des aides visuelles, personnalisation de l'affichage des données, présentation attractive des données et redondance de l'affichage. Il est possible de rajouter et de partager des points d'intérêt sur l'interface. Pour maintenir une taille réduite de ces points sur l'interface de contrôle, ils sont rajoutés comme des éléments du monde physique (même s'ils n'ont pas d'équivalent réel). Pour faciliter la compréhension de leurs positions, des ombres de ces points sur le fond et la surface de l'eau sont calculées et affichées. La synchronisation des informations, à cause des délais, entre les différents sites des opérateurs, est mentionnée comme un souci important. Deux exemples d'interfaces sont montrés : une interface orientée vers la visualisation des données (Figure 4) et une autre pour le contrôle du robot (Figure 5).

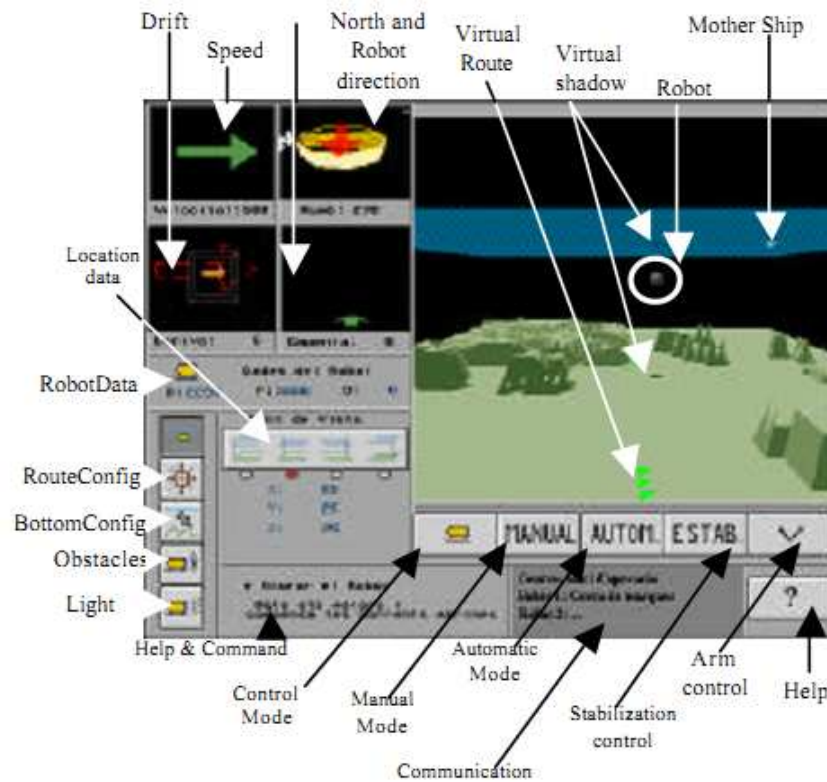


Figure 4 Interface orientée visualisation données

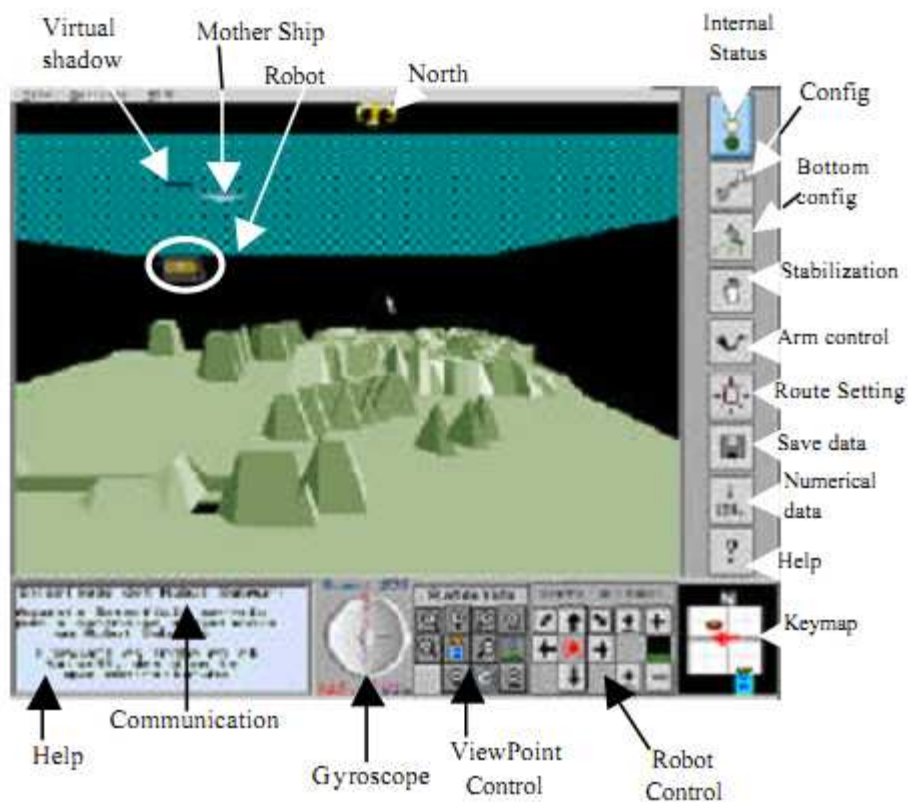


Figure 5 Interface orientée contrôle

Dans [Lane 2001] on retrouve une proposition de robot proche de Robonaut [Bluethmann 2003] (un robot anthropomorphe). En connaissant l'environnement du robot et sachant qu'il

sera le seul qui agira sur son environnement, les auteurs proposent de faire une modélisation en réalité virtuelle de l'environnement. La représentation de l'environnement à partir des dernières données reçues est faite avec des couleurs opaques et la prédiction de la position du bras est faite avec des couleurs transparentes. Des délais jusqu'à trois secondes sont considérés. Le principal souci découvert pendant les manipulations du système est que, même s'il marche bien pour des délais importants et/ou mouvements amples, pour des petits mouvements et/ou délais, l'aide à tendance à être confondue avec le rendu des informations les plus récentes reçues du site esclave, ce qui complique l'évaluation de la position réelle du bras robotique. D'autres problèmes peuvent être engendrés par le fait que les capteurs qui indiquent la position des divers composants du robot ne sont pas parfaits et il est possible qu'ils induisent des erreurs qui vont se cumuler dans le temps et qui vont finir par induire un décalage important entre la position réelle du robot et celle lue à partir de ses capteurs.

Les travaux décrits dans [Han 2001] sont pensés en vue de leur utilisation avec un robot mobile personnel, auquel on se connecte par l'intermédiaire de l'Internet. Des mesures effectuées dans le même domaine trouvent une valeur moyenne de 20 ms. Dans le même pays, la valeur moyenne trouvée est de 4,2 s. En utilisant un modèle du robot, il est possible d'avoir une représentation schématique de son emplacement dans la maison. Il s'agit d'un plan très grossier (vue de haut).

[Bares 1997] est issu d'une série d'expériences sur le terrain. Le robot est utilisé pour explorer un volcan actif. Il est dépendant d'un câble (pour l'alimentation et sûreté) et utilise une connexion par satellite au poste de commande (délai moyen de 4 s), puisqu'une partie de ses capacités de calcul s'y trouvent. Il a des capacités semi-autonomes : l'objectif est l'exploration du volcan. Pour les portions les plus simples du terrain, le robot descend seul (supervisé par l'opérateur humain), pendant que dans les zones plus difficiles, il est télécommandé. L'environnement n'est pas modélisé à l'avance, mais généré au fur et à mesure que le robot avance, en fonction des données qu'il envoie. La représentation graphique des données de télémétrie est très utile pour l'opérateur : ça l'aide non seulement à évaluer la distance du robot par rapport aux obstacles présents dans l'environnement, mais aussi à visualiser de façon rapide la topologie du terrain, comme on peut le constater dans la Figure 6.

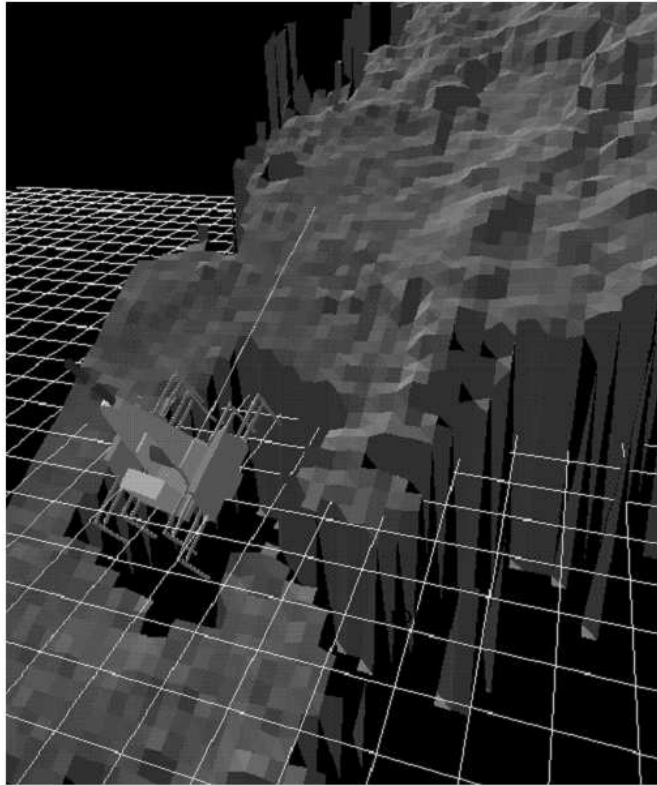


Figure 6 Représentation en réalité virtuelle du relief (la grille représente le plan horizontal)

Dans [Escribano 2004] le robot utilisé est un bras robotique, situé sur une base fixe. Quelques valeurs-type pour le délai sont avancées. Par exemple, pour une connexion entre deux villes, situées à 150 km l'une de l'autre, le délai moyen est de 17,2 ms, avec un écart-type important (9.74 ms) et des pertes négligeables (0.8%). Pour une connexion intercontinentale (10.000 km de distance), le délai moyen est de 326,3 ms, avec un écart-type de 27,2 ms, mais avec des pertes importantes (41,4%). Le robot et son environnement sont modélisés en réalité virtuelle, auquel on vient rajouter l'ombre du bras robotique sur son plan de travail. Les auteurs avaient trouvé que la présentation de l'ombre, même en vision 2D (sur l'écran) aide à obtenir de meilleures performances (moins de collisions et de mouvements en arrière du bras et un temps inférieur) qu'en utilisant un environnement virtuel stéréoscopique (3D).

Dans [Schulz 2000] les auteurs ont profité des contraintes de l'environnement (l'intérieur des bâtiments) pour créer à l'avance un plan 3D (en représentation interne, non pas à l'affichage). La connexion au robot est réalisée par l'intermédiaire de l'Internet. Comme il est possible que l'état du monde change pendant le déplacement du robot, l'environnement est évalué de façon continue et son modèle actualisé, si c'est le cas. Cette démarche est concentrée sur les portes (spécialement marquées dans le modèle), qui peuvent avoir différents angles d'ouverture, bloquant ou permettant le passage du robot. Comme le montre la Figure 7, le robot peut aussi détecter des personnes et les placer dans le modèle de l'environnement. Grâce à des algorithmes de suivi de personne embarqués, il peut, en plus de tâches de type « aller à », assurer aussi le suivi des personnes détectées.

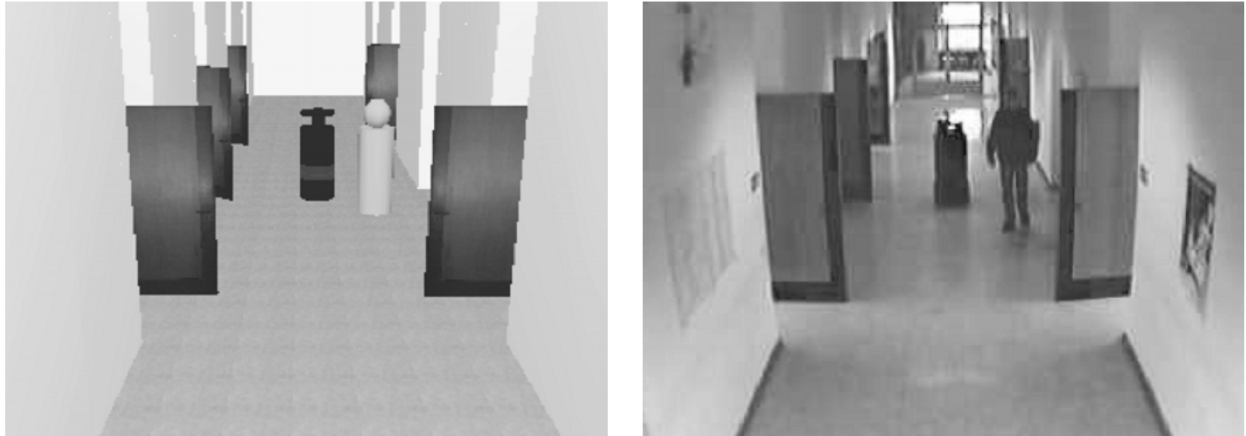


Figure 7 L'environnement et sa représentation, au même point dans le temps

[Schulz 1999] représente une continuation du travail décrit précédemment (dans [Schulz 2000]). Les délais sont maintenant pris en compte au moment de l'affichage de la position du robot. Afin de faire ceci, un schéma de simulation prédictif est utilisé. Un modèle du robot est utilisé pour déterminer sa position dans l'environnement virtuel, position qui est ensuite affichée à l'opérateur. La dernière position connue du robot (pour laquelle on a reçu des données) n'est pas affichée. Une expérience, où le délai est de six secondes, est présentée. La distance totale parcourue par le robot pendant cette expérience a été de 80,8 mètres et le temps a été de 320 s. Les performances sont acceptables, l'erreur moyenne entre la position réelle du robot est celle prédite est de 20 cm.

La téléopération sur Internet est aussi le but de [Lee 2005]. Le robot a des petites dimensions et il se trouve sur une table. Une caméra est positionnée au dessus de la scène. Le robot a un marquage en forme de flèche sur sa partie supérieure. Le flux vidéo de la caméra est présenté à l'utilisateur, ainsi qu'une reconstruction en réalité virtuelle de l'environnement. Le délai entre les deux sites est peu important (51 ms dans un sens et 72 ms dans l'autre). Deux vitesses linéaires maximales (pour deux conditions différentes) sont admises : 60 mm/s et 120 mm/s. L'accélération du robot est considérée comme instantanée, donc il n'a que deux vitesses linéaires possibles : 0 et maximum (60/120 mm/s). On n'a pas d'informations sur sa vitesse linéaire (si elle est fixe ou pas). La position du robot est prédite, en utilisant son modèle et les informations qu'on a sur sa vitesse. Des tests de suivi de trajectoire sont effectués avec et sans la prédiction de la position du robot. Pour un délai fixe, l'erreur en position est environ dix fois moins importante quand la prédiction est utilisée. Les valeurs données par les auteurs sont les suivantes : pour une vitesse maximale de 60 mm/s, on a 22,3 mm (en simulation) et 15,7 mm, sans la prédiction de la position, et 0,3 mm (en simulation) et 2,7 mm, avec la prédiction. Pour une vitesse maximale de 120 mm/s, on a 44,3 mm (en simulation) et 38,7 mm, sans la prédiction de la position, et 5,3 mm (en simulation) et 2,7 mm, avec la prédiction. Les valeurs présentées sont des valeurs moyennes.

L'environnement du robot utilisé dans [Thompson 2000], de type bras robotique, est représenté de façon schématique (moins détaillée) en réalité virtuelle. La tâche à accomplir consiste dans l'introduction d'une cheville dans un trou. Il est possible d'avoir un retour d'effort. Les auteurs constatent, pour un bruit des données faible, une approche en deux

temps : d'abord le bras robotique est rapproché de la cible (le trou), ensuite la cheville est introduite. La première étape est rapide et manque de précision, tandis que la deuxième se déroule plus lentement, et est plus précise. Si le retour d'effort ne présente pas (ou très peu) de biais, il est envisageable de l'utiliser même pour des tâches de routine.

Dans [Yoon 2001] on doit effectuer la même tâche que dans le papier précédent, cette fois le délai étant plus important (six secondes). L'environnement de travail, ainsi que la prédiction du bras robotique sont modélisés et représentés en réalité virtuelle. La Figure 8 montre une situation typique.

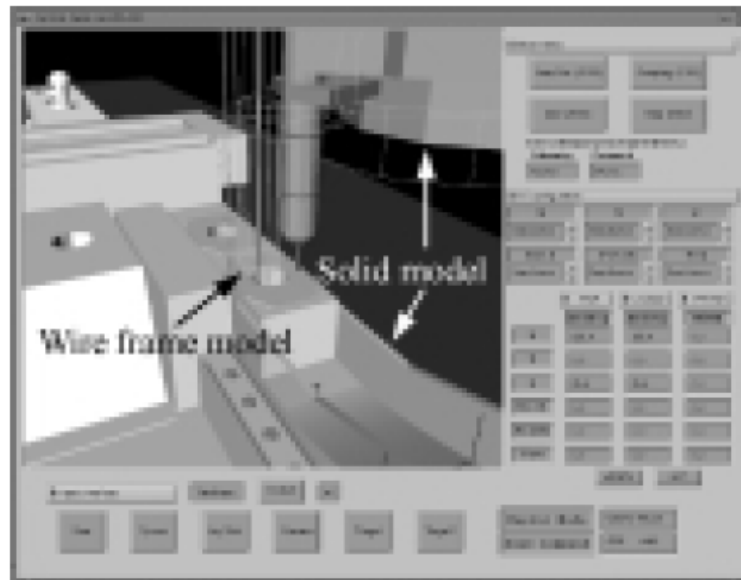


Figure 8 Modèle de l'environnement (de [Yoon 2001])

Il peut y avoir plusieurs opérateurs qui partagent le même environnement virtuel. [Chong 2003] est un tel exemple. Trois sites sont reliés entre eux : deux sites maîtres et un site esclave. La tâche nécessite la coopération des opérateurs. Les robots utilisés sont aussi de type bras.

L'idée de reconstruction décrite dans [Cobzas 2005a] n'a pas besoin d'une connaissance a priori de l'environnement. L'image acquise par la caméra est considérée comme une texture et déformée en fonction de l'ordre envoyé par l'opérateur au robot. Seule l'image initiale (panoramique) est utilisée pour extraire le modèle de l'environnement, d'où une certaine faiblesse du système par rapport aux changements inattendus.

2.4.2.4. Utilisation de la réalité augmentée pour la téléopération

L'intérêt d'utiliser une représentation mixte du site esclave est donné par le fait qu'il est possible d'avoir des informations facilement interprétables par l'opérateur, ce qui reproduit la scène (l'environnement du robot) avec une grande fidélité et où il est possible de rajouter ou modifier des informations pour aider l'opérateur dans sa tâche.

Dans [Krotkov 1996] deux idées sont avancées. La première concerne un mode de commande supplémentaire : donner de points de passage au robot. La deuxième concerne une amélioration de l'interface de contrôle : elle consiste à rajouter une prédiction de la trajectoire

du robot par intermédiaire de deux courbes, qui décrivent les côtés du robot. Celui-ci peut être autonome ou téléopéré, ajuster les ordres reçus ou s'opposer à certaines commandes considérées comme dangereuse. Le robot est supposé se trouver sur la Lune et l'opérateur sur la Terre. Si les deux premières modes ont un fonctionnement clair, les deux modes suivants demandent d'avantage d'explications. Le mode dans lequel le robot peut ajuster (modifier) les commandes reçues est un mode pendant lequel le robot est capable d'utiliser ses informations pour éviter les obstacles ou choisir une trajectoire plus sûre pour lui (une qui assurerait une meilleure traction des roues, par exemple). Le mode suivant, où le robot peut ignorer certaines commandes, est une extension du mode précédemment décrit. Le robot est capable d'évaluer son état futur et d'ignorer des commandes s'il estime que, en les exécutant, son intégrité structurelle serait mise en danger. Une évaluation de ce dernier mode de commande, sur une distance de 10 km, est décrite, avec des résultats satisfaisants. Le délai total était de cinq secondes.

L'interface proposée dans [Nielsen 2004] est très symbolique. A part le retour vidéo, seuls les obstacles détectés par le robot sont représentés dans l'environnement, sous la forme de bâtonnets. L'opérateur doit effectuer des tâches de navigation. Des tests ont été effectués (avec 56 sujets) pour déterminer si ce style d'interface est supérieur à une interface « classique » (où on a le retour vidéo affiché d'un côté de l'interface et un plan 2D de l'environnement affiché dans une autre partie de l'interface). Les indicateurs utilisés pour l'évaluation des deux interfaces ont été de deux types : objectifs et subjectifs. Les indicateurs objectifs sont le temps total et le nombre de collisions avec l'environnement. L'indicateur subjectif est l'impression d'avoir un bon contrôle sur le robot. Les résultats obtenus montrent que, même en l'absence d'un retour vidéo (seule la représentation de la reconstruction 3D de l'environnement était affichée), l'interface en réalité augmentée est supérieure à une interface où les différents éléments sont affichés séparément et où le plan est de type 2D, le robot étant tout le temps dans le centre de cette représentation. Pour l'interface classique, le temps moyen a été de 584 s, avec un écart type de 233 s, 24 collisions en moyenne (avec un écart type de 15,65) et 65% des opérateurs affirmant qu'ils ne se sentent pas du tout en contrôle du robot (0 sur une échelle jusqu'à 10). Pour l'interface mixte, le temps moyen a été de 191 s, avec un écart type de 11 s, 24 collisions en moyenne (écart type de 6,56) et la majorité affirmant qu'ils se sentent en contrôle du robot (7 sur une échelle jusqu'à 10).

Ces travaux sont continués par [Ricks 2004]. La télécommande se fait avec délai. En plus des mesures laser, on a aussi les mesures des sonars qui sont affichées, avec une couleur différente des mesures laser (des gros cylindres bleus pour les mesures de sonars et des bâtonnets verts très fins pour les mesures du laser). La position du robot dans l'environnement est actualisée avec la prise en compte du délai. Un modèle du robot est utilisé. Comme la vidéo reçue du robot n'est pas à jour par rapport à l'interface, le déplacement de celui-ci dans le monde virtuel (reconstruction de l'environnement du robot) est retardé par rapport au déplacement de l'avatar du robot. Dans la Figure 9, à gauche, le robot a tourné de 90° et puis s'arrête. La position de l'image vidéo reste en retard par rapport à l'orientation du robot, mais revient à sa place normale peu de temps après, parce que le robot ne se déplace plus.

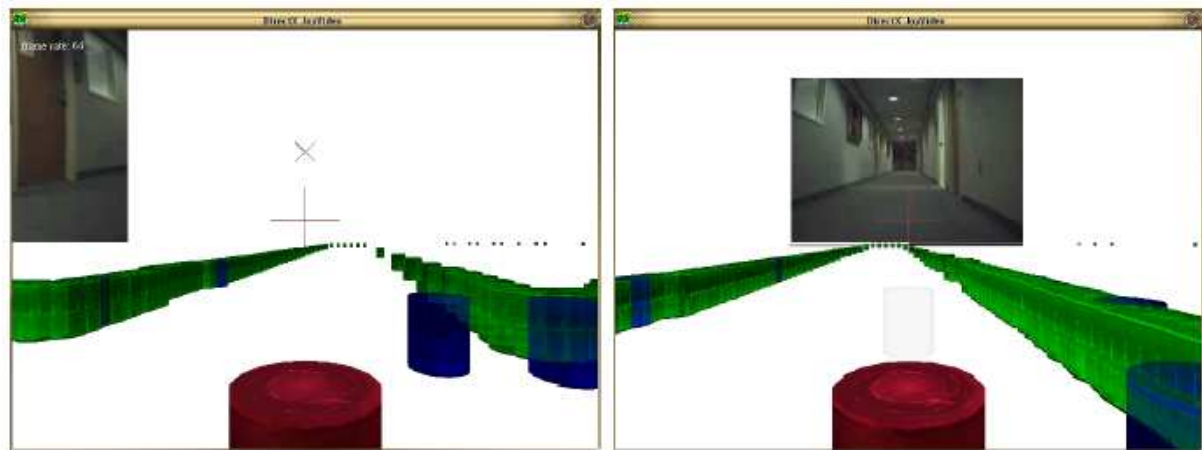


Figure 9 Position de l'image pendant le virage (gauche) et après (droite)

Trois modes d'interaction avec l'interface sont comparés dans [Nielsen 2007a]. 153 sujets ont participé à ces évaluations, 51 pour chaque mode. Les trois modes sont les suivants : joystick (il est utilisé pour contrôler le mouvement du robot et aussi pour déplacer sa caméra) ; souris (on spécifie au robot des points à atteindre en cliquant sur l'interface et on contrôle l'orientation de la caméra directement sur l'image retournée en laissant le bouton de la souris appuyé); pour le dernier mode, la souris est utilisée pour spécifier des points à atteindre et l'orientation de la caméra est contrôlée par l'intermédiaire d'icônes. Le robot n'est pas muni d'un évitement d'obstacles pendant ces essais. L'environnement est reconstruit, similairement à la Figure 9 à partir des données laser et ultrasoniques, pour aider l'opérateur dans sa tâche.

Le but des expériences était de trouver des peluches dans un labyrinthe. Le dernier mode de contrôle a été le meilleur pour cette tâche (les opérateurs ont trouvé le plus grand nombre de peluches en l'utilisant). La caméra, pour ce mode, a enregistré le plus de mouvements. Les distances parcourues et les vitesses moyennes, pour les trois modes, n'ont pas été statistiquement différentes ($p > 0,05$).

Le bras robotique utilisé par [Friz 1998] a une base fixe. L'opérateur doit manipuler différents objets en bois situés sur une table, laquelle se trouve en dessous du bras robotique. La réalité augmentée est utilisée pour surmonter les problèmes posés par une représentation sur un écran 2D de l'environnement du site esclave. Différents outils sont proposés à l'opérateur : mesurer les distances et les angles, déterminer la projection du bout de la pince sur le plan de travail, afficher la position de la pince, etc. Le système a été conçu pour fonctionner avec un délai maximal de 15 s. La Figure 10 montre l'interface proposée à l'opérateur.

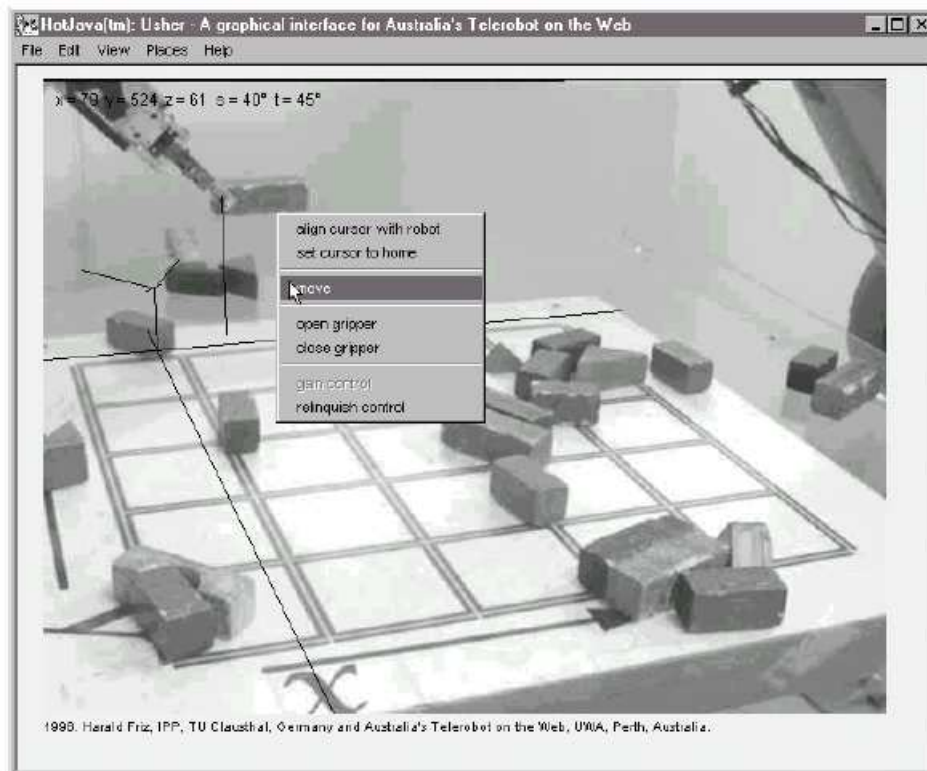


Figure 10 Interface de contrôle proposée par [Friz 1998]

Un affichage prédictif de la scène est utilisé dans [Yerex 2003]. L'image est modifiée afin qu'elle corresponde à la perspective qui serait affichée s'il n'y avait pas de délai. Un étiquetage à la main des points qui vont être suivis dans l'image est néanmoins nécessaire. Ceci réduit le cadre d'utilisation à des applications avec une caméra fixe, où l'environnement n'est pas modifié entre deux connexions.

La même idée d'adaptation de l'image est reprise dans [Cobzas 2005b]. Des propositions d'extraction d'un modèle géométrique de la scène à partir d'une caméra non calibrée sont faites, mais pas testées.

Le cas de la téléopération d'un robot de type rover sur la Lune est analysé en [Miller 2005]. Le délai utilisé dans les expériences menées est de 5,3 s (4 s pour une communication aller-retour avec la Lune et 1,3 s de latence pour le traitement de l'information dans le système de communication). Le contrôle du robot s'effectue en vitesse, par l'intermédiaire d'un joystick. L'aide consiste dans la conservation d'une trace des commandes (en vitesse) envoyées au robot et l'affichage de cet historique sur l'image reçue du robot. On peut observer ceci sur la Figure 11, en bas à gauche. Dans des conditions idéales (sans délai), ce graphique serait un carré, avec l'indicateur situé sur la position du joystick. Dans la figure présentée, le délai est fixé à quatre secondes. L'abscisse montre les commandes envoyées en X, (le robot doit avancer) et l'ordonnée celles envoyées en Y (le robot doit d'abord tourner à gauche, et ensuite revenir en ligne droite). Ainsi, l'anticipation de l'évolution de la trajectoire du robot sera plus facile pour l'opérateur (surtout après une période d'accommodation avec le robot).



Figure 11 Historique des commandes

2.4.2.5. Fenêtres prédictives

Il est envisageable d'essayer de prédire la scène courante (l'image que l'opérateur serait censée voir s'il n'y avait pas de perturbations dans sa liaison avec le robot). Pour y parvenir, une partie de la dernière image reçue par l'opérateur est manipulée afin d'être transformée dans l'image qu'il verrait si les contraintes (délai, bande passante, etc.) n'existaient pas. On appelle la partie de l'image qui sera ensuite transformée une fenêtre.

Dans [Cobzas 2005a], une image panoramique (à 180°) est utilisée pour extraire un modèle de la scène et faire de la prédiction. L'acquisition de la scène s'effectue à l'aide d'un laser et d'une caméra, les deux installés sur un support capable de rotations dans les plans vertical et horizontal (pan et tilt). Des lignes verticales (en 3D) de ce modèle sont comparées à des lignes verticales dans l'image courante, afin de déterminer la position du robot dans l'environnement. Ces lignes peuvent provenir de trois sources : discontinuités dans l'intensité de la couleur, discontinuités de surface et occultation de contours sur une surface lisse. Les auteurs se sont servis uniquement de deux premiers cas. Pour afficher la scène courante, trois éléments sont nécessaires au système : un modèle de l'environnement (acquis à l'aide de l'ensemble rotatif caméra-laser), une position de la caméra du robot (obtenu en identifiant les lignes identiques dans le modèle et l'image du robot) et une texture à utiliser sur le modèle de l'environnement. Cette texture peut être acquise à partir de deux sources : l'image précédente (en provenance de la caméra du robot) ou le panorama de la scène (également utilisé pour extraire le modèle de l'environnement). Les résultats obtenus en utilisant l'image précédente comme source pour la texture sont meilleurs (par rapport aux images réelles) que ceux qui utilisent le panorama. Néanmoins, l'image présentée en utilisant l'image précédente n'est pas complète, à cause d'un manque d'information (si le robot a bougé, bien entendu) sur la scène courante (ces parties seront noires).

[Sun 2001] examine l'utilisation d'une partie de l'image pour la transmission ou l'affichage. L'exemple donné est celui d'un orateur et d'une caméra fixe, à grand angle. La prédiction de la position de l'orateur dans l'image est faite avec l'aide d'un filtre de Kalman. L'idée de cette démarche est d'isoler une portion de l'image, afin de baisser les contraintes de bande passante et d'assurer un zoom intelligent sur l'image. Le déplacement de la fenêtre virtuelle dans l'image complète est conditionné, afin d'éviter que l'image bouge tout le temps, par le déplacement de l'orateur. Si ce déplacement est petit, alors la fenêtre virtuelle ne sera pas déplacée. S'il continue à se déplacer (pour sortir de la fenêtre), alors elle sera recentrée sur l'orateur. Si celui-ci continue de bouger, alors la fenêtre se déplace avec lui, afin qu'il reste dans son centre. La Figure 12 illustre cette idée.

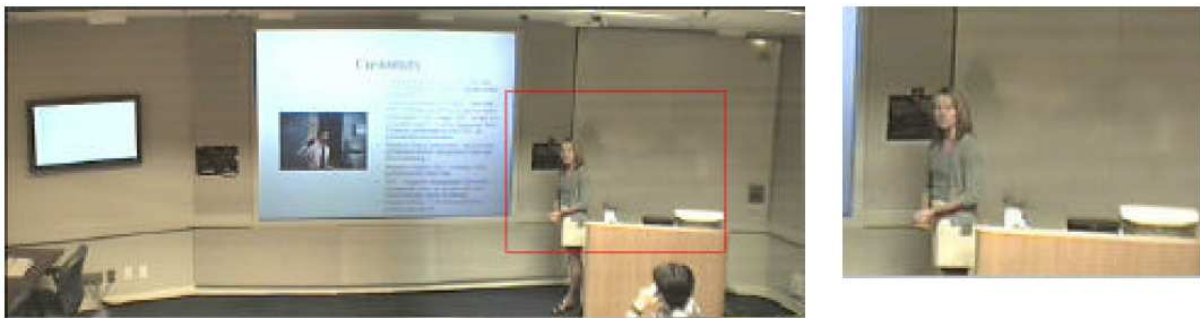


Figure 12 Image panoramique et fenêtre virtuelle (avec agrandissement de taille)

La caméra utilisée dans [Baldwin 1999] est orientée vers un miroir conique. La caméra est orientée verticalement (elle pointe vers le haut). L'image ainsi obtenue couvre, de façon effective, un champ de 360° . La caméra est fixe (stationnaire sur le robot) et a une résolution de 640×480 pixels. Dans cette image panoramique, une seule fenêtre est sélectionnée et affichée. Ce système est utilisé en remplacement pour un système pan-tilt classique. Il offre la possibilité d'avoir une image à 360° sans déplacer la caméra. En contrepartie, sa résolution est réduite. Ce système est installé sur un robot mobile. La Figure 13 montre l'ensemble robot-caméra, la caméra se trouvant normalement montée sur le robot.

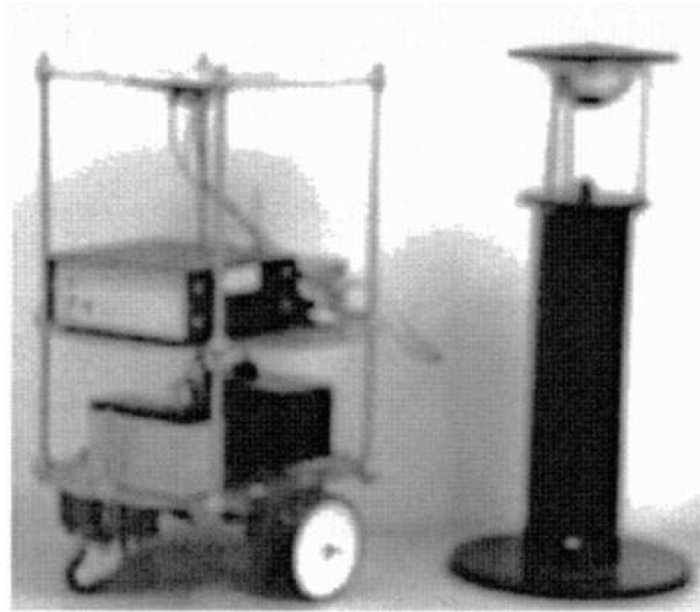


Figure 13 Robot mobile et caméra omnidirectionnelle

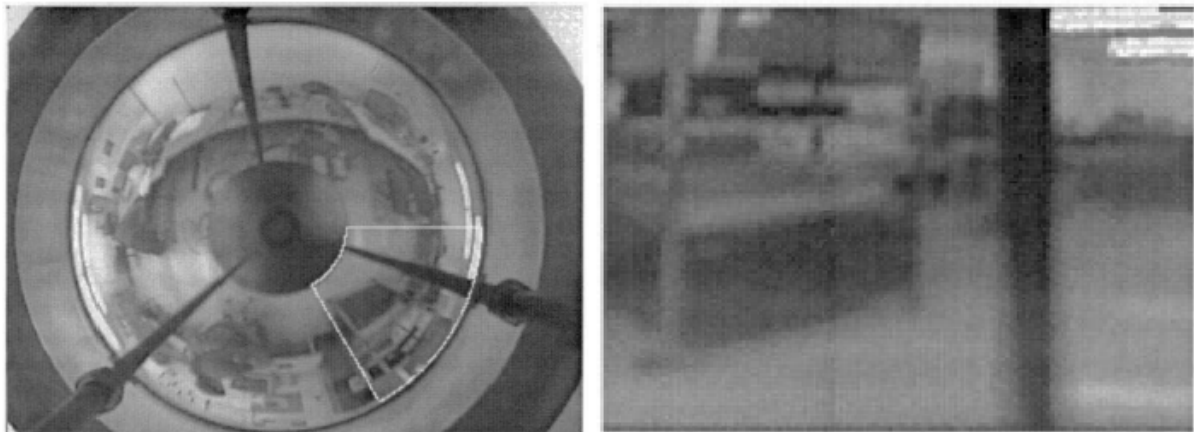


Figure 14 Image panoramique (gauche) et fenêtre corrigée (droite)

La Figure 14 illustre une situation typique. A gauche on a l'image panoramique. A droite on a l'image obtenue par extraction et déformation d'un secteur d'une couronne. L'image panoramique a une résolution de 640x480 pixels. Elle correspond à un champ de vue horizontal de 360°. Le secteur de couronne a une résolution de 240x180 pixels (environ), qui sont projetés sur une image de 320x240 pixels. Cette image correspond à un champ de vue horizontal de 60°.

Les auteurs rappellent les facteurs qui interviennent dans le délai entre le robot et le poste de l'opérateur. Le premier facteur est lié à la capture de l'image suivante par le système d'acquisition vidéo. Il désigne le temps nécessaire pour la capture d'une scène par la caméra et sa transmission. Les mesures effectuées par les auteurs placent ce temps entre 33 et 143 ms. Le facteur suivant est le temps de transmission entre les deux sites (maître et esclave). Le réseau et la taille de l'image à transmettre intervient ici. Comme l'image qui sera transmise est un fragment de l'image acquise, sa taille est moins importante. Le troisième facteur est lié

au traitement qui a lieu du côté du client. Il s'agit du processus de déformation de l'image et le délai de l'affichage de cette image à l'écran.

La prédiction de la fenêtre dans l'image panoramique est réalisée par l'intermédiaire d'un filtre de Kalman. Le contrôle de la fenêtre vidéo dans l'image panoramique est réalisé par l'opérateur humain par l'intermédiaire de la souris. La trajectoire de la souris, prédite par un filtre de Kalman, est utilisée pour fournir une nouvelle image extraite à partir de la fenêtre à l'opérateur. La souris est modélisée comme un point, avec une certaine masse, lequel se trouve sous l'influence d'une force externe, constante. La force de frottement est prise en compte (considérée comme étant non-négligeable). Dans le modèle de l'environnement, on a du bruit de processus et de bruit de mesures. Chaque type de bruit est supposé être un processus aléatoire, avec une moyenne égale à zéro. La matrice de covariance des erreurs dues au bruit du processus, Q_k , et la matrice de covariance des erreurs dues aux bruits de mesure, R_k , sont des matrices diagonales et ont la forme suivante :

$$Q_k = \text{diag}(q_k^2, \quad q_k^2, \quad q_k^2)$$

$$R_k = \text{diag}(r_k^2, \quad r_k^2)$$

En utilisant des profils de la souris définis par les opérateurs, les auteurs ont pu déterminer les valeurs de q_k et r_k qui minimisent l'erreur carrée moyenne pour tous les profils.

Une étude empirique a été menée pour vérifier l'efficacité du prédicteur. Quand la souris n'enregistre pas des changements subits de direction, il arrive à bien prédire sa position. Dans le cas de changement rapides, le prédicteur a besoin de quelques instants (pour récupérer des positions plus récentes de la souris) pour se stabiliser et de revenir à une bonne approximation de la trajectoire. La Figure 15 montre les résultats recueillis pendant une telle expérience. Sur l'abscisse on a le temps, en secondes, et sur l'ordonnée on a la distance, en pixels, de l'axe x de la souris sur l'écran.

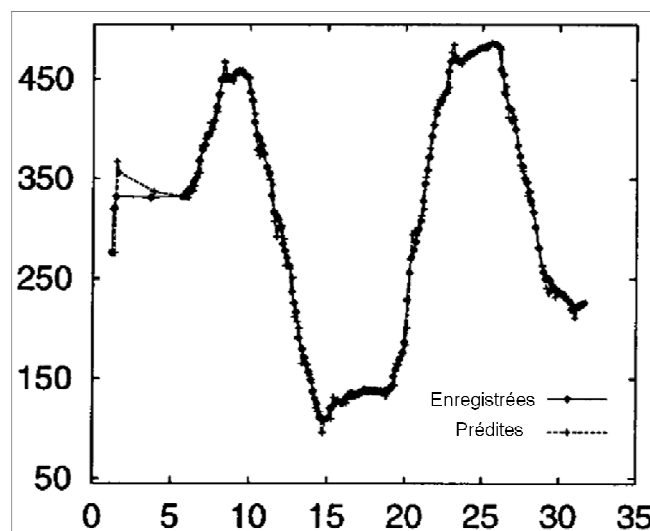


Figure 15 Différences entre les positions enregistrées et prédites [Baldwin 1999]

Une deuxième étude s'est penchée sur les effets de cette aide sur les opérateurs. Son but était de sélectionner et d'identifier une série de formes, situées autour de la caméra. L'utilisation de la prédiction confère une qualité supérieure de l'image. Mais, quand les sujets doivent réaliser

le meilleur temps possible pour cette tâche, ils semblent être moins concernés par la qualité des images reçues. Pour pouvoir tirer ces conclusions, les auteurs ont enregistré les temps de réalisation de la tâche, ainsi que les impressions subjectives des opérateurs.

L'utilisation de la prédiction et d'une caméra panoramique sont une bonne alternative à une caméra qui a un champ visuel habituel et une fréquence de rafraîchissement réduite. La prédiction est susceptible d'améliorations si la modélisation des erreurs et l'identification des paramètres peuvent être raffinées.

3. Aide proposée

Ce paragraphe contient, au début, une description de l'aide proposée. Deux campagnes d'évaluations sont ensuite présentées. La première a été réalisée dans le cadre du laboratoire IBISC. La deuxième a été réalisée au sein d'une institution spécialisée dans le suivi des personnes âgées, en partenariat avec l'université de la Corogne. Pour cette campagne, les sujets ont été des professionnels de la santé et de la gérontologie. Les leçons tirées de ces deux campagnes sont présentées à la fin.

3.1. Proposition

La méthode décrite dans la partie 2.4.2.5 est limitée aux caméras panoramiques. Nous souhaitons étendre cette méthode, pour pouvoir l'utiliser avec des caméras normales (avec un angle horizontal compris entre 40° et 80°). Le robot sur lequel la caméra est embarquée est un robot mobile, non holonome (deux roues motrices indépendantes). Les mouvements du robot s'effectueront sur un plan (pas de montées/descentes) et le glissement des roues sur le sol sera minimal. Une autre supposition est que la caméra est orientée parallèlement au sol. On a également une idée sur l'environnement où le robot évoluera. Il s'agit de l'intérieur de bâtiments, plus exactement dans des résidences, maisons, appartement ou bureaux. Dans ces conditions, on s'attend à ce que les dimensions des pièces soient comprises entre $2m \times 2m$ et $10m \times 10m$. On part de la supposition qu'on n'a aucune connaissance sur le contenu des pièces (objets ou personnes s'y trouvant), leur composition ou leur emplacement.

Le schéma présenté dans la Figure 16 montre le cas de la téléopération avec délais, sans aide. L'opérateur humain (H sur le schéma) envoie une commande au robot réel (RR sur le schéma). Cette commande est retardée avec le délai d_1 . Après l'exécution de la commande, le retour arrive à l'opérateur avec le délai d_2 . Le délai total (perçu par l'opérateur) est composé de $d_1 + d_2$. Si cette somme dépasse un certain seuil, l'opérateur n'aura plus un comportement de télécommande continu et il adoptera une stratégie de type « move-and-wait ». Ce seuil dépend du contexte (la vitesse maximale du robot - plus le robot peut être déplacé vite, plus la téléopération est sensible aux délais) et de l'opérateur (plus il est expérimenté, plus ça lui sera facile de prédire la trajectoire du robot, et également de ses capacités de concentration). En adoptant cette stratégie, l'opérateur suit, petit à petit, la trajectoire qu'il désire, en évitant les collisions avec l'environnement du robot. Les désavantages de cette stratégie sont le fait que la vitesse moyenne du robot n'est pas élevée et qu'on court le risque que l'opérateur se fatigue de façon prématurée ou qu'il s'ennuie et devient frustré avec cette façon de pilotage du robot.

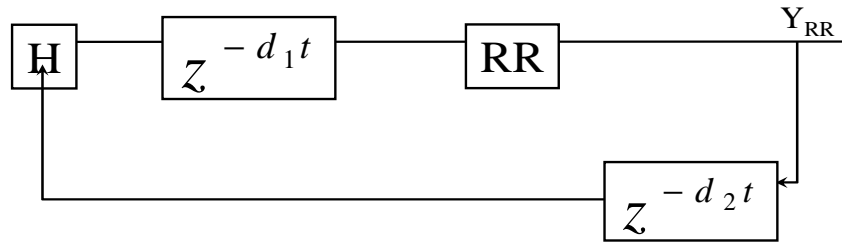


Figure 16 Téléopération avec délais

Si on connaît le type du robot utilisé par le téléopérateur, il est possible de déterminer une modélisation de sa réponse aux commandes de l'opérateur. Ainsi, il est possible de déterminer quel sera le déplacement robot, comme conséquence de l'exécution des ordres qui lui ont été envoyés par le poste de l'opérateur. Si on connaît le déplacement du robot, alors on connaît aussi celui de la caméra. Ce modèle du robot est utilisé sur le poste de l'opérateur pour anticiper la réponse du robot aux ordres (Figure 17). Cette anticipation est utilisée pour modifier l'image affichée sur l'écran de l'opérateur. En remplaçant l'image courante (la dernière image envoyée par le robot) par cette même image, modifiée en prenant en compte la position où le robot devrait se trouver (donnée par VR) s'il n'y avait pas de retard, on souhaite que l'opérateur ignore le fait qu'il y a un retard. Si la modification crée une image proche des attentes de l'opérateur, on suppose que celui-ci utilisera ce retour modifié d'une façon normale (sans arrêter le robot pour attendre des rafraichissements de l'interface de pilotage). Comme présenté dans la Figure 17, la modification de l'image dépend des commandes envoyées par l'opérateur et du délai global. Plus ce délai est important, plus les commandes envoyées (à une fréquence fixe) seront cumulées en vue de déterminer les modifications à faire. Le modèle du robot, le robot virtuel, est indiqué comme VR sur le schéma. Les mêmes commandes que celles pour le robot réel lui sont envoyées. Sa sortie (Y_{VR}) est utilisée pour la modification du retour, en provenance du robot réel, qui est affiché sur l'interface de contrôle de l'opérateur. A chaque instant, la différence de position entre VR et RR est calculée, et la vidéo est modifiée en fonction de cette différence. Cette utilisation d'un robot virtuel à partir duquel on modifie les informations reçues du robot réel a été décrite dans [Nadrag 2010].

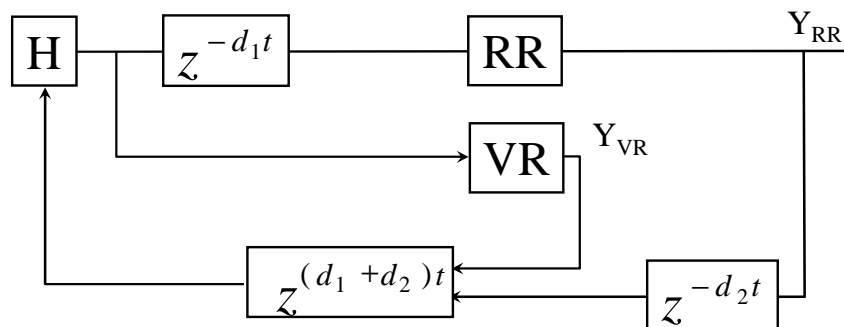


Figure 17 Téléopération avec correction des délais

Pour obtenir des modifications vraisemblables, un travail de calibration est réalisé hors ligne. On part d'une mire située à une distance connue devant le robot. Comme on a une idée sur la taille des pièces, il est possible de partir d'une valeur initiale approximative (un tiers de la

taille de la pièce, par exemple), et de la raffiner en fonction du contexte d'utilisation. Pour notre cas, le robot ne sera pas proche des obstacles, en fonctionnement normal, grâce à l'évitement d'obstacles qu'il utilise. Une calibration est ensuite faite, de façon que la position de la mire dans l'image modifiée soit la même que sa position dans une image obtenue à la suite du déplacement du robot. Ceci sera fait pour des mouvements de rotation et de translation. Ainsi, les objets situés à cette distance par rapport au robot seront bien représentés dans l'image modifiée. Un désavantage est que, plus un objet est loin de la distance utilisée pour la calibration, pire sa vraisemblance avec son correspondant dans l'image modifiée sera.

La dernière image reçue par le poste de l'opérateur sera modifiée en fonction de la réponse du « robot virtuel » (le modèle du robot qui se trouve sur le poste de l'opérateur). Si, à la suite des commandes, le robot virtuel détermine que le robot réel avance, alors l'image affichée à l'opérateur sera agrandie (on effectuera un zoom sur elle). Ceci est une approche globale, sur le délai total ($d_1 + d_2$). Si l'image obtenue à la suite de ces modifications dépasse l'espace qui lui est attribuée sur l'interface homme-robot, ses bords seront coupés, pour qu'elle y rentre. Inversement, si la commande envoyée au robot le fait reculer, la taille de l'image affichée sera diminuée. Ceci laissera un espace vide entre les bords de l'image et son espace sur l'interface de commande. Si on a décidé, par le biais du robot virtuel (le modèle du robot qui se trouve sur le poste de l'opérateur) que le robot tourne à gauche, alors l'image affichée à l'opérateur est déplacée à droite (par un mouvement de translation). Inversement, si le robot tourne à droite, alors l'image affichée à l'opérateur est déplacée à gauche. Ces quatre cas pourront être combinés, bien sûr, si le résultat des commandes envoyées (mais pas encore exécutées par le robot) le demande (par exemple, si on avance le robot et on le fait tourner vers la droite, alors l'image sera agrandie et déplacée vers la gauche). La Figure 18 montre le résultat (prédit) pour une commande dans laquelle le robot avance en tournant à gauche. Etant donné qu'on ne dispose pas d'informations sur ce qui se trouve à gauche de la dernière image reçue, une bande noire y sera affichée.



Figure 18 Image prédite (gauche) et dernière image reçue

Comme on souhaitait observer le comportement des opérateurs, on n'a pas utilisé l'évitement d'obstacles pendant la première campagne d'évaluations de cette aide. Les résultats, détaillés le chapitre suivant, ont montré son utilité, mais ont aussi rappelé l'importance de l'utilisation de l'évitement d'obstacles. Le schéma de contrôle final retenu est présenté dans la Figure 19. L'évitement d'obstacles est embarqué sur le robot. Des expériences, analysées dans le chapitre suivant, ont montrée que l'aide proposée est toujours utile, même si les commandes fournies par l'évitement d'obstacles ne sont pas incluses dans la modification du retour vidéo.

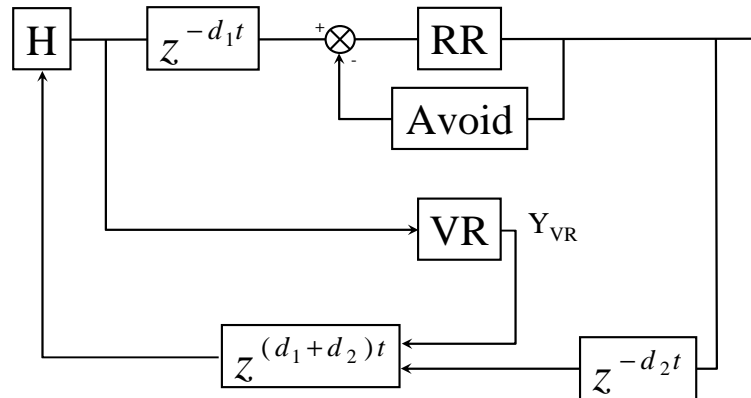


Figure 19 Téléopération avec délais et évitement d'obstacles

3.2. Évaluations

En utilisant l'aide décrite dans le chapitre précédent, deux séries d'évaluations ont été menées. La première série d'évaluations s'est déroulée au sein du laboratoire IBISC. Cette première série s'est concentrée sur la perception des délais par les opérateurs et avait comme objectif de vérifier que l'aide proposée masquait bien les délais. La deuxième série d'évaluations a eu lieu à l'Université de la Corogne (UDC), en Espagne. L'objectif a été d'observer si des opérateurs de terrain (des professionnels du maintien à domicile) réussiraient à piloter le robot en présence de délais de transmission avec la commande complète (Figure 19).

3.2.3. Évaluations à IBISC

Dans cette première série d'évaluations, le délai a été choisi à une seconde (500 ms dans chaque direction). Le choix de ce délai a été fait en accord avec les travaux de [Garcia 2003], qui trouve que le délai, sur des distances importantes (intercontinentales), a une composante continue d'environ 400 ms et, de temps en temps, des sauts importants. La valeur que nous avons choisie (une seconde) est nettement supérieure à 400 ms (délai total - aller-retour), ce qui signifie que, si l'aide proposée est efficace, elle pourrait être utilisée dans des cas réels.

3.2.3.1. Description du protocole expérimental

Le robot utilisé, qui est montré dans la Figure 20, est un robot non holonome. Il a une base circulaire, avec un diamètre de 55 cm. Douze capteurs ultrasons sont disposés autour de la base, sur un cercle, de façon équidistante. Un capteur de distance laser est installé à l'avant du robot. Ce capteur laser a une ouverture de 210° , pour une distance maximale de détection de 5m. La caméra utilisée pendant cette série d'expériences a une ouverture horizontale de 70° . Elle peut être déplacée sur la verticale (modifier son inclinaison – tilt) et sur l'horizontale (mouvement de rotation panoramique – pan), grâce à une tourelle installée sur le robot. La masse du robot est de 70 kg. Le robot est muni d'une connexion Wi-Fi, pour communiquer avec l'extérieur.



Figure 20 Le robot Lina

La Figure 21 montre l'interface de contrôle utilisée. L'interaction entre l'opérateur et le robot s'effectue par l'intermédiaire de la souris. L'opérateur l'utilise pour déplacer la flèche blanche, à l'intérieur de l'ellipse.

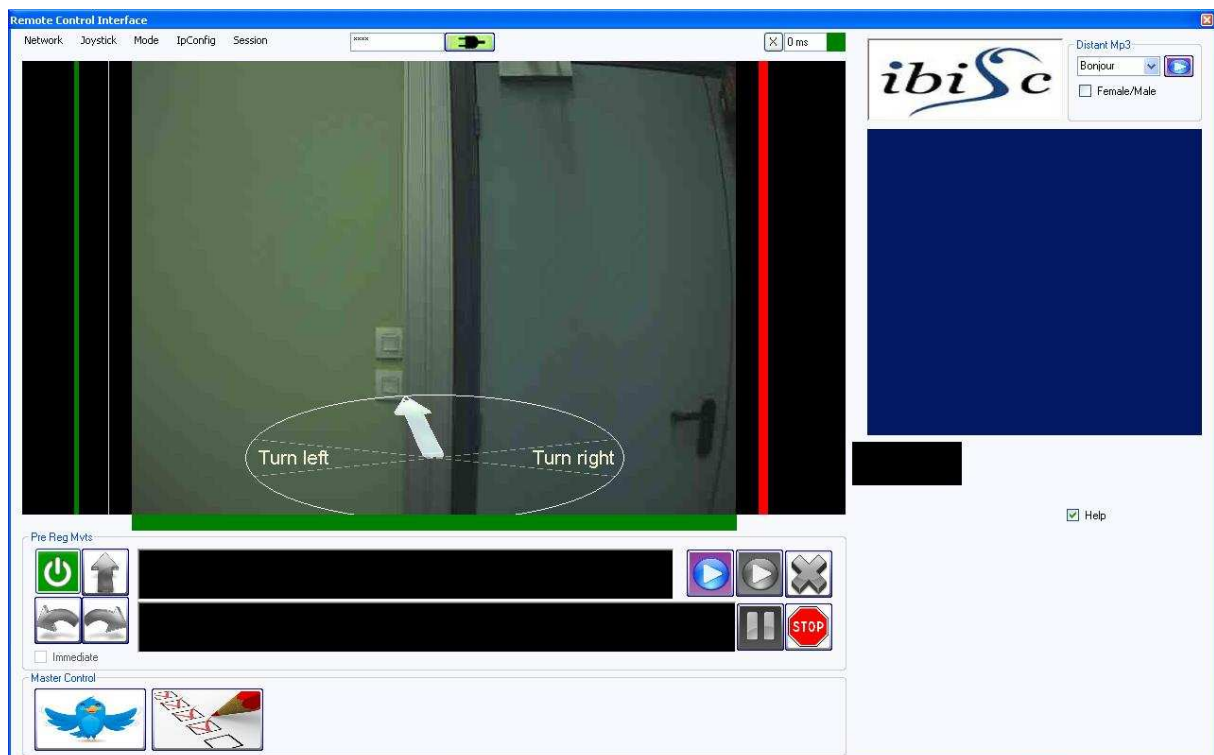


Figure 21 L'interface de contrôle utilisée pendant les expériences

La longueur de la flèche donne une indication sur la vitesse du robot. Sa base se trouve au centre de l'ellipse qui contrôle la vitesse du robot. Sa valeur maximale est délimitée par le contour de l'ellipse (la flèche ne peut pas la dépasser).

On a constaté qu'il était difficile de faire tourner le robot sur place. Pour faciliter cette tâche, deux zones ont été définies sur l'ellipse de contrôle des vitesses. Ces deux zones se trouvent à gauche et à droite du centre de l'ellipse, sous la forme de deux secteurs (« Turn left » et « Turn right » sur la Figure 21). Si la flèche de contrôle des vitesses se trouve dans une de ces régions, alors la vitesse linéaire qui sera transmise au robot sera nulle. Seule la vitesse angulaire sera envoyée au robot.

Les informations concernant la proximité des obstacles sont présentées sous une forme synthétique. Les mesures brutes (obtenues directement des capteurs de distance) sont agrégées pour décrire de façon symbolique la position des obstacles dans trois directions. Ces trois directions sont l'avant du robot (la barre horizontale qui se trouve sous l'image vidéo) et les côtés (gauche et droite) du robot.

Pour la distance séparant le robot des obstacles se trouvant devant lui, la barre varie en dimension (longueur), en réduisant ses parties gauche et droite simultanément, sa couleur variant (vert – distance suffisante ; rouge – attention, obstacle très proche du robot).

Concernant les côtés, la distance entre les barres latérales et le retour vidéo constitue un premier indice. Plus la barre est proche de l'image, plus le robot est près des obstacles. Son épaisseur peut aussi varier. Si les obstacles sont éloignés ou inexistant, la barre est fine. Plus les obstacles sont proches, plus elle grossit. Comme pour la distance frontale, la couleur varie également, entre vert (obstacle éloigné) et rouge (obstacle proche).

Cette première série d'évaluations s'est déroulée dans une situation écologique (des vraies pièces dans le laboratoire). L'opérateur n'a pas de contact visuel avec le robot. Il se base sur le retour vidéo et sur les capteurs de distance embarqués sur le robot.

Dix-huit sujets volontaires ont participé à cette série d'expériences. Leur âge était compris entre 20 et 23 ans et ils connaissaient tous comment utiliser un ordinateur. Ils n'avaient pas de connaissances antérieures sur la téléopération des robots.

Six trajectoires ont été retenues pour cette série (Figure 22). Chaque sujet a conduit le robot sur cinq trajectoires différentes, choisies de façon aléatoire.

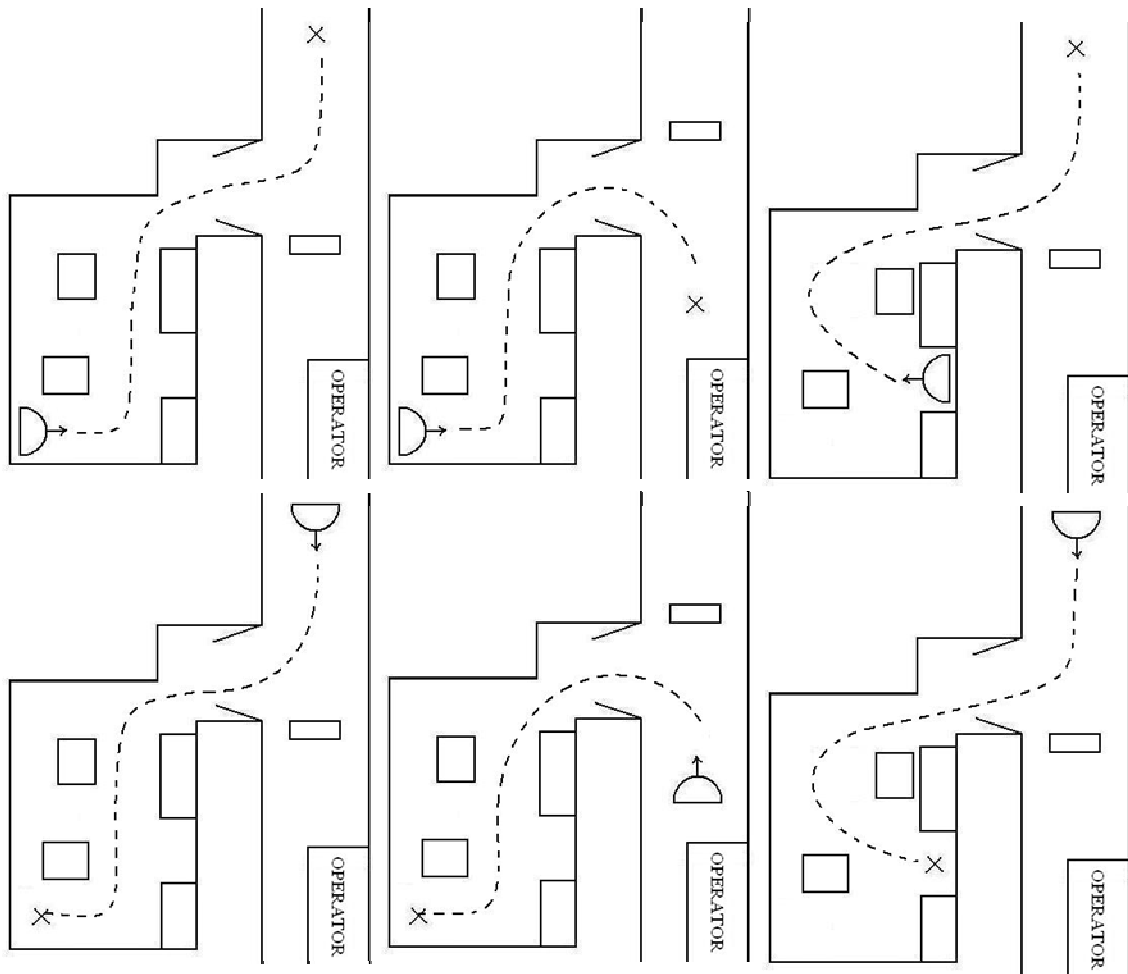


Figure 22 Trajectoire possibles

Une familiarisation avec le système a été effectuée par chaque participant, avant de commencer les expériences de téléopération.

Deux conditions ont été analysées : une condition de contrôle, sans correction des effets du délai (sans aide) et une condition avec la correction des effets du délai activée (appelée « condition zoom »).

L'hypothèse qu'on a voulu tester était de savoir si la condition zoom pouvait cacher le délai à l'opérateur. Le nombre de collisions avec l'environnement et le temps pour parcourir une trajectoire ont été enregistrés. Dans le cas où, à la suite d'une collision, le robot reste bloqué, on ne compte pas le temps de ramener le robot dans une position antérieure, d'où l'opérateur pourra contourner l'obstacle (ou, au moins, passer de côté, en le touchant, mais en ne restant pas bloqué).

3.2.3.2. Résultats obtenus

Pour la première condition (sans correction du délai), on a observé le fait que les opérateurs n'employaient pas un mode de contrôle continu. Le type de comportement qu'ils adoptaient était de faire un court déplacement et d'attendre la réception de nouvelles données, correspondant à la position (à l'arrêt) du robot (comportement de type « move-and-wait »).

Dans la deuxième condition (avec aide), ce type de comportement n'a plus été observé.

Le temps moyen pour parcourir une trajectoire a été de 85 s pour la première condition. Avec aide, il n'a été que de 65,05 s. En effectuant une analyse de la variance (ANOVA), on trouve que la différence importante (19,95 s) nous permet de rejeter hypothèse nulle ($p = 0,009$). En utilisant une stratégie de déplacement qui implique des arrêts fréquents, l'opérateur perd beaucoup de temps en attendant d'observer le résultat de ses actions.

Des résultats intéressants ont été obtenus concernant les collisions avec l'environnement. Sans aide, on a observé une moyenne de 0,33 collisions/trajectoire. Avec l'aide activée, cette moyenne est passée à 0,74 collisions/trajectoire. En effectuant une analyse de la variance (ANOVA), on trouve qu'on peut rejeter l'hypothèse nulle ($p = 0,039$). La différence moyenne est de 0,41 collisions/trajectoire.

3.2.3.3. Discussion des résultats

L'efficacité de l'aide est démontrée par le temps inférieur nécessaire à parcourir les différentes trajectoires proposées aux opérateurs par rapport à la condition avec retard mais sans aide. Non seulement le temps est inférieur, mais aussi la façon de piloter le robot est proche de celle habituelle (sans délai, sans aide), c'est à dire d'avoir une continuité dans l'envoi des commandes au robot.

On peut avancer l'hypothèse que l'aide facilite le contrôle des opérateurs, qui se comportent, en pilotant le robot, comme s'il n'y avait pas de délai. En cas de délais de transmission, ils vont plus vite avec le robot dans la condition avec aide que dans la condition sans aide, en n'ayant pas besoin de faire des pauses pour attendre d'avoir des informations récentes sur l'environnement du robot, quand celui-ci est arrêté (« move-and-wait »). Même s'ils ont besoin de changer la direction du robot d'une façon importante (90°), les opérateurs continuent à appliquer une vitesse linéaire au robot pendant le virage. On peut conclure que l'aide masque les délais de transmission à l'opérateur.

Les opérateurs ont appris comment piloter le robot, mais ils ne sont pas des experts. Le champ de vue de la caméra rend possible la situation qu'un obstacle se trouve à côté du robot, mais hors du champ de vision de la caméra. Quand l'aide n'est pas activée, les opérateurs avancent petit à petit et, quand ils tournent le robot sur place, ont tendance de survivre. Par conséquent, ils observent plus les alentours du robot (ses côtes, surtout). Ils perçoivent si l'obstacle est dépassé ou non et peuvent ainsi prendre la bonne décision pour la poursuite de la trajectoire : si l'obstacle est dépassé le robot peut déclencher son virage ; dans le cas contraire, le robot doit continuer à éviter l'obstacle avant de déclencher son virage. En pilotant de manière continue le robot, grâce à l'aide, les opérateurs survivent moins, donc ils observent moins les alentours du robot. Comme ils n'ont pas beaucoup d'expérience en matière de téléopération, ils ont du mal à apprécier si l'évitement de l'obstacle, qui est sorti du champ de vue de la caméra, est terminé ou non. Ainsi, ils tournent parfois trop tôt ou avec un rayon de courbure trop petit et le robot entre en collision avec l'obstacle. On a déjà remarqué un comportement identique sans retard : c'est une raison pour laquelle nous avons travaillé sur une interface écologique, présentée au chapitre 3. Ce problème des collisions a été rencontré parce qu'on a voulu observer l'influence de l'aide sur les opérateurs, si elle arrivait à leur cacher l'existence des délais. Pour éviter les collisions entre le robot et l'environnement, on peut activer l'évitement d'obstacles, ce qui a été réalisé dans la série d'expériences suivante.

3.2.4. Évaluations à UDC

Une deuxième série d'évaluations s'est déroulée en Espagne, à l'Université de la Corogne (UDC). Cette série d'évaluations s'est inscrite dans un cadre plus large d'évaluations, destinée à observer la facilité d'utilisation du robot par des personnes atteintes de troubles cognitifs légers, dans le cadre des exercices (ludiques) de stimulation cognitive.

3.2.4.1. Description du protocole expérimental

Par rapport à la précédente campagne d'évaluations, la principale différence a été l'utilisation de l'évitement d'obstacle réactif sur le robot. Ce type d'évitement est déclenché en fonction de la proximité du robot des obstacles (il n'y a pas de planification dedans).

Deux conditions ont été comparées, comme pour la première série d'évaluations : contrôle du robot avec évitement d'obstacles et sans aide visuelle et contrôle du robot avec évitement d'obstacles et avec aide visuelle (zoom).

Une période de familiarisation avec le contrôle du robot, sans délais, a été laissée aux opérateurs avant le début des évaluations. Quatre personnes y ont pris part, elles étaient âgées entre 22 et 33 ans. Elles connaissaient toutes comment utiliser un ordinateur et n'avaient pas de connaissances antérieures sur la téléopération des robots.

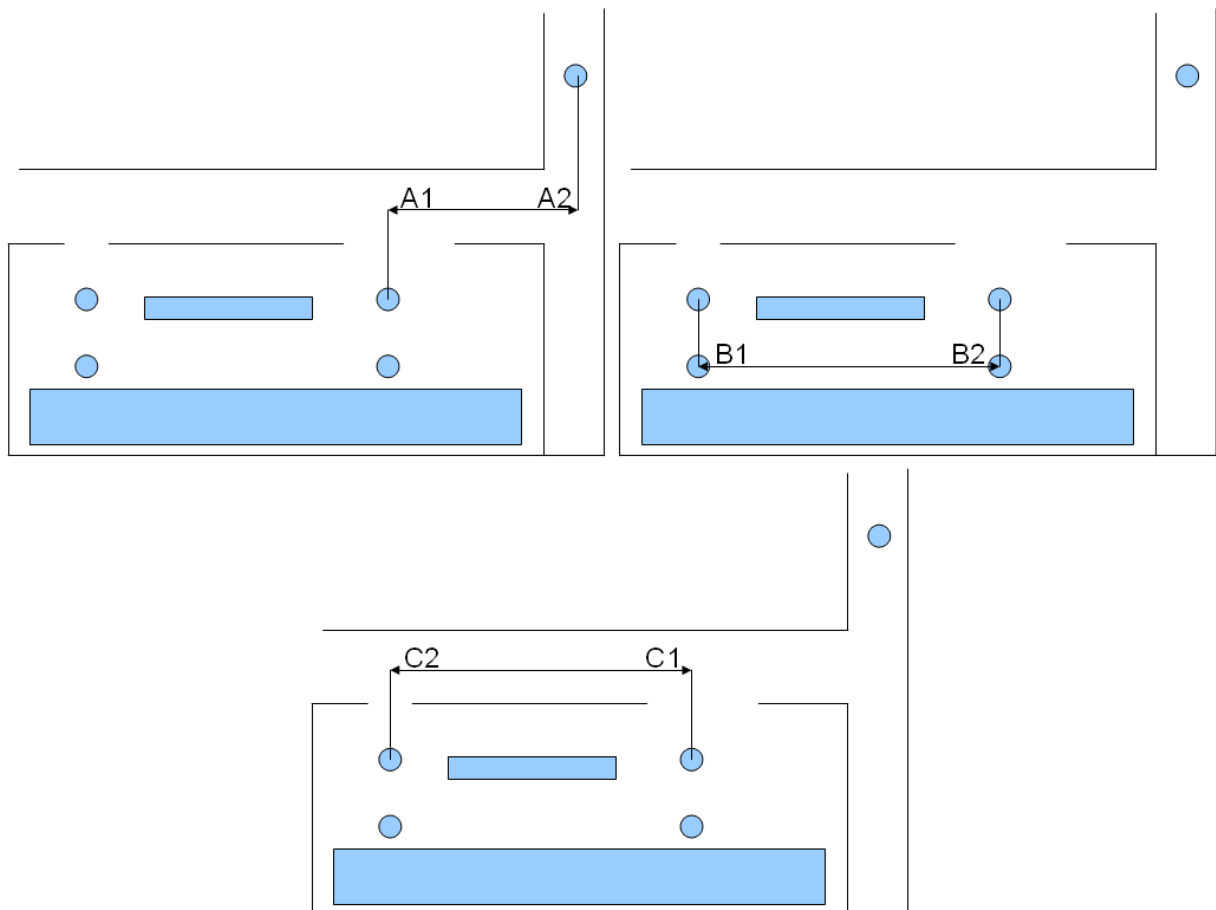


Figure 23 Trajectoires possibles à UDC

La Figure 23 montre les six trajectoires potentielles. Les cercles représentent les positions initiales et finales du robot et les carrés représentent des obstacles dans les pièces. On peut observer l'existence de deux portes (ouvertes pendant les expériences), une double, plus large, (à droite sur la figure) et une simple, plus étroite (à gauche sur la figure).

Tout comme pour la première série d'expériences, l'ordre de présentation des conditions et de chemins aux sujets a été aléatoire.

3.2.4.2. Résultats obtenus

Le temps moyen pour la première condition (sans aide) a été de 51,7 s. Pour la deuxième condition (avec zoom), il a été de 49s. Cette différence n'est pas significative ($p=0,71$).

Quant au nombre de collisions avec l'environnement, il a été nul, grâce à l'activation de l'évitement d'obstacle sur le robot.

Le nombre de fois que le robot n'a pas été déplacé, en attendant que l'opérateur reçoive des images plus récentes (le comportement de type « move-and-wait ») a été lui aussi comparable pour les deux conditions.

Le pourcentage temps que le robot s'est retrouvé dans cette condition d'arrêt (vitesse nulle) a été de 21% pour le premier cas et de 12% pour le deuxième. En effectuant une analyse de la variance (ANOVA), on trouve qu'on peut rejeter l'hypothèse nulle ($p = 0,04$).

3.2.4.3. Discussion des résultats

L'activation de l'évitement d'obstacles a permis aux opérateurs d'obtenir un temps comparable pour parcourir les trajets proposés pour les deux conditions. Le nombre de fois que le robot a été arrêté est aussi comparable dans les deux conditions.

L'aide proposée a eu un effet sur le temps passé à l'arrêt. Quand elle était activée, les opérateurs ont pu se décider beaucoup plus rapidement, comme le montre le pourcentage du temps total pendant lequel le robot s'est retrouvé à l'arrêt. Ceci signifie que les opérateurs ont eu confiance dans les informations fournies par l'aide et n'ont pas attendu, comme pour le cas où l'aide n'était pas active, la réception des images correspondant sûrement à la position arrêtée du robot.

L'évitement d'obstacles opère sur les vitesses du robot réel. Le robot virtuel n'est pas informé sur les modifications effectuées par l'évitement, lesquelles ne sont pas réalisées tout le temps, mais uniquement quand le robot est trop proche des obstacles. Par conséquent, si l'évitement d'obstacles modifie de façon significative les consignes de l'opérateur, il y aura une différence non-négligeable entre l'image affichée sur l'interface de contrôle (basée sur les ordres envoyés par l'opérateur et une image reçue avec un certain délai) et l'image que le robot enverra, une fois que la commande modifiée par l'évitement d'obstacles sera exécutée. Si ce désaccord entre la prédiction et le retour avait eu un impact significatif sur les opérateurs, on aurait observé une gêne des opérateurs, donc le retour d'un comportement de type « move-and-wait ». Mais, comme on l'a déjà remarqué, ceci ne se passe pas, ce qui signifie que l'aide proposée est assez robuste pour cacher les délais, même s'il y a un écart entre l'image prédite et l'image qui sera reçue. En fait, l'image envoyée par le robot tient compte du

mouvement du robot réel et intègre donc l'évitement d'obstacles. Ainsi, si le retard n'est pas trop important (nous nous sommes limités à 2 secondes), la correction de trajectoire est suffisamment faible pour ne pas perturber l'opérateur.

3.3. Conclusions sur l'utilité de l'aide évaluée

Les résultats obtenus montrent que l'aide proposée permet de masquer les délais et d'obtenir une commande du robot plus fluide.

Lorsque l'aide était activée (sans évitement d'obstacles), on a constaté une amélioration du temps de parcours par rapport à la condition quand l'aide n'était pas présente (et que le délai était présent, bien sûr). Ceci est principalement dû au fait que l'opérateur arrête le robot moins souvent (réduction du comportement de type « move-and-wait »). Le souci est que les opérateurs peuvent enregistrer des collisions avec l'environnement, surtout s'il y a des obstacles qui ne sont pas visibles par la caméra du robot. Dans un système réel, un évitement d'obstacle empêchera ces collisions. La deuxième série d'évaluations a utilisé un tel évitement.

La deuxième campagne d'évaluations a associée l'aide proposée avec un évitement d'obstacles. Après la période d'accommodation avec le contrôle du robot, les sujets ont eu confiance dans les capacités du robot à éviter les obstacles sans leur intervention, donc, même si l'image était en retard, l'évitement d'obstacles faisait que le robot revenait vers une trajectoire acceptable. La conséquence de l'exploitation de cet évitement d'obstacles par les opérateurs est que les temps moyens ont été similaires dans les deux conditions.

En ce qui concerne le comportement de l'opérateur par rapport au retour vidéo, il n'a pas eu besoin de garder le robot à l'arrêt si souvent dans la condition où l'aide et l'évitement d'obstacles étaient actifs simultanément. Comme pour la série d'expériences précédente (sans l'évitement d'obstacles), l'opérateur arrive plus facilement à suivre l'évolution du robot dans le monde et donc l'arrête moins souvent.

Ces résultats montrent que l'aide proposée arrive à masquer les délais. Un comportement de type « move-and-wait » sera beaucoup moins fréquent pendant le pilotage du robot. Une conséquence est que les opérateurs ne s'arrêtent pas pour vérifier la position du robot par rapport aux obstacles qui l'entourent, surtout pour essayer de localiser ceux qui se trouvent en dehors du champ visuel de la caméra embarquée. Cette conséquence n'est pas grave si le robot est muni d'un évitement d'obstacles. Les résultats obtenus pendant ces deux séries d'évaluations ont été présentés dans [Nadrag 2010]. Leur impact sur une personne âgée (à côté de laquelle se robot se trouvera) a été présenté dans [Maseda 2011].

3.4. Perspectives

Le bon fonctionnement de cette technique dépend de plusieurs facteurs. La prédiction de la trajectoire du robot est importante, d'où la nécessité d'avoir un bon modèle implémenté sur l'interface de contrôle.

L'évaluation correcte du délai est aussi importante à effectuer. L'Internet n'est pas un milieu de transport de l'information déterministe. Dans le meilleur des cas, on peut avoir une information sur le délai dans une seule direction. Ceci peut être achevé en synchronisant les

horloges du robot et de l'ordinateur qui héberge l'interface de contrôle. Ensuite, chaque paquet de données transmis devra être daté.

Les hypothèses qu'on a faites sur l'environnement sont aussi contraignantes pour l'utilisation de l'aide. Pour rappel, on a supposé que le robot se trouve dans un bâtiment, soit une maison, un appartement ou des bureaux. Ceci implique une limite supérieure pour la taille des pièces. La représentation incorrecte des occlusions visuelles est moins gênante pour les courtes distances, comme celles qu'on retrouve à l'intérieur des bâtiments.

La prise en compte du modèle de la caméra présente sur le robot dans le calibrage de la transformation de la dernière image reçue du robot est susceptible d'améliorer la prédiction de l'image qui sera affichée sur l'interface de contrôle de l'opérateur.

Si, à cause des problèmes de réseau, l'intégralité des commandes envoyées au robot n'est pas reçue par celui-ci, le décalage entre l'image prédite et celle réelle peut devenir assez important pour qu'il soit remarqué par l'opérateur. S'il n'y a que quelques commandes qui sont perdues, ceci n'est pas grave, grâce au fait qu'on envoie des commandes au robot avec une fréquence élevée (des dizaines d'hertz). Dans le sens inverse (de robot vers l'opérateur), la situation est plus délicate. Ceci est dû au fait que les images sont transmises à une fréquence maximale de vingt hertz, mais ceci peut baisser en dessous de dix, et, donc, si on perd une, il y a plus de chance que l'opérateur se rende compte de la disparité entre les commandes qu'il pense avoir envoyée et l'affichage (à cause d'une perturbation plus prononcée de celui-dernier). Pour une perturbation ponctuelle, cette disparité, même si elle est détectable par l'opérateur, n'est pas gênante pour l'ensemble de la téléopération. Le système d'aide est capable, grâce au fait qu'il travaille sur une période courte de temps et non pas sur un long historique, de recorréler les commandes envoyées avec la modification de la vidéo (supposant que la perturbation a cessé).

Les êtres ou objets qui sont en mouvement dans le champ visuel de la caméra peuvent générer des différences dérangeantes entre l'image prédite et l'image réelle. Plus leur vitesse sera importante, plus cette différence sera dérangeante. Si on arrivait à les distinguer de l'arrière-plan et faire une bonne prédiction sur leurs mouvements, on pourrait envisager une modification de l'aide, pour les prendre en compte. Un modèle de ces êtres ou objets, avec leur position courante dans l'environnement lointain et une identification de l'arrière-plan améliorerait la prédiction de la scène pour ce type de situations.

Commande en environnement inconnu

Dans la partie précédente de cette thèse on a montré une technique permettant de téléopérer un robot en présence d'un délai moyen (entre 300 ms et 2 s). Dans cette partie on va analyser le cas d'un grand délai (supérieur à 2s).

On considère que le délai est trop important pour une téléopération confortable. Le téléopérateur habituel sera obligé d'adopter un mode saccadé de téléopération (il fera avancer le robot sur une petite distance, ensuite il attendra la réception des nouvelles données). Ce mode saccadé est souvent appelé « move and wait ». Afin d'éviter que ceci se passe et pour rendre plus facile la tâche de l'opérateur, un nouveau mode de commande sera proposé.

L'idée derrière ce mode est que l'opérateur indiquera au robot un point à atteindre, et celui-ci sera chargé soit de déterminer le plus court chemin pour l'atteindre, soit d'alerter l'opérateur sur le fait que le point indiqué n'est pas atteignable, suite à une exploration éventuelle de l'environnement. Ce mode peut être aussi utilisé avec un délai faible ou moyen, bien sûr.

On souhaite que le laps de temps entre l'envoi de la position à atteindre et le début du déplacement du robot soit minimal. Le robot doit se localiser, planifier et exécuter la trajectoire déterminée simultanément. On veut que toutes ces tâches s'effectuent sur le robot lui-même. Ainsi, en cas de délai entre le site esclave et le site maître, le robot n'aura pas de problèmes à fonctionner normalement. L'obtention d'un plan de l'environnement est nécessaire pour la localisation et la planification, afin de lui permettre d'atteindre le point spécifié par l'opérateur.

1. Description du mode de commande

L'idée centrale de ce mode de commande est que l'opérateur envoie une destination au robot, que celui-ci est ensuite responsable d'atteindre. Ainsi, on décharge partiellement l'opérateur du contrôle du robot et il peut faire d'autres tâches en parallèle. Bien sûr, il peut toujours intervenir, soit en arrêtant le robot, soit en lui envoyant une nouvelle destination (avant que celui-ci ait réussi à atteindre la précédente). Par rapport au mode de commande utilisé dans la première partie de cette thèse (le mode manuel), l'opérateur n'envoie pas des commandes (position x , y à atteindre) en continu au robot : pour ce mode les commandes sont envoyées occasionnellement.

Une fois un ordre reçu, le robot doit déterminer comment faire pour se rendre à la position souhaitée. Un chemin est planifié afin d'atteindre la destination. Ce chemin est calculé sur un plan de l'environnement du robot. Dans un premier temps, le robot ne se base que sur un plan issu d'un seul scan laser (il n'a pas encore avancé vers son objectif, pour découvrir plus l'environnement). Si l'objectif est suffisamment loin du robot, il sera situé au delà de la possibilité de détection du capteur laser. Dans cette situation, une partie du chemin passera par une partie de l'environnement encore inconnue. Au fur et à mesure que le robot avance, ce chemin est replanifié, s'il y a besoin (à cause de la détection des obstacles invisibles auparavant, par exemple). La cartographie et localisation simultanées (SLAM - simultaneous localization and mapping) est utilisée, à la fois pour acquérir le plan de l'environnement (utilisé pour la planification de la trajectoire) et aussi pour connaître la position du robot (pour

s'arrêter si l'objectif a été atteint et pour corriger l'odométrie issue des codeurs situés sur les roues du robot).

La caméra embarquée sur le robot n'a pas le même champ visuel qu'un humain. Une conséquence de ceci est le fait que l'opérateur ne voit ni les obstacles qui sont sur les côtés du robot, ni les obstacles se trouvant juste devant le robot, mais qui, à cause de leur taille et de leur rapprochement du robot, sont sortis du champ visuel vertical de la caméra. L'évitement d'obstacles présent sur le robot a été conçu comme une sécurité pour la téléopération.

Une première façon d'implémenter cet évitement est de le faire de manière réactive. Cette façon est basée exclusivement sur la distance du robot par rapport aux obstacles. Plus le robot est près des obstacles, plus l'évitement le guidera dans la direction opposée (si les obstacles se trouvent sur les côtés du robot). Si le robot rencontre des obstacles devant lui, l'évitement réduira sa vitesse linéaire, pouvant même l'arrêter.

Une deuxième façon d'effectuer l'évitement des obstacles est de prendre en compte les consignes reçues par le robot. En faisant ceci d'une façon réflexive, il est possible de démarrer la procédure d'évitement avant que le robot s'approche trop des obstacles. Ceci est fait en évaluant le temps jusqu'à la collision avec les obstacles. Plus le robot se dirige vite vers un obstacle, plus l'évitement sera actionné tôt. Un autre avantage est celui de pouvoir déplacer le robot parallèlement aux murs, même si la distance qui le sépare d'eux est très petite. En utilisant l'évitement réactif, ceci serait plus gênant, voire impossible. Supposons que seule la vitesse linéaire soit envoyée comme consigne au robot. Si celui-ci n'est pas orienté parallèlement aux murs (et si la trajectoire est suffisamment longue, bien entendu), le robot se rapprochera trop d'un mur (mais la direction empruntée entraînera une collision beaucoup plus loin), l'évitement le fera dévier, il s'approchera de l'autre mur, d'où il sera dévié, en ainsi de suite. Ce type de comportement (comparable à un rebondissement) est beaucoup atténué si l'évitement prend en compte les consignes reçues. Le suivi de près des obstacles (la navigation dans un couloir, par exemple) ou le passage à ras des objets n'est plus un souci avec ce type d'évitement.

Le mode présenté dans cette partie repose encore plus sur l'autonomie du robot. Il est responsable de générer un plan pour accomplir le souhait de l'opérateur (atteindre un certain point) et de modifier ce plan, si besoin. Après la réception de la consigne envoyée par l'opérateur, le robot analyse son environnement et génère une première version de la trajectoire qu'il doit suivre pour atteindre l'objectif. S'il lui est impossible de suivre la trajectoire calculée, une nouvelle trajectoire sera choisie (si le robot considère toujours que l'objectif est atteignable). Une utilisation possible de ce mode serait sur une courte distance (comparable à la longueur maximale des mesures des capteurs de distance du robot). Sur cette distance, l'odométrie du robot est assez précise. La destination pourrait servir comme un attracteur et le robot utilise uniquement les informations les plus récentes concernant les obstacles pour la planification. Cette approche est, malheureusement, susceptible de se retrouver piégée par des minima locaux, ainsi que de ne pas se rendre compte quand la destination n'est pas accessible (à cause d'un manque d'un historique sur les endroits déjà visités). La solution proposée consiste à rajouter cet historique au plan de la partie de l'environnement déjà explorée et de le prendre en compte dans l'étape de planification.

L'extension de ce mode rajoute deux problèmes théoriques : la localisation du robot (l'odométrie n'étant pas fiable sur des distances importantes) et la planification de la trajectoire. La solution consiste dans la génération d'un plan de l'environnement par le robot lui-même, plan qu'il utilise également pour sa localisation. Cette génération du plan et la localisation du robot dedans font l'objet de la cartographie et localisation simultanées (SLAM). Une contrainte supplémentaire, pratique, cette fois, est que le robot doit répondre en temps réel aux demandes de l'opérateur(dû à des soucis d'acceptabilité de ce mode par les opérateurs).

2. Étude bibliographique

Le mode de commande présenté dans ce chapitre a nécessité plusieurs directions de recherche. Le but a été, d'abord, de mieux comprendre le fonctionnement de différents algorithmes d'évitement d'obstacles (2.1). Ensuite, les différentes façons de représenter le plan de l'environnement ont été analysées (2.2). Sa génération et la localisation du robot (SLAM) ont été étudiées par la suite (2.3). La dernière partie de l'étude a été consacrée à la planification en environnement inconnu (2.4), pour apprendre plus sur les différentes façons de conception des systèmes entiers. Un tel système est capable de planifier seul son chemin vers la destination qui lui a été envoyée, de le suivre et de l'adapter, en fonction de l'évolution du terrain (différences entre les suppositions faites pendant la planification de la trajectoire et ce que les capteurs détectent, une fois sur place). La planification de la trajectoire, ainsi que la correction de la position du robot (donnée par son odométrie), nécessitent l'acquisition d'un plan de l'environnement (SLAM).

2.1. Évitements avancés d'obstacles

Cette section présente différentes façons de suivre une consigne envoyée par l'opérateur, en prenant également l'environnement en compte.

Historiquement, [Borenstein 1991] représente une extension de la méthode du champ de force virtuel. Cette méthode utilise une grille-histogramme. Chaque case contient une valeur de certitude sur la présence d'un obstacle à cet endroit-là. Pour des capteurs ultrason, seule la case qui se trouve dans le centre du cône de détection est incrémentée. En utilisant une fenêtre active autour du robot, chaque case de cette fenêtre exercera une force sur le robot. Cette force est directement proportionnelle à la valeur de certitude de la case et inversement proportionnelle avec la distance de la case au robot. Une force attractive constante est aussi appliquée sur le robot, dans la direction de l'objectif qu'il doit atteindre. Des problèmes avec cette méthode peuvent apparaître dans des situations comme le passage de porte, par exemple, où la force répulsive générée par le cadre de la porte éloignera le robot. D'autres problèmes sont générés par des puits de potentiels (minima locaux), d'où, sans traitement spécifique (recuit simulé, par exemple), le robot ne peut pas sortir. Un autre problème est induit par le fait qu'on utilise des cases discrètes, ceci pouvant entraîner des fortes variations de la force résultante. Un filtre peut lisser ces variations, mais il introduira un délai dans la réponse aux nouveaux obstacles. Un dernier souci est l'émergence d'un comportement oscillatoire pendant la traversée d'un couloir étroit, si le robot n'est pas parfaitement centré. La méthode de

l'histogramme de champ de vecteurs (VFH – vector field histogram) est censée améliorer ou résoudre tous ces soucis. Les auteurs identifient comme source de problèmes précédemment décrits la réduction brutale de données à deux éléments : une direction et une magnitude pour la force. La nouvelle méthode emploie une réduction à deux étapes (par rapport à une seule, utilisée par la méthode précédente). Ainsi, trois niveaux pour la représentation des données existent : au plus haut niveau on a une représentation détaillée de l'environnement du robot. Au niveau intermédiaire on retrouve un histogramme polaire centré sur la position courante du robot. Au plus bas niveau on retrouve le résultat du VFH – les valeurs pour la vitesse linéaire et angulaire du robot. Étant donné que ce planificateur est local, il est possible qu'il ne trouve pas le chemin optimal. Il est aussi possible que le robot rentre dans une boucle. Pour résoudre ces soucis, les auteurs utilisent un moniteur de la trajectoire. Si le robot dévie de la direction vers l'objectif, le moniteur enregistre la direction de la déviation et essaie de ramener le robot vers la bonne direction. Des tests avec un robot réel ont été effectués (dans un environnement de type intérieur). Un exemple de trajectoire est montré dans la Figure 24.

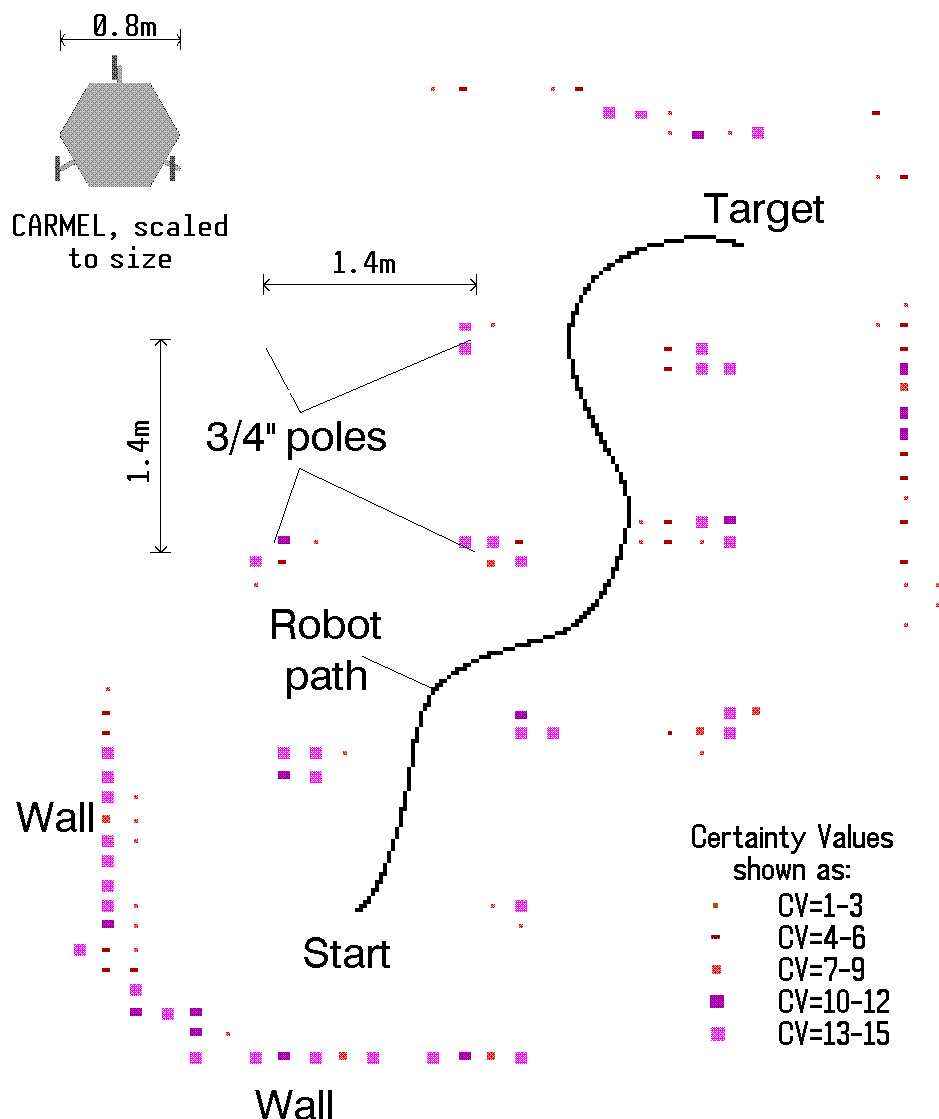


Figure 24 Trajectoire avec le robot réel [Borenstein 1991]

Une amélioration de l'algorithme présenté dans le papier précédent est décrite dans [Ulrich 1998]. Une grille-histogramme est utilisée comme donnée d'entrée pour l'algorithme. Un processus de réduction en quatre étapes est employé pour obtenir une direction pour le mouvement. Pendant les trois premières, le plan est réduit à des histogrammes polaires unidimensionnels, centrés sur la position courante du robot. Pendant la quatrième étape, l'algorithme sélectionne la meilleure direction en s'appuyant sur l'histogramme polaire masqué et une fonction de coût. Pendant la première étape (la génération de l'histogramme polaire primaire), les dimensions du robot sont prises de façon explicite en considération (dans la version précédente, elles ne l'étaient pas). Pendant la deuxième étape (l'histogramme polaire binaire), dans la version précédente, le seuil est fixe, fait qui pourrait générer des problèmes quand la distance entre les obstacles est à la limite de la traversabilité. Ce problème est résolu en utilisant deux seuils, l'un pour la valeur maximale et l'autre pour la valeur minimale. A la place des densités polaires, les secteurs polaires sont maintenant binaires (libre ou occupé). Pour la troisième étape (le calcul de l'histogramme polaire masqué), la version précédente ignorait la dynamique et la cinématique du robot. Ensuite, une approximation de la trajectoire du robot est utilisée. On suppose que le robot se déplace sur des arcs de cercle. Plus le robot avance vite, plus la courbure maximale est réduite. Dans la quatrième étape, une direction est choisie. La méthode précédente favorise la direction de l'objectif. La méthode présentée trouve toutes les directions possibles et ensuite détermine un ensemble de directions-candidates. La direction qui présente le coût minimal est choisie. La fonction de coût utilise trois paramètres : la direction par rapport à l'objectif, la nouvelle direction par rapport à l'orientation courante du robot et la différence entre la nouvelle direction et la direction précédemment empruntée. Le premier paramètre encourage un mouvement vers l'objectif. Les deux autres paramètres encouragent le robot à s'engager dans une certaine direction. Ces deux derniers paramètres fonctionnent comme une mémoire à court terme. Le deuxième paramètre fait office de mémoire mécanique. Le troisième paramètre encourage le robot à garder une direction constante. L'intérêt d'utiliser cette fonction de coût est évident quand le robot s'approche d'un obstacle. Sans elle, le robot pourrait heurter l'obstacle en hésitant entre un évitement par la gauche et un évitement par la droite. Un autre avantage est que le comportement du robot peut être changé facilement en modifiant soit les poids des paramètres, soit la fonction elle-même. Cette méthode a été implémentée sur un robot réel. Les auteurs ont observé que cette méthode est plus fiable que la précédente, étant même capable de gérer des coins. Un désavantage, à cause de sa nature locale, est que le robot peut se retrouver, parfois, enfermé dans des impasses.

Ce problème de limitation locale est résolu dans [Ulrich 2000], avec une nouvelle version de l'algorithme VFH. Pour chaque position-candidate, on considère que le robot a avancé dans cette direction et une nouvelle analyse des directions possibles est effectuée. Cette étape est répétée un certain nombre de fois. Pour chaque position finale, son coût est calculé. Ce coût représente la somme des coûts des branches qui la caractérisent. Le coût pour le premier nœud exploré est celui décrit dans le papier précédent [Ulrich 1998] (la fonction de coût). Pour les nœuds suivants, les auteurs utilisent un coût basé sur la bonne orientation vers l'objectif. Ils proposent également un coût associé avec la direction effective de mouvement, mais ils ne l'utilisent pas à cause temps de calcul requis (le robot doit se déplacer en temps réel).

Dans [Fernandez 2004] une autre façon d'avancer vers une cible est décrite. Les obstacles sont définis comme des secteurs circulaires. Une réduction est ensuite effectuée, pour réunir les obstacles qui se recouvrent partiellement. Chaque secteur candidat est évalué en fonction de sa distance par rapport à l'objectif et l'erreur angulaire entre l'objectif et le faisceau analysé. La Figure 25 montre une comparaison avec d'autres méthodes (CVM et LCM). CVM signifie Curvature Velocity Method, LCM signifie Lane Curvature Method et BCM signifie Beam Curvature Method, la méthode développée par les auteurs.

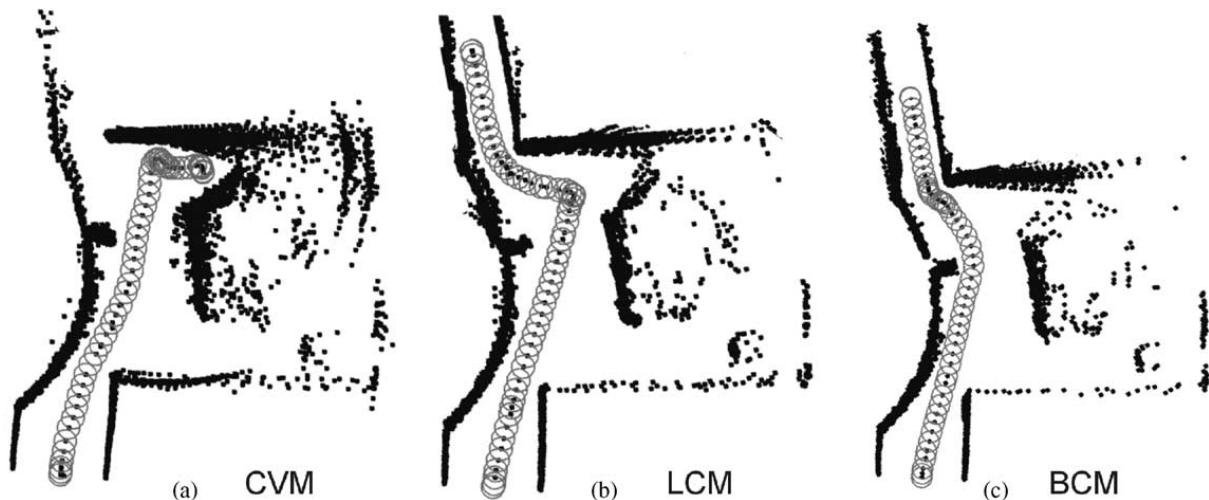


Figure 25 Comparaison entre différents algorithmes

Des secteurs circulaires sont aussi employés par [Minguez 2004]. L'état futur du robot peut être caractérisé par l'une des cinq situations suivantes :

1. sécurité faible 1 : les obstacles sont trop près du robot d'un seul côté
2. sécurité faible 2 : les obstacles sont proches du robot des deux côtés
3. sécurité haute et destination proche : la destination est visible et les obstacles sont éloignés des deux côtés sur la trajectoire (une droite) pour l'atteindre
4. sécurité haute et région large : le robot n'a pas d'obstacles proches dans la direction considérée et la largeur de la direction considérée est grande
5. sécurité haute et région étroite : le robot n'a pas d'obstacles proches dans la direction considérée et la largeur de la direction considérée est petite

Pour le premier cas, la direction du robot est calculée en fonction de la position de l'objectif et de l'angle par rapport à l'obstacle le plus proche.

Pour le deuxième cas, la direction du robot est calculée en fonction de la bissectrice de l'angle des obstacles les plus proches sur les deux côtés.

Pour le troisième cas, le robot se déplace directement vers l'objectif.

Pour le quatrième cas, la direction est donnée par une combinaison de la direction vers l'objectif et un angle donné. Ceci engendrera un mouvement qui sera légèrement en biais vers l'obstacle.

Pour le dernier cas, la direction du robot est calculée en fonction de la bissectrice de l'angle des plus proches obstacles sur les deux cotés.

La Figure 26 montre un parcours-type. Le robot a évité d'entrer dans les zones en forme de U parce que la méthode d'évitement d'obstacles a employé la recherche d'espace libre pour éviter ce type de pièges.

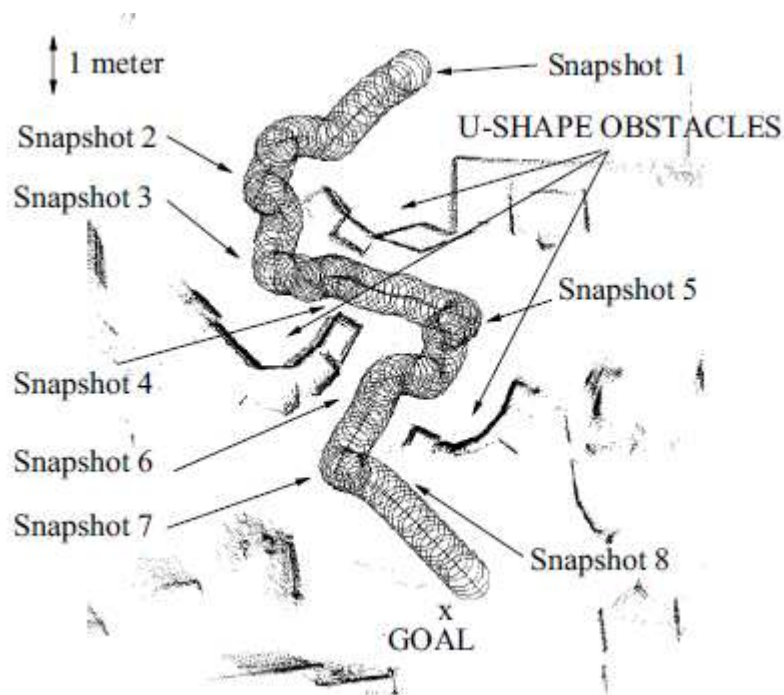


Figure 26 Parcours avec le robot réel [Minguez 2004]

Un problème avec cet algorithme est le fait qu'il a été conçu pour des robots circulaires. Comme seule l'information locale est utilisée, l'algorithme ne peut pas garantir la convergence globale sur la destination.

Des cônes de collisions (directions interdites) sont employés par [Becker 2006]. Les vitesses atteignables par le robot pendant le cycle suivant sont représentées sous la forme d'un parallélogramme. Le robot essaie de maintenir sa vitesse constante, en la choisissant parmi celles proposées par le parallélogramme.

La fenêtre dynamique est utilisée par [Fox 1997]. Elle correspond à l'espace que le robot pourra atteindre dans l'intervalle de temps suivant. Son avantage est de réduire la taille de l'espace où une solution pour le déplacement suivant sera recherchée. Pour déterminer la meilleure direction, une fonction-objectif, composée de la direction de la cible, la distance par rapport aux obstacles et la vitesse du robot, est utilisée. Le facteur correspondant à la distance par rapport aux obstacles prend en compte uniquement l'obstacle le plus proche et, si celui-ci est trop éloigné, sa valeur est remplacée par une constante.

Une extension de la fenêtre dynamique à une échelle globale est faite dans [Brock 1999]. Le calcul est fait exclusivement dans la fenêtre globale, l'algorithme étant assez rapide pour qu'il soit effectué à chaque cycle. Des connaissances a priori sur l'environnement ne sont pas requises pour le bon fonctionnement de la fenêtre globale. La fonction de navigation NF1,

basée sur la propagation d'ondes, est employée. Sa particularité est que l'objectif du robot sera le seul minimum dans l'environnement. A chaque itération de la fenêtre globale, la NF1 est recalculée et une nouvelle commande est envoyée au robot. Le plan de l'environnement est acquis sous la forme d'une grille d'occupation. Des tests effectués avec un robot réel montrent le bon fonctionnement et les résultats obtenus avec cette méthode.

2.2. Génération des plans

Afin de pouvoir naviguer efficacement en mode autonome dans un environnement inconnu, le robot a besoin de mémoriser le chemin déjà parcouru. En cas de souci (arrivée dans une impasse), un nouveau chemin pourra être généré, en s'appuyant sur le chemin déjà parcouru. Bien entendu, si l'opérateur a plus d'informations sur l'environnement que le robot, il pourra le diriger sur le bon chemin, avant que celui-ci arrive dans une impasse ou, si plusieurs chemins sont possibles, d'emprunter celui qui intéresse le plus l'opérateur (le robot essaie d'utiliser le plus court chemin pour arriver à son objectif).

Dans cette partie de l'étude bibliographique je présente différentes façons de générer des plans de l'environnement. On s'est concentré sur des cartes basées sur des grilles d'occupation, parce que l'évitement d'obstacles est plus facile à réaliser en utilisant directement ce type de cartes qu'en utilisant des cartes topologiques.

[Elfes 1989] est un des premiers auteurs à utiliser les grilles d'occupation probabilistes. Il est possible d'utiliser des modèles probabilistes des capteurs dans la construction de la grille. Un exemple (pour un capteur Gaussien bidimensionnel) est montré dans la Figure 27. Le plan précédemment obtenu est actualisé en fonction des mesures courantes des capteurs. Les cases sont initialisées à 0,5 (pas d'information sur leur état).

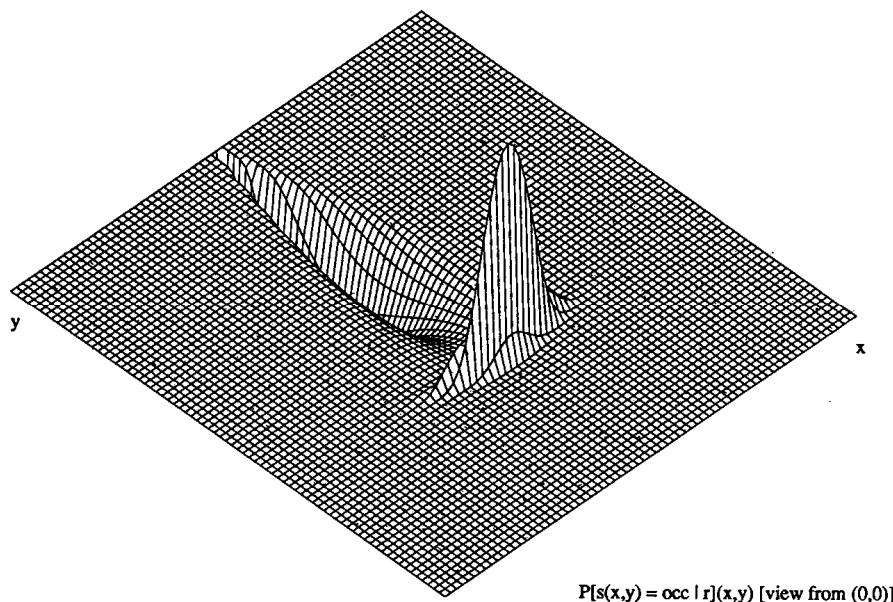


Figure 27 Capteur probabiliste gaussien [Elfes 1989]

Parce que l'odométrie du robot n'est pas parfaite, deux manières d'intégrer des nouvelles mesures sont avancées : soit on rajoute de l'incertitude sur les anciennes mesures (méthode centré robot), soit on rajoute de l'incertitude sur la position du robot (méthode centré environnement).

Parce que les traitements qui peuvent être effectués sur la grille ainsi obtenue sont proches de ceux de traitement d'images, les équivalences sont données dans le Tableau 1.

Grilles d'occupation	Images
Classement des cases comme occupée, libre ou inconnu	Seuillage
Gestion de l'incertitude de la position	Floutage / convolution
Élimination des mesures fausses	Filtrage passe-bas
Mise en correspondance des plans/résolution du mouvement	Corrélation multi-résolution
Élargissement des obstacles pour la planification de la trajectoire	Élargissement des régions
Planification de la trajectoire	Suivi des bordures
Extraction des aires occupées, libres ou inconnues	Segmentation/coloration des régions/étiquetage
Déterminer les bords des objets	Détection des bordures
Incorporation des plans pré-générés	Conversion des scans
Prédiction des mesures des capteurs) partir des plans	Corrélation
Détection du mouvement des objets sur des séquences de plans	Filtrage spatio-temporel

Tableau 1 Techniques similaires pour traiter les grilles d'occupation et les images

Dans [Elmenreich 2008] l'auteur montre deux méthodes pour construire une grille d'occupation en utilisant des capteurs peu fiables. La première nécessite que les cônes de trois capteurs ultrasons soient partiellement superposés (Figure 28). La valeur médiane des résultats des capteurs est enregistrée dans la carte. Le souci avec cette méthode est que la bonne détection des obstacles par les capteurs dépend de l'orientation de l'objet par rapport au robot.

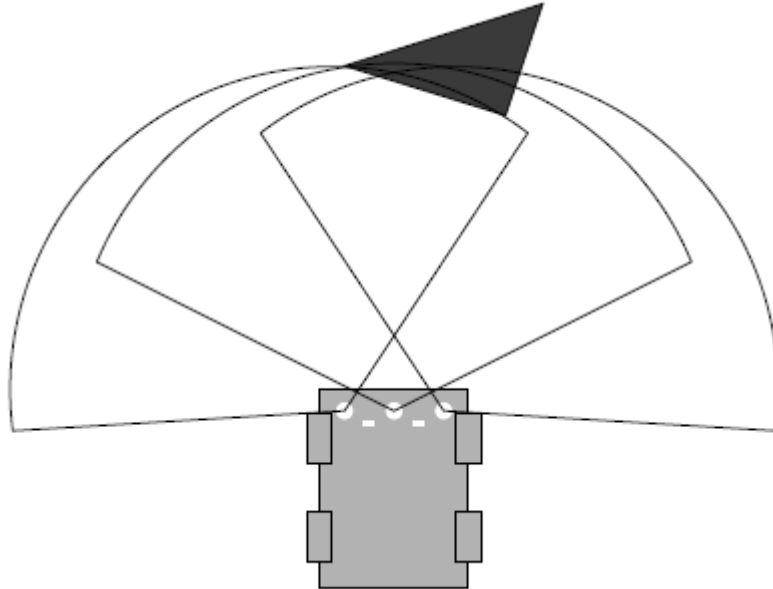


Figure 28 Zone de superposition des capteurs

Pour la seconde méthode, on fait plusieurs hypothèses sur les défaillances des capteurs. D'abord, on les considère comme étant indépendants, la défaillance d'un capteur n'entraînant pas des défaillances dans d'autres. Trois types de défaillances sont identifiés : une défaillance passagère est déclenchée de façon aléatoire et le capteur enregistre une seule mesure dégradée ou fausse. Ce type peut être considéré comme négligeable. Un deuxième type de défaillance est une défaillance intermittente (à cause d'un miroir d'un capteur laser couvert par la poussière, par exemple). Le dernier type est une défaillance permanente, qui nécessite une intervention explicite pour réparer le capteur. En ce qui concerne la grille d'occupation générée, chaque case aura un capteur attaché. La valeur donnée par ce capteur sera retenue (avec prise en compte de la valeur précédente) dans la case.

Pour cette méthode, chaque capteur produira sa grille d'occupation indépendamment des autres capteurs. Le plan sera construit en fusionnant toutes ces grilles. Chaque case finale doit être vue par au moins trois capteurs pour qu'elle soit actualisée. L'identifiant du capteur considéré est aussi enregistré pour la case actualisée.

L'algorithme de mise à jour est le suivant : si la case n'a pas de capteur attaché ou si le capteur attaché est le même que le capteur courant, alors elle est actualisée. Si le capteur est différent, on compare la valeur associée à la case avec celle fournie par le capteur. Si elle est au dessus d'un seuil de confirmation, les confiances dans le capteur actuel et le capteur qui avait précédemment actualisé la case sont accrues. Si la différence est au dessous d'un seuil d'infirmité, les confiances dans le capteur actuel et le capteur qui avait précédemment actualisé la case sont diminuées. Si la comparaison n'est pas significative, on n'a ni une confirmation, ni une contradiction. La nouvelle valeur de la case est une moyenne pondérée entre l'ancienne valeur et la mesure courante. Les poids sont dérivés à partir des confiances dans les capteurs et de l'importance de la mesure. Plus une mesure est éloignée de 0,5, plus elle est importante. La contribution du capteur à une case donne une idée sur la cohérence de la mesure enregistrée dans cette case et la valeur que le capteur mesure. Si cette contribution

est au-delà d'un certain seuil, le capteur devient le nouveau propriétaire de la case. Sinon, le propriétaire est marqué comme étant inconnu.

Dans [Thrun 2003] l'auteur propose l'utilisation du modèle direct des capteurs pour l'acquisition des plans. De manière classique [Elfes 1989], le modèle inverse des capteurs était utilisé. Le plan était généré à partir des mesures des capteurs. En utilisant le modèle direct, on calcule la vraisemblance des mesures pour tout plan et position du robot, afin de trouver un plan qui maximise la vraisemblance. Les capteurs ultrason sont utilisés pour illustrer ce concept. Le modèle direct utilisé prend en compte deux causes pour les mesures des capteurs : le cas non-aléatoire, quand chaque case comprise dans le cône de mesure d'un capteur a une possibilité de réfléchir le son, et le cas aléatoire, quand le capteur a la possibilité de fournir un résultat aléatoire, généré par une distribution uniforme, répartie sur toute la plage de mesures possibles du capteur. À la suite des expériences en simulation et avec un robot réel (Figure 29), l'auteur constate que l'algorithme proposé offre des performances meilleures évidentes pour des discontinuités dans l'environnement (portes, par exemple), ou quand il y a un fort bruit dans les mesures (capteurs ultrason et des murs lisses). Un désavantage est que les objets en mouvement ne sont pas exclus.



Figure 29 Plan obtenu avec le modèle inverse (gauche) et direct (droite)

Dans la Figure 29, gauche, on ne peut pas observer les portes ouvertes, mais on n'a pas la personne qui se déplace autour du robot (les points dans la figure de droite).

Dans [Thrun 1998] l'auteur propose d'utiliser un plan mixte, métrique-topologique. La représentation topologique est construite à partir d'une grille d'occupation. La planification de la trajectoire du robot est faite à partir de la représentation topologique de l'espace. Pour chaque triplet de points de la représentation topologique, une trajectoire utilisant des points de la grille qui font aussi partie du triplet couramment considéré est calculée. La dernière étape est de déplacer le robot vers sa destination, qui se trouve dans une certaine région topologique. Le plan topologique est toujours consistant avec la grille d'occupation. Pour chaque plan abstrait généré en utilisant le plan topologique, il est possible de trouver un plan correspondant dans la grille d'occupation. Malheureusement, il y a une certaine perte de détails dans la représentation topologique, qui peut entraîner la génération des chemins sous-optimaux. Le grand avantage de ce type de représentation d'environnement est la rapidité avec laquelle un plan d'action est généré.

La mise en correspondance des mesures courantes avec un plan historique, sauvegardé en utilisant les quadrees, est présentée dans [Visser 2009]. Les auteurs considèrent que, à partir d'une mesure laser, trois types d'erreurs sur l'incertitude du déplacement effectif du robot peuvent être faites : à cause du bruit de la mesure, à cause du biais du capteur et à cause de la mise en correspondance de la mesure avec l'historique. La mise en correspondance de deux mesures uniquement revient à retrouver la translation et rotation du robot. Parce que faire ça de façon parfaite n'est pas possible, le problème revient à minimiser cette erreur. En comparant le scan courant avec l'intégration des mesures proches (obtenues à partir de plusieurs scans antérieurs), l'erreur de correspondance est, non plus en fonction de la distance, mais en fonction de la densité des mesures proches. Un quadtree est utilisé pour garder toutes les mesures pertinentes. La pertinence d'un scan est déterminée à partir de seuils (fixes) sur le déplacement estimé. Si la translation et/ou la rotation du robot les dépassent, alors le scan est enregistré dans le quadtree. Une comparaison avec des approches basées sur des techniques de SLAM est également montrée. La comparaison a été faite à partir des bases de données publiques, non pas en utilisant un robot. Un exemple est montré dans la Figure 30.

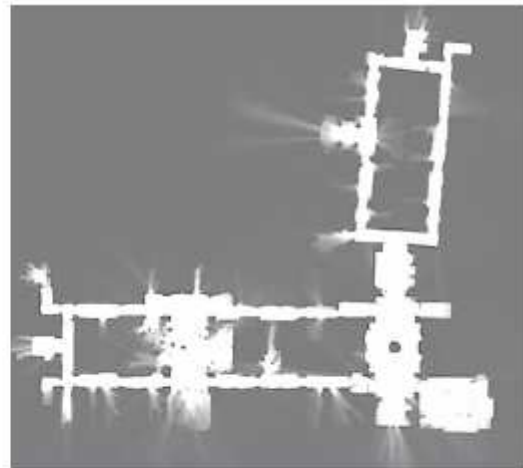
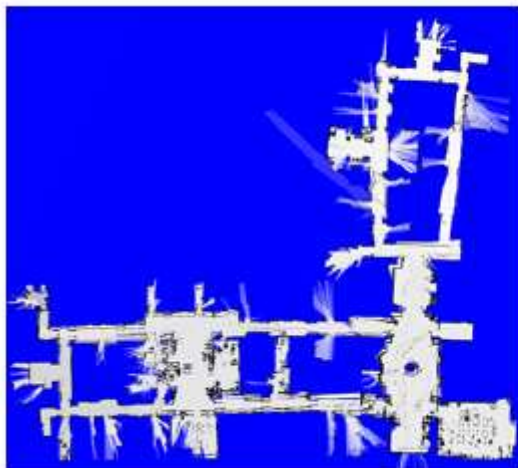


Figure 30 Comparaison avec SLAM

À gauche on peut observer le plan obtenu en utilisant l'algorithme présenté pour la mise en correspondance des mesures. À droite on a le plan produit en utilisant un algorithme de SLAM [CARMEN 2008].

2.3. Cartographie et localisation simultanées (SLAM)

Le chapitre précédent présente différentes façons de construire le plan de l'environnement du robot. Afin de savoir quand le robot a atteint son objectif, on a également besoin de sa position.

Il est possible d'obtenir la position du robot directement à partir de son odométrie. Pour des distances courtes et de petits déplacements, ceci est suffisant. Pour des distances plus longues et, surtout, pour des rotations importantes, l'odométrie n'est pas assez fiable. Il est pourtant possible de la corriger en rapportant le robot à son environnement (qu'on suppose statique). En comparant l'environnement courant (obtenu à partir des mesures laser) avec les observations

précédentes, il est possible d'obtenir son déplacement par rapport à sa position précédente, et de corriger ainsi son odométrie.

L'essence de la cartographie et de la localisation simultanée (SLAM) est de laisser un robot mobile dans un endroit inconnu (on ne connaît pas ses coordonnées), dans un environnement inconnu (on n'a pas un plan de celui-ci), et de faire en sorte que le robot construise un plan de l'environnement et qu'il se localise dedans en parallèle.

Différents capteurs peuvent être utilisés pour faire de SLAM. Des capteurs ultrason, laser ou infrarouges peuvent être utilisés, ainsi que des caméras (stéréoscopiques ou non) ou autres capteurs plus exotiques (Kinect, etc.). Des capteurs externes de position (GPS, balises magnétiques, etc.) peuvent être aussi employés dans le système. Ces capteurs peuvent être utilisés seuls ou à plusieurs, pour essayer d'améliorer la précision.

Un algorithme général pour le SLAM est le suivant :

- on prédit la position du robot, en utilisant la commande moteur et l'odométrie (fiable sur une période courte de temps)
- on interroge le(s) capteur(s) et on traite les données reçues, pour les mettre dans le bon format, si besoin
- on estime la position courante à partir des capteurs extéroceptifs ; ceci est l'apport principal des différentes techniques de SLAM
- on met à jour des informations comme la carte, la position du robot, l'odométrie, etc.

Deux catégories d'approches pour faire du SLAM sont utilisées : celles basées sur des filtres de Kalman (ou dérivées) et celles basées sur des filtres particuliers. Pour le filtre de Kalman, on effectue la prédiction de la position, l'estimation de la position courante et on actualise la carte. Un souci est que des modèles linéaires du mouvement et des observations sont utilisés. Ce problème est résolu pour les filtres particuliers, pour lesquels on effectue la prédiction de la position, on émet des hypothèses sur la position du robot, on sélectionne la meilleure et on actualise la carte.

Un bref historique du SLAM, ainsi qu'une présentation plus détaillée des principales techniques (filtre du Kalman et filtre particulier) sont réalisés par [Durrant-Whyte 2006]. Une liste avec des différentes implémentations de SLAM et des liens pour les récupérer est aussi présentée. Pour tester différents algorithmes de SLAM, des liens vers des bases de données sont aussi fournis.

Dans [Civera 2010] le capteur utilisé pour observer l'environnement du robot est une caméra monoculaire. L'utilisation d'un seul point pour RANSAC (random sample consensus) est possible grâce à l'utilisation des hypothèses sur l'environnement. Les meilleurs résultats sont obtenus quand la caméra a une vitesse constante par rapport à l'environnement. Ce système de SLAM est capable d'utiliser uniquement la caméra comme source d'information, mais aussi d'inclure l'information en provenance de l'odomètre du robot pour améliorer sa précision. La Figure 31 montre un résultat obtenu en utilisant l'odométrie en complément (trajectoire fine), en comparaison avec l'enregistrement GPS (trajectoire épaisse).

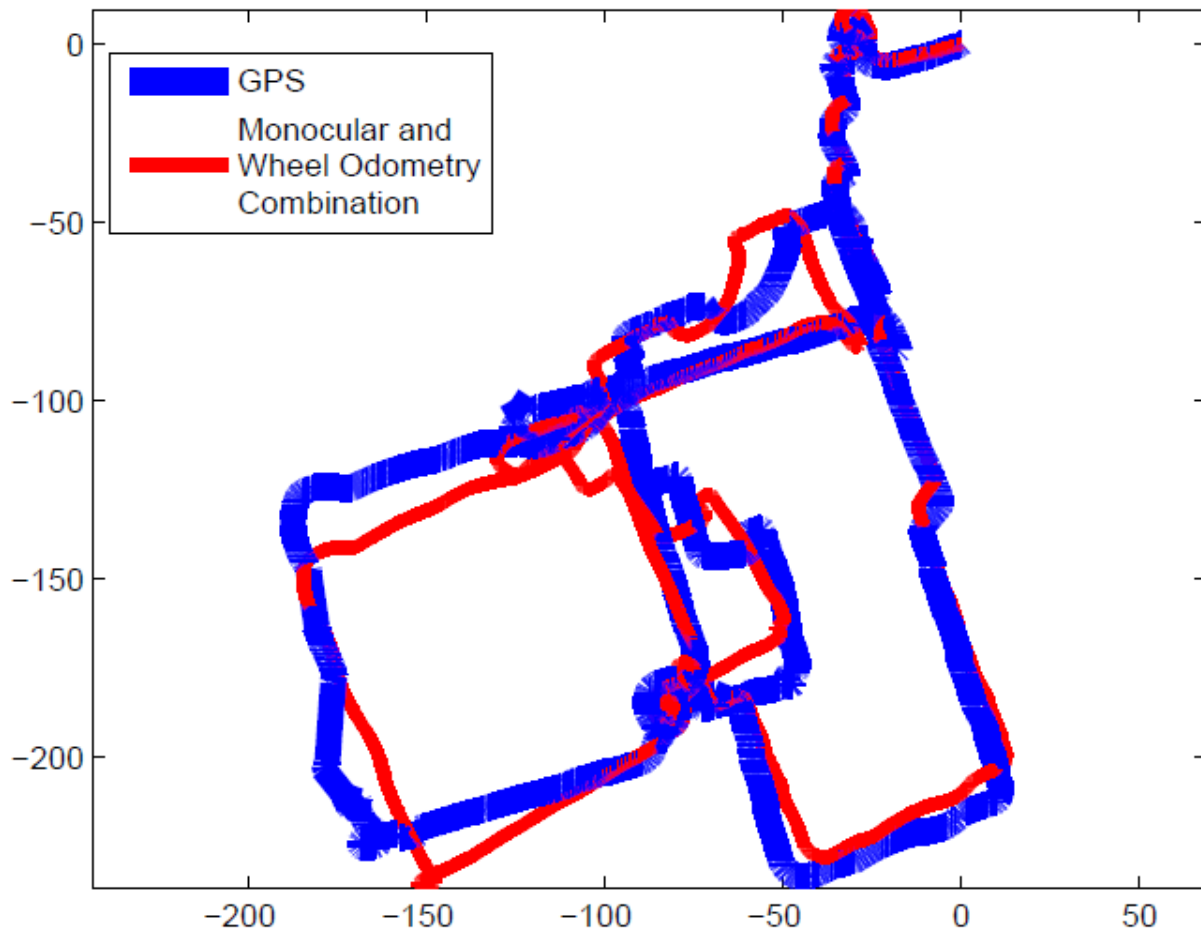


Figure 31 Comparaison SLAM monoculaire et GPS

Dans [Strasdat 2010] une caméra monoculaire est également utilisée comme source pour le SLAM. La fermeture de boucle est effectuée pour chaque image, afin d'améliorer les résultats. Il est possible d'améliorer le plan obtenu, en utilisant plusieurs degrés de liberté (six ou sept). Les meilleurs résultats sont obtenus en utilisant sept (rotations et translations dans l'espace 3D et la taille du plan obtenu).

Dans [Diosi 2004] la fusion des capteurs laser et ultrason est employée pour améliorer la précision de la localisation. Les capteurs ultrason sont capables non seulement de déterminer la distance qui les sépare de l'obstacle, mais aussi de classifier les objets détectés dans des catégories comme angles droits concaves, plans et arêtes/points. Du point de vue des performances, les erreurs sont de $\pm 0,2$ mm et de $\pm 0,1^\circ$ à une distance de trois mètres. L'odométrie est supposée être affectée par un bruit blanc, sur la valeur de l'entraxe et sur la distance parcourue par les roues. Pour les sonars, il s'agit d'un bruit blanc gaussien. Le type de SLAM utilisé est celui basé sur un filtre de Kalman étendu.

La coopération sonar-laser a plusieurs avantages : une augmentation de la précision des mesures, une amélioration de l'exactitude des mesures, une segmentation laser plus robuste et l'élimination des réflexions spéculaires.

En ce qui concerne l'augmentation de la précision des mesures, plus le nombre de mesures agrégées est grand, plus la précision de la détection est grande. Si le même amer est observé à

partir de deux capteurs fonctionnant selon des modalités différentes, son existence est plus probable (d'où l'amélioration de l'exactitude des mesures).

Un autre souci est que, à cause de la précision limitée du laser pour la mesure de la distance et de l'angle, l'observation des différents amers peut être, de façon erronée, segmentée dans une seule ligne. En rajoutant les observations du sonar, le résultat de la segmentation peut être amélioré. Ceci peut éviter quelques erreurs qui se produisent systématiquement, à cause des associations basées sur les mesures laser qui se font de façon incorrecte. L'utilisation du laser peut aider à éliminer les réflexions spéculaires, en comparant les scans ainsi obtenus avec ceux obtenus en utilisant le sonar uniquement.

L'algorithme employé par les auteurs pour la combinaison des deux sources d'informations (laser et sonar) est le suivant : le sonar aide la segmentation effectuée à partir des mesures laser, le laser aide à éliminer les fausses détections produites à partir du sonar et pour la sélection des bons amers de type point. Les amers détectés par le laser et le sonar, correspondant au même objet, sont fusionnés et un historique est gardé pour chaque amer en ce qui concerne le capteur qui y a principalement contribué. Les auteurs sont partis de l'hypothèse que, pour la combinaison des mesures laser et ultrason, les deux capteurs observent les mêmes amers, la valeur de la vitesse du son utilisée pour les calculs est le résultat d'une bonne approximation et que l'odométrie à court terme du robot est suffisamment précise.

En ce qui concerne la segmentation obtenue à partir des mesures, les auteurs gardent une mémoire de 12 s pour le sonar (6 s avant et 6 s après un scan laser). Les mesures obtenues à partir du sonar sont ensuite rassemblées. Les clusters éloignés des détections laser sont ignorés pour la suite du traitement. Pour la segmentation laser, chaque point qui ne fait pas déjà partie d'une ligne est considéré comme étant le début d'un nouveau segment. Ce segment est étendu jusqu'à ce qu'une certaine condition (d'une liste) soit atteinte (discontinuité dans les scans, la suite est un cluster des mesures sonar contenant des coins ou des arrêtes, longueurs supérieures à 70 cm etc.). Si le segment dépasse une certaine taille, alors il est sauvegardé dans la mémoire.

Des clusters basés sur le sonar qui contiennent des coins et des arrêtes qui se trouvent dans la prolongation des segments sont des bons candidats pour des amers de type point. Plus ils sont proches d'une ligne et d'un autre amer de même type, meilleurs ils sont. Des segments extraits à partir du laser, qui se trouvent à côté d'un coin obtenus à partir du sonar, sont ensuite vérifiés pour voir s'ils ne forment pas un angle droit entre eux. Si c'est le cas, ils sont marqués comme faisant partie du coin.

Pour la fusion, un segment ou coin détecté par le laser qui a un segment ou coin détecté par le sonar correspondant déjà présent dans la mémoire est fusionné avec si les temps d'acquisition sont proches. La fusion est faite à l'aide d'un filtre du Kalman. Pour les coins, seules la distance et la position dans le scan sont prises en compte, à cause du fait que le sonar non plus ne peut pas mesurer leur orientation.

L'algorithme de SLAM est basé sur un filtre du Kalman, implémenté sous MATLAB et réalisé hors-ligne. Les segments sont obtenus soit à partir du laser, soit à partir de sa fusion

avec le sonar. Une validation sur une trajectoire mesurant 150 m, avec un robot réel, est aussi présentée.

Dans [Steux 2010] l'algorithme de SLAM présente plusieurs caractéristiques similaires avec ceux qu'on retrouve dans notre contexte : l'algorithme doit être embarqué sur un robot mobile et ses résultats (plan et coordonnées du robot) doivent être générés à une fréquence suffisante pour permettre au robot de se déplacer en temps réel (le robot utilisé dans le papier a une vitesse linéaire maximale de 3 m/s). Cet algorithme est basé sur l'utilisation d'un filtre particulaire. Seul le laser est utilisé comme source de mesures. Un problème potentiel est que la fermeture des boucles n'est pas gérée, mais, pour des trajectoires courtes, à l'intérieur d'un bâtiment, ceci n'est pas un problème. La Figure 32 montre un plan obtenu en utilisant cet algorithme.

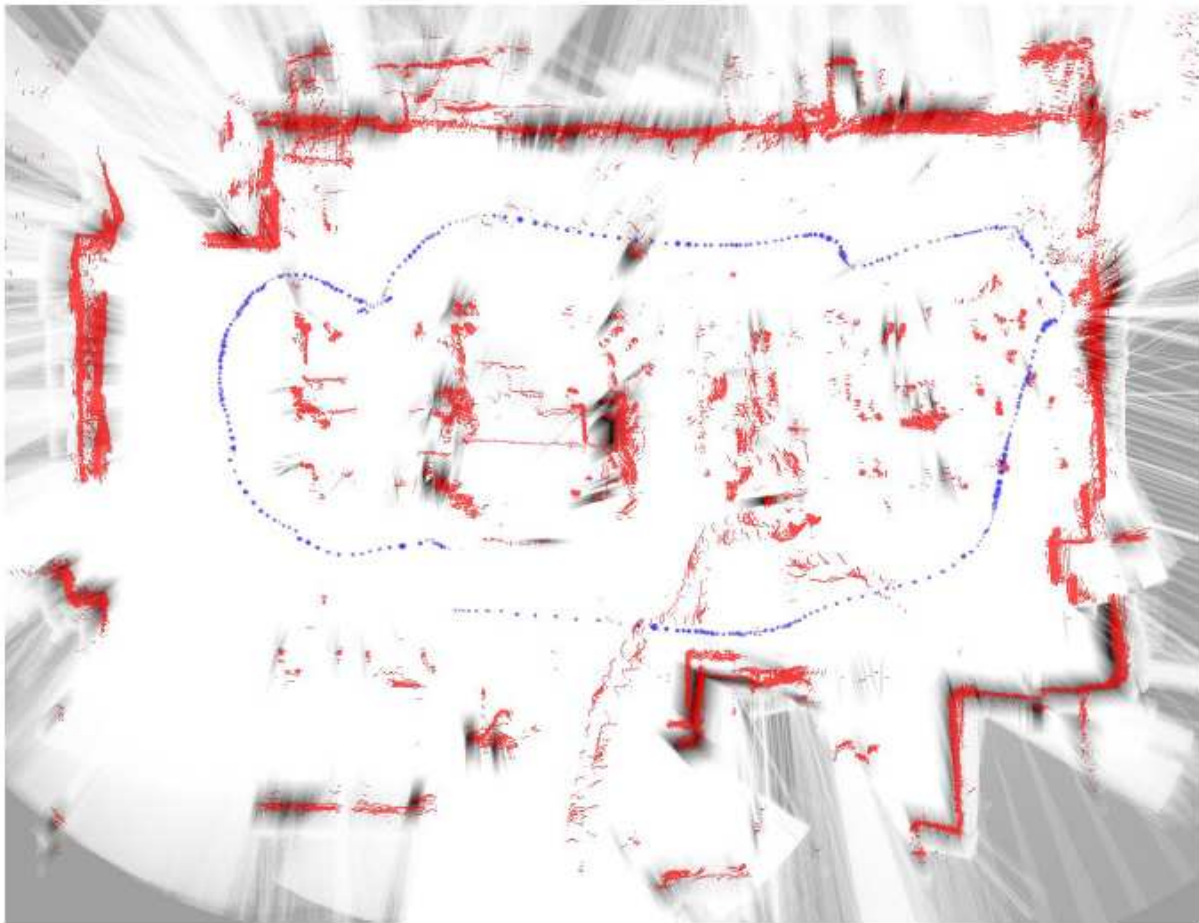


Figure 32 Plan obtenu avec le robot réel

Dedans, la trajectoire parcourue par le robot est présentée en pointilles. Les obstacles détectés sont marqués sur la figure avec des segments. Les zones qui n'ont pas été explorées par le robot (typiquement vers l'extérieur de la figure), où il y a encore une incertitude sur leur état d'occupation, sont marquées à l'aide des surfaces grises.

2.4. Systèmes de planification

L'objectif qu'on s'est fixé est de produire un système qui permettra au robot d'avoir une grande autonomie dans le choix de ses mouvements. Seule la destination (sous la forme des

coordonnées x et y , on suppose que le sol est un plan) est indiquée par l'opérateur. C'est au robot ensuite de calculer le chemin, de le suivre, et, si besoin est, de le modifier, au fur et à mesure qu'il découvre son environnement.

Dans cette section on présente différents systèmes qui sont capables d'accomplir ceci. Le contexte n'est pas forcément le même que le nôtre, l'environnement des robots pouvant être varié (routes, mer, mines, etc.).

Dans [Pradalier 2005] une voiture doit naviguer dans un parking. Le véhicule peut contrôler les quatre roues pour tourner. La localisation du robot et la génération du plan n'est pas faite à partir des caractéristiques naturelles de l'environnement, à cause du fait que leur détection n'est pas fiable et insuffisamment précise. Des balises artificielles sont utilisées (des cylindres réfléchissants). L'erreur ainsi obtenue est de 10 cm pour la position et de 2° pour l'orientation. Pour la planification, une grille d'occupation est utilisée. Dans un premier temps une trajectoire qui ne prend pas en compte les contraintes imposées par le robot (forme et non-holonomie) est calculée. Ensuite, le chemin est approximé par une suite de segments sans collisions. Ces segments sont reliés par des virages faisables. Pour accélérer les calculs, les mesures du laser ont été réduites à huit. Les ordres reçus par le moteur peuvent avoir des priorités différentes : suivre la commande est un ordre faible, éviter les obstacles est un ordre fort. Plus les obstacles sont près, plus le poids de l'évitement croît. Une fois la trajectoire idéale vers l'objectif calculée, le robot essayera d'y revenir, une fois l'obstacle contourné. La Figure 33 illustre ce concept.

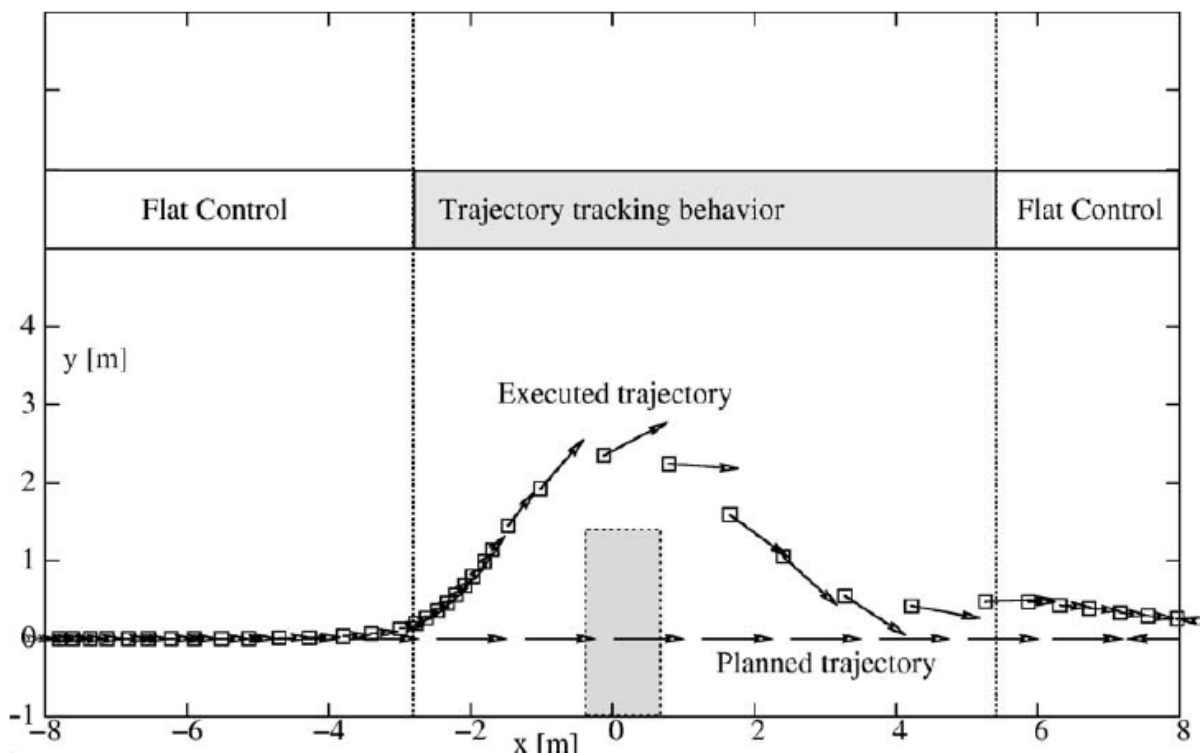


Figure 33 Obstacle sur la trajectoire

Dans [Xu 2003] le robot évolue dans un environnement souterrain. Le robot doit transporter des palettes Europe. L'environnement étant fixe, il est connu à l'avance, mais certains éléments peuvent changer (obstacles mobiles – personnes et obstacles qui sont

occasionnellement repositionnées). Si le robot ne rencontre pas d'obstacle sur sa trajectoire, il suivra, évidemment, la direction principale calculée pour atteindre l'objectif. Au moment où un obstacle est détecté, le robot adopte un comportement proche (selon les auteurs) de celui de l'humain : le robot décide une direction pour le détour, il la prend et ensuite il doit retrouver le chemin initialement planifié. Le robot évite l'obstacle selon la distance des bords visibles. Il prend la direction la plus près du bord (gauche ou droit). Il suit ensuite les bords de l'obstacle (il en est « attaché »). Une fois le chemin original retrouvé, le robot se « détache » de l'obstacle (arrête de le suivre) et revient sur le chemin. Un autre problème est d'éviter les collisions avec des objets se trouvant derrière lui (le robot n'est pas holonome et, quand il tourne, il peut toucher ces objets). L'hypothèse qu'il n'y a pas de glissement des roues est faite. Le centre de rotation du robot est calculé, en fonction de la direction du virage souhaité. Le cadre du robot est dilaté, pour avoir une marge de sécurité. L'angle maximum est ainsi calculé. La Figure 34 montre ceci.

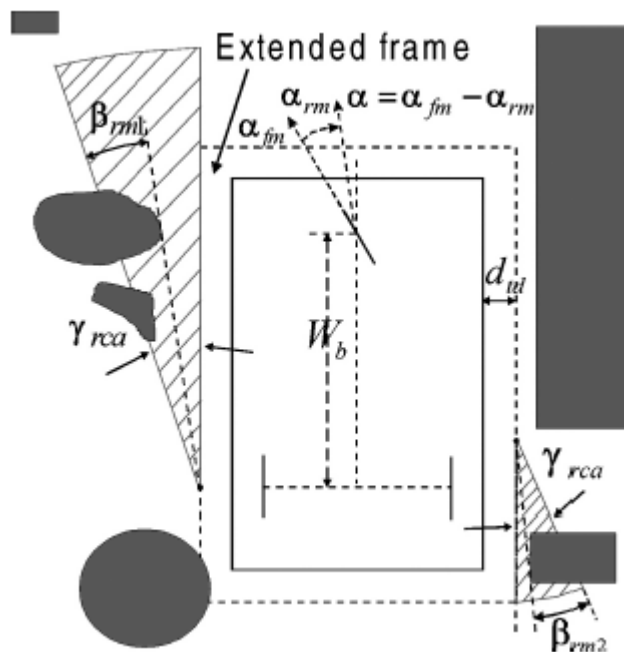


Figure 34 Prise en compte de la forme du robot [Xu 2003]

La vitesse du robot est limitée en fonction de son orientation, selon une parabole. Si la vitesse angulaire est zéro, alors la vitesse linéaire est maximale. Des simulations de cet algorithme sont présentées ensuite. Un exemple est montré dans la Figure 35.

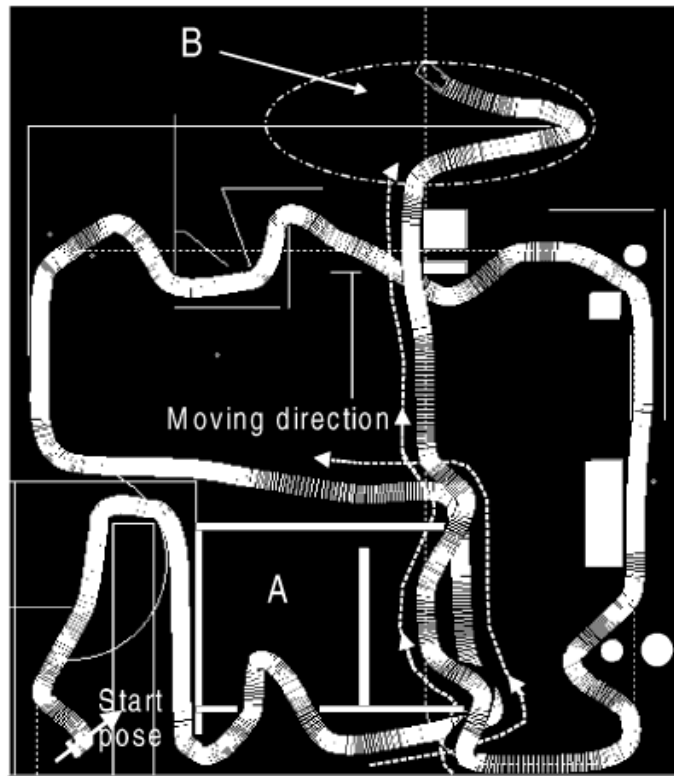


Figure 35 Résultat obtenu en simulation [Xu 2003]

Dans la zone B de l'image on observe une des limites de cette approche : si le mur continuait encore à droite, il y aurait une possibilité que le robot choisisse d'aller à gauche, ce qui le mettrait dans l'impossibilité d'atteindre son but.

Dans [Elfes 1996] l'architecture suivante est utilisée par l'auteur (Figure 36).

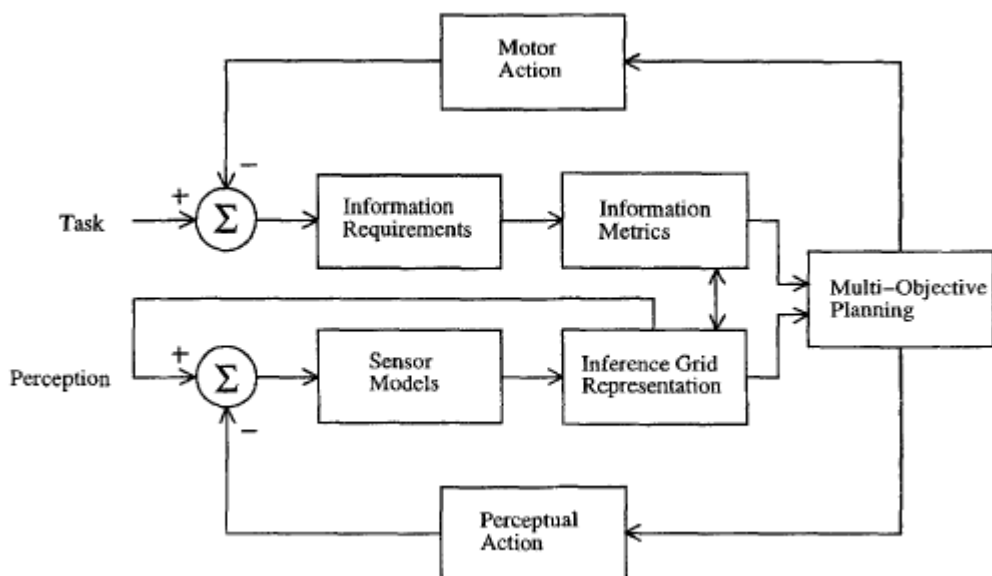


Figure 36 Architecture de contrôle [Elfes 1996]

Le plan de l'environnement est représenté comme une grille d'occupation. Le problème de la combinaison de deux grilles d'occupation est analysé. Un estimateur bayésien est développé

pour ceci. Le problème de la planification de la trajectoire est exprimé comme une fonction à minimiser. Cette fonction prend en compte la longueur de la trajectoire et la probabilité d'occupation des cases qui seront parcourues.

Dans [Wang 2008] le système de contrôle du robot a été spécialement conçu pour explorer les environnements qui présentent beaucoup d'impasses. Pour mémoriser son environnement, le robot utilise une grille d'occupation augmentée, appelée plan de type grille, avec mémoire. Une case a deux propriétés : la première est la probabilité qu'elle soit occupée et la deuxième est le nombre de fois que le robot y est passé. En ce qui concerne cette deuxième propriété, chaque fois que le robot traverse une case (ce qui est déterminé à partir du plan), la valeur de cette propriété est incrémentée. Le comportement global du robot, dans sa recherche d'un chemin vers son objectif, est réparti sur trois niveaux :

- au plus haut niveau on retrouve la recherche de l'objectif
- au niveau intermédiaire on a la recherche de la trajectoire, elle doit diriger le robot vers des régions de risque minimum, afin d'éviter que le robot tombe dans des impasses
- au plus bas niveau on a l'évitement d'obstacles

Le comportement final du robot est donné par une combinaison des résultats de ces trois comportements, en utilisant des poids. Ces poids sont mis à jour en continu, en fonction de la distance qui sépare le robot des obstacles. Des règles sont ensuite appliquées de façon cyclique pour déterminer la direction et la vitesse du robot. La vitesse du robot peut avoir trois valeurs : nulle, faible et forte. Le résultat est que le robot a une stratégie de risque minimum : il essaye d'éviter d'emprunter les chemins déjà explorés jusqu'à ce moment-là dans la recherche de son objectif (chemins qui n'ont pas aboutis) en faveur des nouveaux chemins, pas encore explorés, tout en évitant les obstacles.

Des résultats sont montrés d'abord en simulation. Une comparaison avec d'autres méthodes est présentée. L'environnement choisi est un environnement concave (en forme de U). L'algorithme se montre, pour ce type d'environnement, assez performant.

Des résultats avec un robot réel sont présentés à la fin du papier. L'environnement présente aussi des obstacles mobiles. Sur la Figure 37, le robot part du point S et doit arriver au point T. Une personne bloque une sortie (point A) sur les deux que le parcours possède. Le robot essaie cette piste d'abord et ensuite s'éloigne, à la recherche d'une autre. La personne, en entre temps, se déplace pour bloquer l'autre (point B). Le robot retourne, pour trouver enfin, la première sortie, cette fois traversable, pour atteindre son objectif (point T).

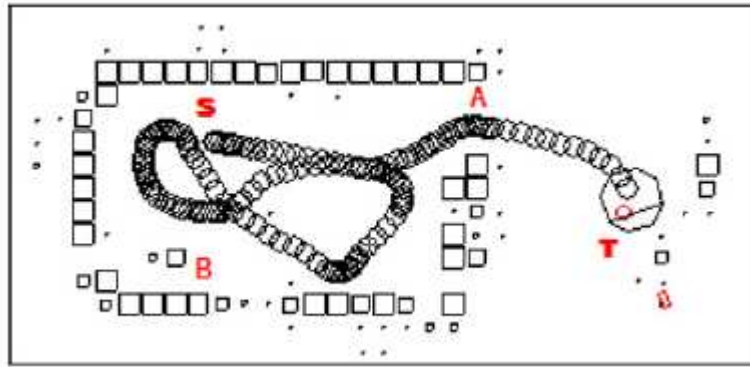


Figure 37 Trajet effectué avec le robot réel [Wang 2008]

La méthode de navigation présentée est capable de trouver un chemin dans des environnements divers et difficiles, comme un mur long, des obstacles en forme de U et des environnements dynamiques. Un défaut de l'implémentation pratique est que la correction de l'odométrie n'est pas effectuée, ce souci n'étant pas présent en simulation.

L'environnement du robot utilisé dans [Thrun 1999] est un musée (le Smithsonian's National Museum of American History). À cause de la grande surface du musée et du fait qu'il peut y avoir beaucoup de visiteurs dedans, ce qui rend la localisation du robot difficile, deux types de plans sont utilisés par le robot : un basé sur les grilles d'occupation et un autre, basé sur les textures du plafond du musée. La génération de deux types de plans est effectuée en utilisant l'algorithme d'espérance-maximisation (EM). Dans l'étape E on estime les positions passées du robot, supposant que le plan de l'environnement est connu. Étant donné qu'au début on n'a pas du tout observé le plan, les premières étapes utilisent l'odométrie. Ensuite, dans l'étape M, on calcule le plan le plus probable en utilisant le déplacement du robot. Même si le robot a un plan de l'environnement, l'algorithme de navigation est modifié pour que le robot reste près des murs, afin que les données laser contribuent aussi à la localisation.

Dans [Ersson 2001] les auteurs se concentrent sur la planification de la trajectoire et considèrent que le robot connaît sa position et son orientation. D'autres hypothèses utilisées par les auteurs sont que la distance de mesure des capteurs est courte (en comparaison avec la surface des régions d'intérêt), et que les capteurs sont positionnés de façon radiale autour du robot. La planification de la trajectoire est faite en utilisant l'algorithme du simplexe. Le coût de passage d'un nœud correspondant à une case libre vers un nœud correspondant à une case occupée est considéré comme étant infini. Deux soucis avec cette méthode de planification sont que la solution obtenue est optimale uniquement pour un réseau discret, pas pour un environnement réel, et que, dans certains cas, un mauvais choix du réseau peut induire une infaisabilité du problème de planification.

Pour le suivi de la trajectoire, les nœuds-cibles sont dilatés et le robot doit suivre un point, plutôt que d'aller d'un nœud à un autre. Le rôle du point-cible est de lisser la trajectoire suivie par le robot.

Dans [Krishnamurthy 2007] les auteurs n'emploient pas du tout un plan global. En dépit de ça, ils supposent que le robot connaît sa position, son orientation et les coordonnées de la cible. L'algorithme se base uniquement sur une approche locale, utilisant la mesure la plus récente

de l'environnement. L'idée est de maximiser une fonction qui a trois composantes : une qui pénalise tout mouvement qui ne se fait pas dans la direction de la cible, une autre qui pénalise le déplacement vers les obstacles et une dernière qui pénalise la marche arrière. La sortie des impasses est faite en utilisant le recuit simulé. La convergence de cet algorithme est montrée par les auteurs. Des résultats en simulation sont montrés aussi. La Figure 38 décrit une situation avec une impasse d'où le robot arrive à se sortir.

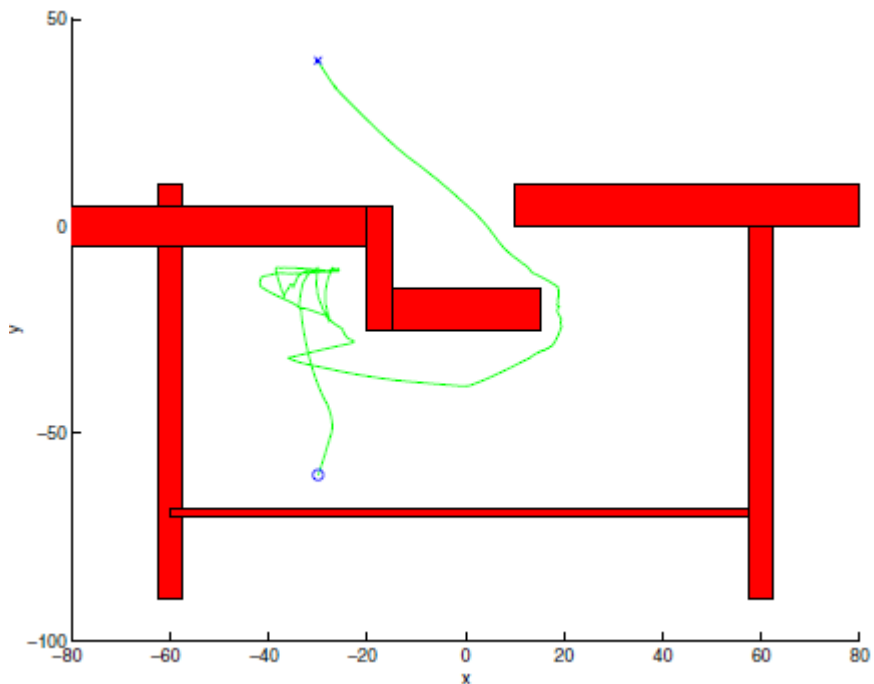


Figure 38 Le robot arrive à sortir de l'impasse [Krishnamurthy 2007]

Quand le robot détecte l'impasse, on voit clairement la navigation aléatoire étant effectuée –la zone située autour de (-30, -10). Cet algorithme est naturellement extensible aux environnements en trois dimensions. Les auteurs montrent aussi des résultats en simulation avec un monde 3D. A cause du fait que le robot n'a pas une connaissance globale de son environnement, il peut se retrouver bloqué dans des grandes impasses. Même si les auteurs présentent la convergence de l'algorithme, elle se fait dans un temps infini, fait sur lequel on ne peut pas se baser dans des contextes réels.

Dans [Leishman 2010] le robot est un fauteuil roulant. L'utilisateur (une personne handicapé) se trouve dedans. Pour le piloter, un joystick est utilisé. Le robot a trois modes de pilotage : un mode manuel (l'opérateur contrôle directement ses vitesses linéaire et angulaire), un mode où l'opérateur lui indique un mur à suivre et le robot le fait de façon automatique et un mode spécialement conçu pour le passage de portes. Pour ce dernier mode de commande, les passages étroits (les portes) sont identifiés à partir d'un scan laser. Une mise en correspondance avec l'image perçue par le robot est faite et la partie traversable de la porte est mise en évidence (par l'intermédiaire d'un rectangle) sur l'image affichée sur l'interface de pilotage du robot. Le choix du mode de commande est fait par l'opérateur. Le robot n'utilise pas un plan de l'environnement pour le suivi des murs et le passage des portes. La planification et l'éventuel évitement d'obstacle sont faits à partir du dernier scan laser. Un

problème remarqué par les auteurs pendant des évaluations avec le système a été lié à la sélection d'une porte qui se trouve trop sur le côté du robot. À cause de sa forme particulière, il a besoin de bien se positionner devant la porte pour la franchir sans collision. Ainsi, il est possible de trouver des situations où le robot devrait s'éloigner de la porte, pour bien se positionner, mais, en faisant ceci, elle disparaît de son scan laser et il faudra qu'elle soit sélectionnée à nouveau par l'opérateur.

Dans [Stentz 1995] le robot est une voiture tout-terrain. Elle possède une centrale inertielle et un odomètre très performants. Ces deux capteurs combinés donnent une erreur de 1% maximum sur la distance parcourue et de quelques degrés sur l'orientation, par heure. Il y a également un GPS et un laser volumétrique embarqués. L'architecture du système de navigation est présentée dans la Figure 39.

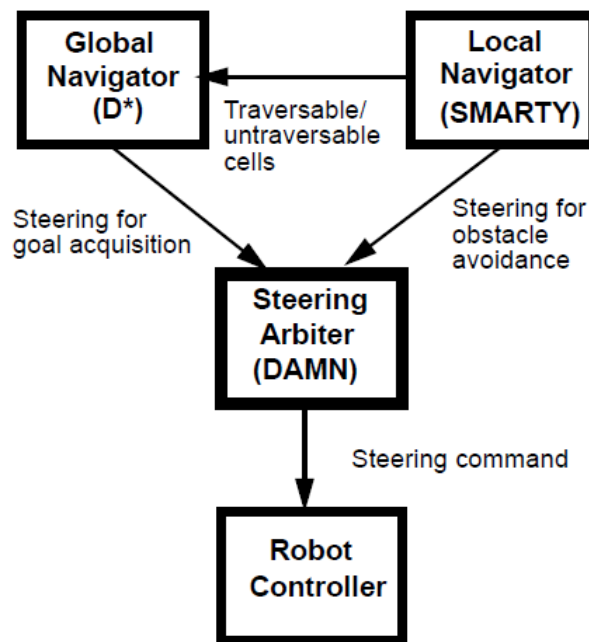


Figure 39 Architecture du système de navigation [Stentz 1995]

Le navigateur global garde un plan grossier (d'une résolution plus faible que le plan utilisé par le navigateur local) dans sa mémoire. Chaque case est marquée comme étant traversable (pas d'obstacle), coûteuse à traverser (proche d'un obstacle) ou pas traversable (au moins un obstacle présent). Au début, toutes les cases sont initialisées comme étant traversables. Le navigateur global prévoit au début une trajectoire vers la cible et envoie des recommandations de direction à l'arbitre de direction pour qu'il déplace le robot vers le but. Pendant que le robot se déplace, le navigateur local balaie le terrain devant le robot pour détecter des obstacles. Le télémètre laser est utilisé pour mesurer la forme tridimensionnelle du terrain, et le navigateur local analyse les cartes pour trouver les portions inclinées (pentes) et les discontinuités qui correspondent probablement aux obstacles. Le navigateur local envoie des recommandations de direction à l'arbitre pour qu'il déplace le robot autour de ces obstacles. En plus, le navigateur local renvoie les cellules intraversables et traversables détectées au navigateur global pour traitement. Le navigateur global compare ces cellules avec sa carte, et, si une anomalie existe (c.-à-d., une cellule traversable est maintenant intraversable ou vice versa), il

calcule une nouvelle trajectoire. L'avantage principal du navigateur global est qu'il peut efficacement prévoir les chemins globaux optimaux et peut produire un nouveau chemin pour chaque série de cellules dans une fraction de seconde (entre 100 et 250 ms). Le navigateur global met à jour sa carte et renvoie de nouvelles recommandations de direction à l'arbitre de direction. Les recommandations de direction sont des arcs de courbure constante, choisis parmi un ensemble donné. Le navigateur global donne un poids important aux directions qui rapprochent le robot de son objectif et le navigateur local donne un poids important aux directions qui évitent les obstacles. Le navigateur global utilise l'algorithme D* pour déterminer le chemin optimal. Des recommandations de trajectoire sont envoyées toutes les 500 ms par le navigateur global.

L'organisation du navigateur local est présentée dans la Figure 40.

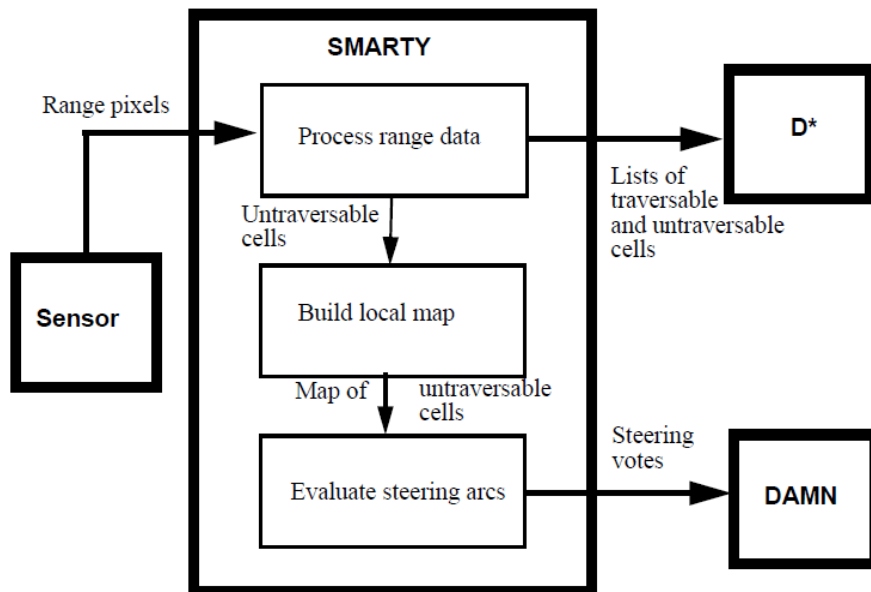


Figure 40 Organisation du navigateur local [Stentz 1995]

Le scan 3D est converti dans une grille d'occupation. En moyenne, cinq points d'un scan correspondent à une cellule. Si une cellule du plan local n'est pas traversable, elle contient aussi les coordonnées 3D du point détecté comme étant intraversable par le robot. Pour le plan local, les dimensions des cellules sont de 40x40 cm.

Le module DAMN combine les recommandations reçues des deux navigateurs pour produire des arcs de direction. Les arcs ont des valeurs comprises entre -1 (à ignorer) et 1 (très bon). Pour les expériences, les poids des recommandations des navigateurs ont été de 0,1 pour les recommandations en provenance du navigateur global et de 0,9 pour les recommandations en provenance du navigateur local. Ce choix favorise clairement l'évitement d'obstacles. Un problème avec l'utilisation des arcs, pour ce type du robot, et que, si le robot arrive dans une impasse étroite, il ne pourra pas faire de marche arrière et, pour les expériences présentées, il sera piloté manuellement.

L'environnement utilisé pendant les expériences est présenté dans la Figure 41, qui correspond à une surface de 800x1000 mètres.

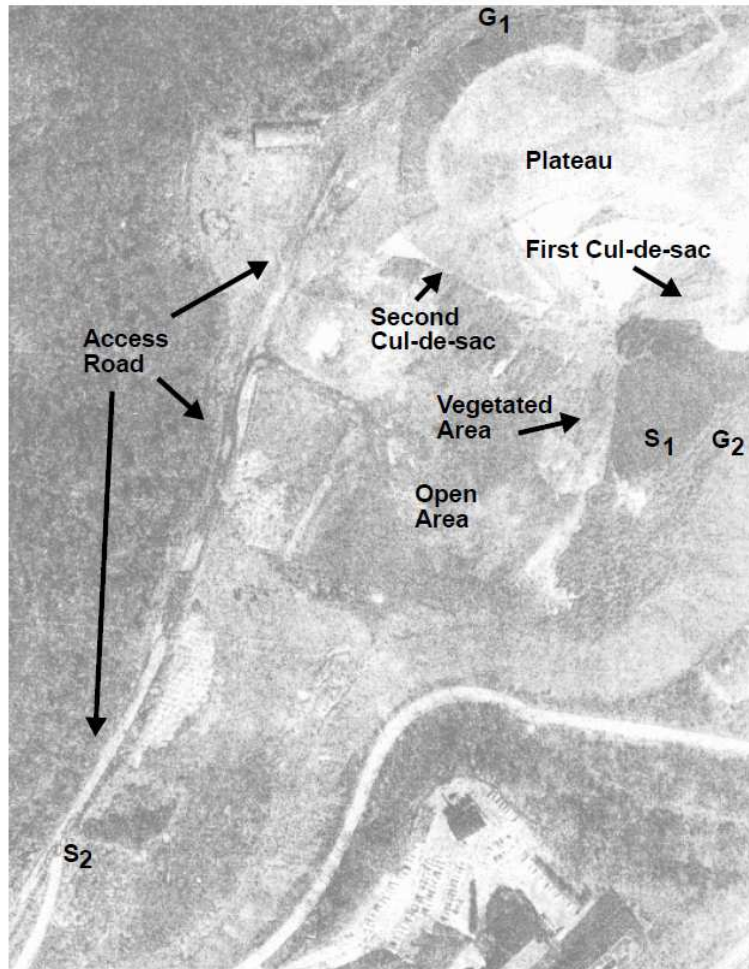


Figure 41 Plan de l'environnement réel [Stentz 1995]

Dans cet environnement, le robot partira de S1, sans aucune connaissance a priori sur l'environnement, et il faudra qu'il arrive en G1 (indiqué comme coordonnées géographiques). Le chemin parcouru par le robot est montré dans la Figure 42.

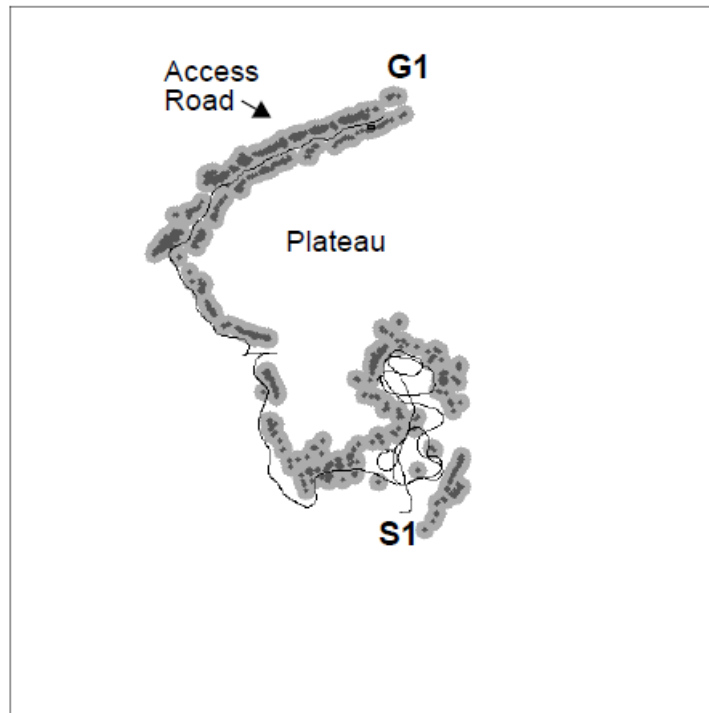


Figure 42 Chemin parcouru par le robot [Stentz 1995]

On peut observer clairement comment le robot rentre dans la première impasse, fait marche arrière et repart à gauche. La deuxième impasse où le robot est entré est trop petite pour lui permettre de faire marche arrière et le robot a dû être piloté manuellement pour sortir de cette situation. La distance totale parcourue par le robot a été, pour cette expérience, de 1410 mètres. Le robot a nécessité l'intervention de l'opérateur humain à six reprises. Trois fois c'était pour le conduire en marche arrière et trois fois pour le diriger autour des flaques d'eau ou de boue, que les capteurs du robot ne pouvaient pas détecter. La classification des cellules a été changée plusieurs fois d'une capture de données à la suivante, principalement à cause du bruit des capteurs, mais aussi parce que, en se rapprochant d'une cellule, l'exactitude des détections augmente.

Un inconvénient avec cette approche est que le contrôle du robot est effectué en direction, sa vitesse étant ignorée. Un autre est qu'il peut uniquement avancer, la marche arrière lui étant impossible.

L'évolution de cette approche est présentée dans [Yahja 2000]. Le même système, avec deux planificateurs, un global et un local, est employé. La différence majeure par rapport au papier précédent est l'utilisation des quadrees bornés. Un quadtree offre un moyen simple et élégant de diminuer la taille occupée en mémoire par le plan d'occupation, et aussi de diminuer le temps de calcul de la meilleure trajectoire. Chaque case du plan peut être divisée en quatre sous-cases, quand on rencontre un obstacle dedans, et ainsi de suite, jusqu'à ce que les sous-cases atteignent la dimension minimale. Dans le cas d'un quadtree borné, toute case qui n'est pas élémentaire est entourée de cases élémentaires. L'avantage de cette approche est évident si on regarde la Figure 43.

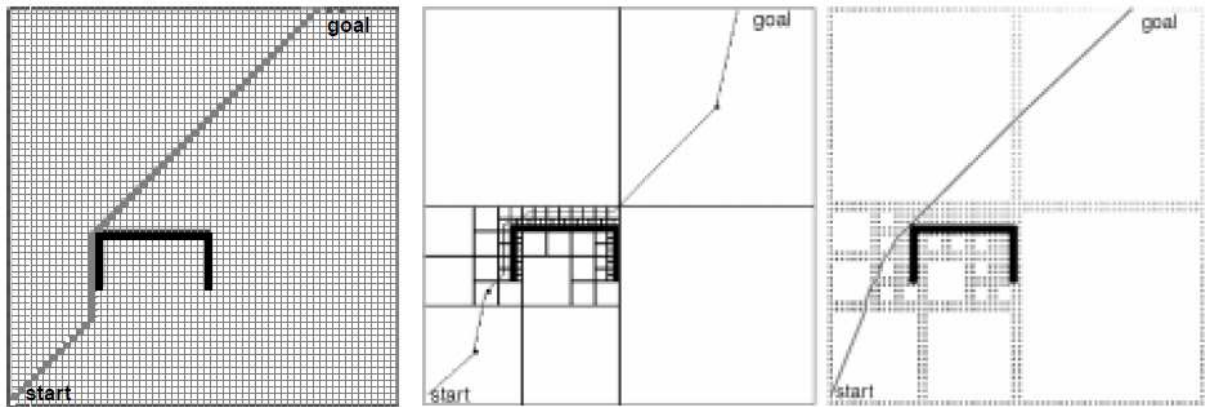


Figure 43 Avantage des quadrees bornés [Yahja 2000]

A gauche on a une trajectoire déterminée à partir d'une grille d'occupation classique. Au milieu, la même trajectoire est déterminée à partir d'un quadtree. A droite, la même trajectoire a été déterminée en utilisant un quadtree borné.

En ce qui concerne le contrôle du robot, la même architecture que celle de la Figure 39 est employée, avec les mêmes avantages et inconvénients. Des résultats obtenus sur le terrain sont présentés. Dans la Figure 44, le robot a parcouru 600 m en 6 minutes et a détecté et évité 80 obstacles.



Figure 44 Parcours effectué par le robot [Yahja 2000]

Le plan correspondant à cette expérience est présenté dans la Figure 45.

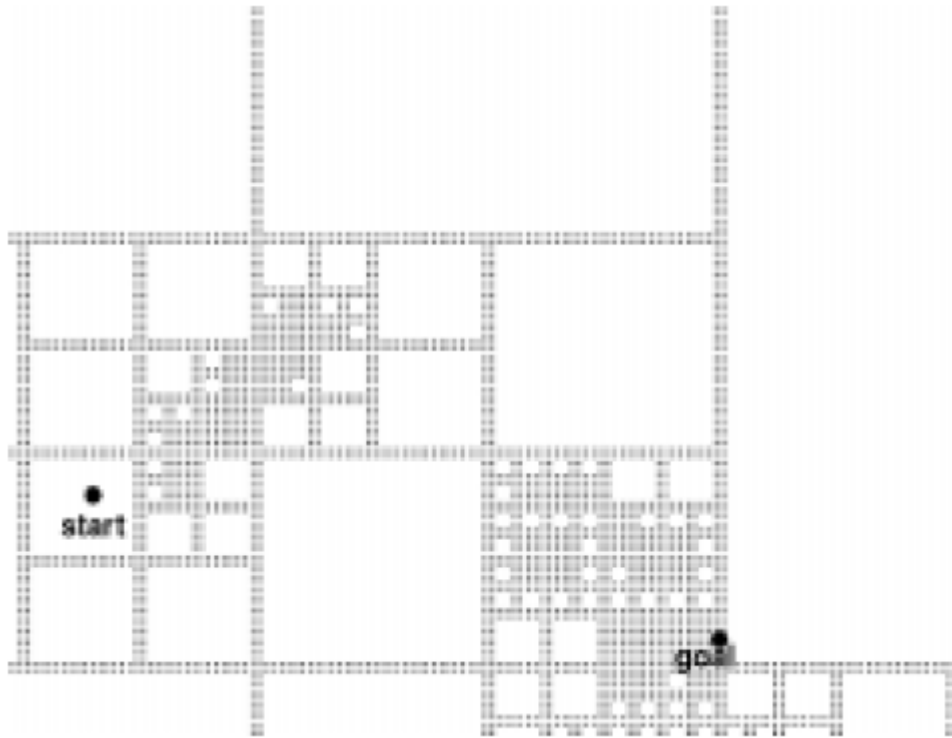


Figure 45 Plan de l'environnement [Yahja 2000]

Comme on peut l'observer, dans la réalité, on a rarement des obstacles repartis de façon uniforme dans l'environnement. Généralement, les obstacles sont groupés, ce qui convient à ce type de représentation (quadtrees bornés).

L'environnement du robot utilisé dans [Larson 2006] est de type marin. Le robot est de type USV (Unmanned Surface Vehicle) – véhicule de surface sans pilote humain à bord. L'évitement d'obstacles pour ce navire est conçu sur deux niveaux : un niveau proche, réactif, et un niveau lointain, délibératif, responsable de la planification de la navigation à moyen-long terme. Le plan du robot est connu dans les grandes lignes, à partir des plans maritimes. Ceci fournit une connaissance assez bonne sur la géographie de l'environnement. A* est utilisé pour assurer la planification de la trajectoire. Quand il y a un danger de collision avec un autre navire, l'évitement proche prédit la position du navire (Figure 46) et applique les règles de priorité maritime: si les deux navires vont droit l'un vers l'autre, ils passeront à bâbord (les navires seront positionnés comme sur la route - volant à gauche) ; si ils se croisent bâbord à bâbord ou tribord à tribord, ils continuent leurs routes ; les navires venant de tribord ont la priorité.

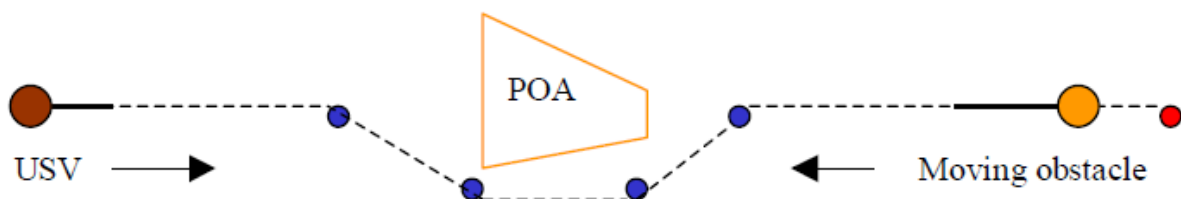


Figure 46 Exemple d'évitement d'obstacles [Larson 2006]

Le contrôle de la trajectoire du robot est fait en choisissant un des arcs présents dans la Figure 47. Chaque arc reçoit un vote, les longs arcs étant favorisés en détriment des arcs plus courts ou qui sont bloqués par des obstacles proches du robot.

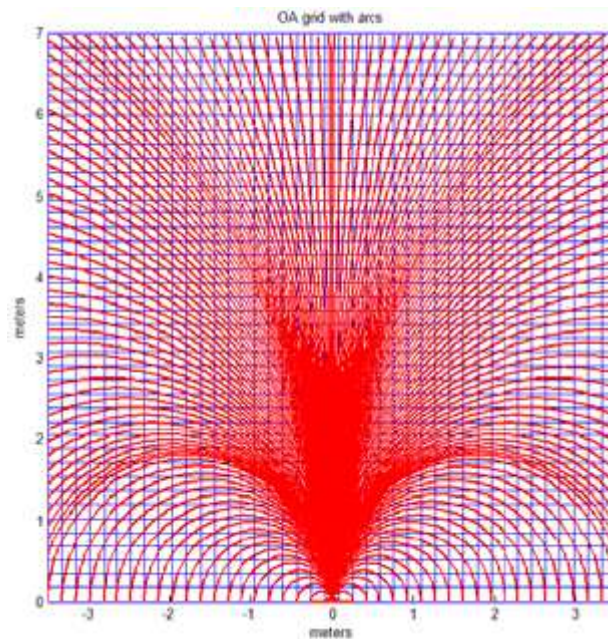


Figure 47 Choix possibles [Larson 2006]

Le robot utilisé dans [Hamner 2008] est de type rover, utilisé dehors. Il est équipé avec deux scanners laser, un GPS et une centrale inertielle. Sa vitesse maximale est de 5 m/s. Deux niveaux de planification sont utilisés dans son architecture : un niveau global et un niveau local. Le planificateur global utilise A* pour déterminer le meilleur chemin. Le planificateur local utilise les champs de forces, mais ne les applique pas sur la position du robot, mais sur la fonction de direction. L'objectif attire le robot et les obstacles le repoussent. La vitesse du robot est ajustée en fonction de sa distance aux obstacles et de leurs positions. Si le robot se retrouve bloqué (Figure 48), alors le planificateur global calcule une nouvelle destination, plus proche du robot, pour sortir de cette situation.

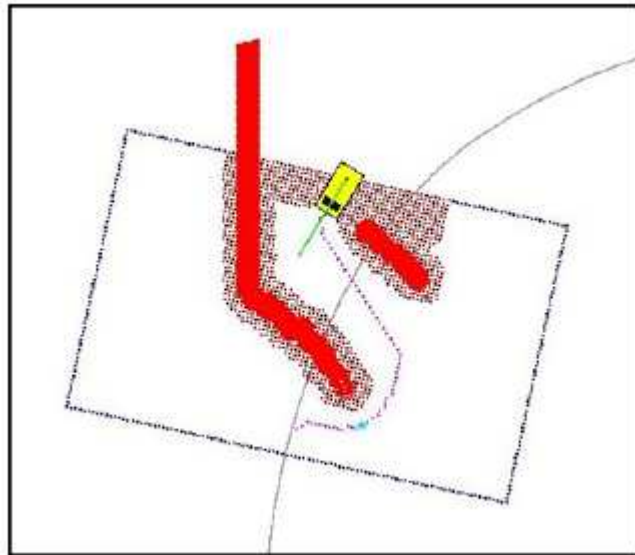


Figure 48 Le robot a besoin d'une aide supérieure [Hamner 2008]

La ligne émanant du robot est le résultat du planificateur local. Le grand arc de cercle présent dans l'image est la trajectoire calculée avant d'avoir d'informations plus détaillées sur l'environnement. La trajectoire pointillée, en forme de S, qui relie le robot avec la trajectoire globale, est la trajectoire que le robot empruntera, en utilisant le point situé derrière les obstacles comme nouvelle destination (attracteur).

Pour la validation, le robot a parcouru plusieurs fois le trajet décrit dans la Figure 49, pour une longueur totale de 22,8 km.

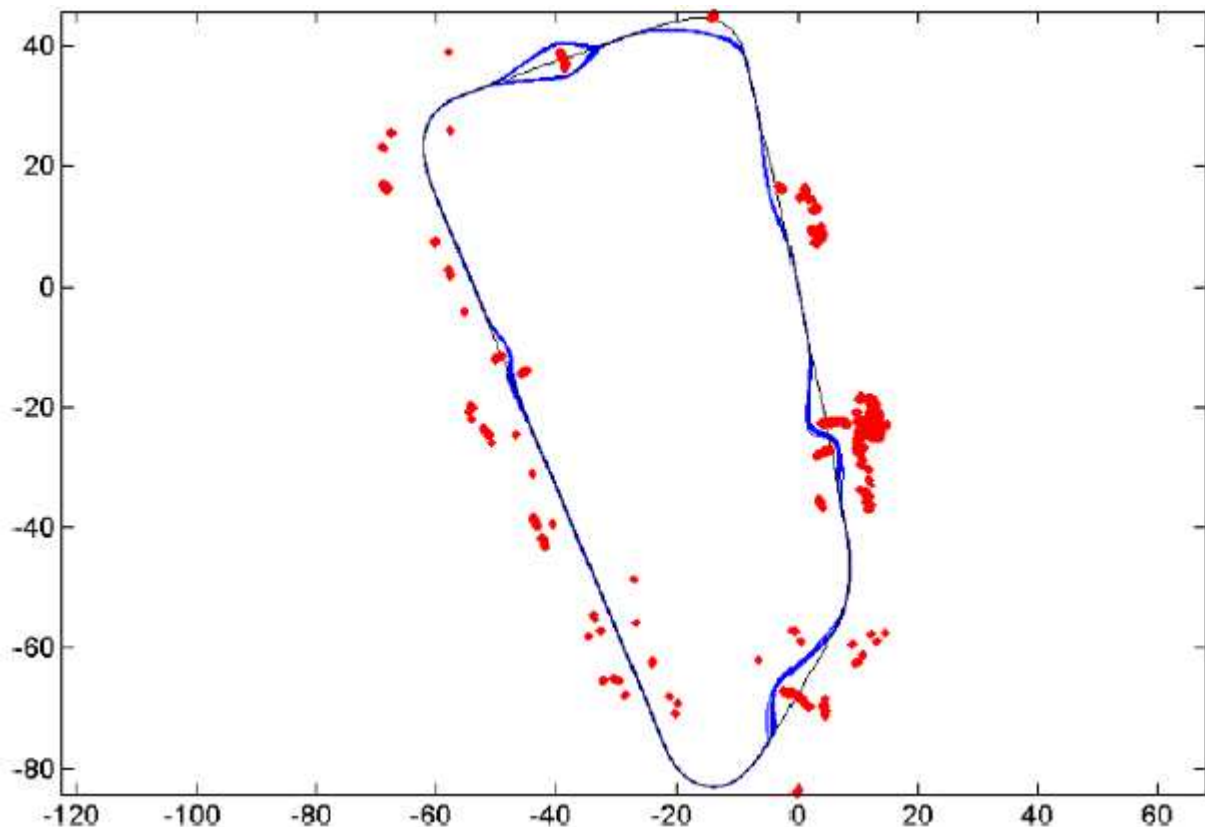


Figure 49 Boucle parcourue par le robot [Hamner 2008]

Une nouvelle méthode de calcul de parcours est introduite dans [Philippsen 2004]. La méthode, appelée E*, est similaire à D*. Elle emploie la propagation d'un front d'onde pour déterminer la trajectoire optimale. L'algorithme ne dépend pas d'une méthode précise. Deux méthodes pour la réaliser sont présentées : l'une est basée sur le principe de Huygens et l'autre sur l'approximation du gradient. Un parcours utilisant cette méthode de planification est montré (en simulation) dans la Figure 50. Les points noirs représentent les endroits où le robot a dû replanifier sa trajectoire, à cause des obstacles.

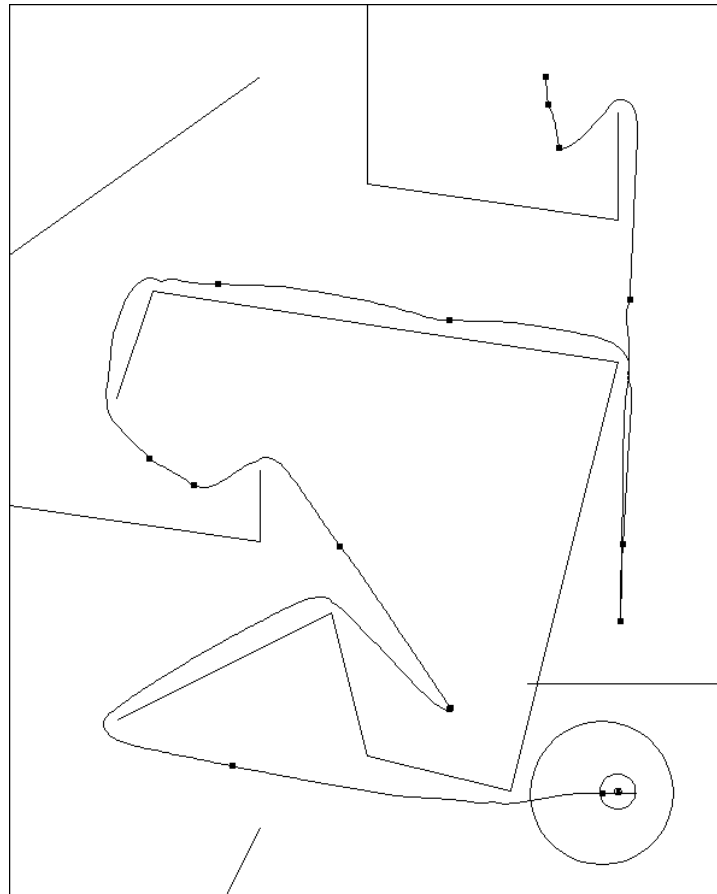


Figure 50 Parcours en simulation [Philippsen 2004]

[Stachniss 2002] montre comment faire l'intégration de la planification de la trajectoire et l'évitement d'obstacle en utilisant un espace multi-dimensionnel (x, y, θ, v, ω – les coordonnées du robot dans un espace 2D, son orientation, sa vitesse linéaire et sa vitesse angulaire). Comme l'espace de recherche est trop grand, les auteurs décrivent un raccourci pour aboutir à la solution. Les quatre étapes de ce raccourci sont les suivantes :

- mise à jour du planificateur
- calcul d'une trajectoire en (x, y) en utilisant le nouveau plan
- en partant de cette trajectoire, détermination de l'espace à explorer (le canal)
- recherche d'une trajectoire en $(x, y, \theta, v, \omega)$ dans l'espace à explorer

La taille du canal (étape trois) a été choisie par les auteurs de façon que ce canal soit exploré en 250 ms maximum. Les dimensions du canal peuvent être modifiées d'une itération à une autre.

La Figure 51 montre le mode de fonctionnement de cet algorithme.

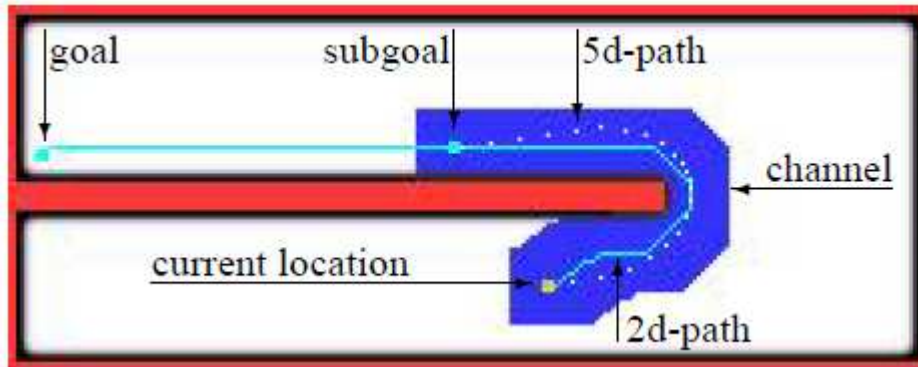


Figure 51 Illustration du tunnel virtuel [Stachniss 2002]

On peut voir une représentation bidimensionnelle du canal autour de la trajectoire (en x, y) du robot (ligne continue) et une projection de la trajectoire planifiée en 5D à l'intérieur du canal (en pointilles). Des résultats avec un robot réel sont également présentés (Figure 52).

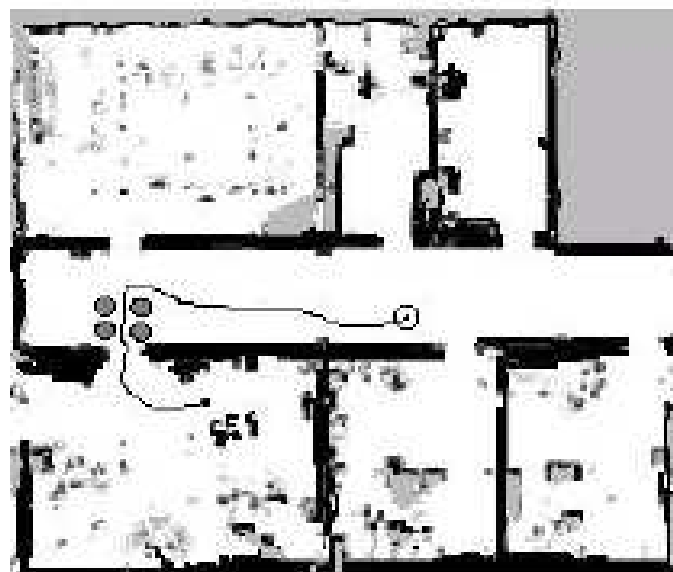


Figure 52 Résultat obtenu avec un robot réel [Stachniss 2002]

Sur cette figure, on peut observer le passage d'une porte par le robot. Devant la porte, les quatre cylindres forment un passage étroit, que le robot arrive à négocier. Le principal avantage de cette méthode est que le robot détermine simultanément une partie de la trajectoire à utiliser et sa vitesse à chaque point de cette trajectoire

3. Implémentation de la commande en environnement inconnu

Le système de commande en environnement inconnu reçoit une position-cible et doit guider le robot à cet endroit-là. Même si on n'a pas de connaissance sur l'environnement du robot, on a des hypothèses sur son monde. Le robot se trouve dans un bâtiment, habitation ou bureau. L'objectif principal de l'opérateur n'est pas d'envoyer le robot dans des différents endroits (faire de la navigation). Son but est d'explorer cet environnement et de rechercher une personne dedans. Si l'opérateur utilisait le mode de commande manuel, son attention serait consacrée au pilotage du robot (quelle trajectoire prendre, quels virages, et la corriger pour éviter les obstacles). Pour ce nouveau mode de commande, l'opérateur peut se permettre de n'intervenir que sporadiquement, pour donner une nouvelle destination au robot ou l'arrêter, si besoin. Ainsi, l'opérateur aura la possibilité de distribuer son attention entre la supervision du robot et l'analyse de la vidéo reçue, pour chercher quelqu'un, ou même de s'occuper d'une activité parallèle (visioconférence avec un tiers, par exemple). Le robot devra déterminer et suivre une trajectoire jusqu'à la destination qui lui a été indiquée par l'opérateur, et éviter les obstacles potentiels.

Pour atteindre l'objectif reçu de la part de l'opérateur, le robot doit être capable de calculer une trajectoire jusque là (on suppose que l'objectif n'est pas toujours situé dans l'espace libre autour du robot). Cette trajectoire implique de la planification. La planification nécessite un plan, qui doit être créé par le robot lui-même. Parce qu'il s'agit d'un mode de commande, et non pas d'exploration autonome, le temps est une contrainte forte (le robot doit répondre instantanément aux consignes de l'opérateur, on ne peut pas se permettre d'accorder au robot un intervalle long de temps pour des calculs variés avant qu'il commence à se déplacer ou qu'il fasse des pauses sur le parcours).

La génération du plan de l'environnement est prise en compte par un algorithme de SLAM, plus spécifiquement celui utilisé par [Steux 2010]. En faisant ceci, on obtient aussi la position du robot dans son environnement, avec une correction sur l'odométrie. L'algorithme de SLAM utilise en entrée uniquement les données laser. Le résultat du SLAM est une grille d'occupation. Dedans, des algorithmes classiques de recherche de chemin optimum (D^*) sont facilement utilisables.

3.1. Architecture de commande

La Figure 53 décrit le début de l'architecture de commande utilisée par notre robot.

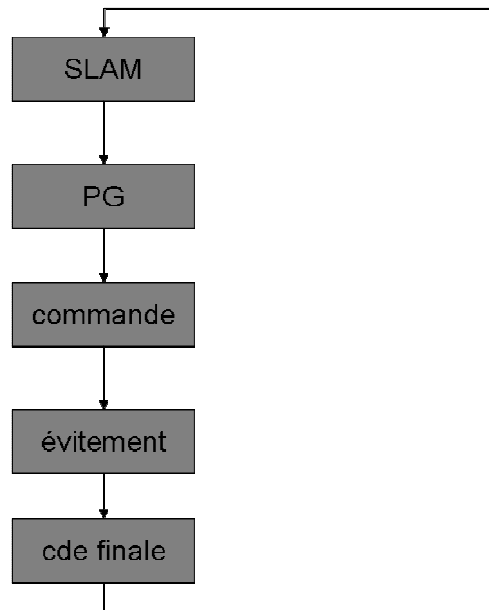


Figure 53 Début de l'architecture de commande

Le système reçoit les coordonnées de la destination (x, y) donnée par l'opérateur humain dans le repère lié au robot comme entrée. Le SLAM se déclenche et envoie la position courante du robot (toujours $(0, 0)$ au début) au planificateur global (PG). Le plan initial ne contiendra qu'un seul scan laser. Dedans, le planificateur global détermine une première trajectoire (les zones inconnues sont traitées comme traversables). Si la destination est à l'intérieur de ce scan laser et qu'elle est directement atteignable (ou on observe tous les obstacles), la première trajectoire déterminée sera la bonne. Le robot suit cette première trajectoire et il découvre plus de son environnement (le plan est actualisé par le SLAM). Si la trajectoire n'est pas faisable (génération de commande impossible à cause des obstacles qui bloquent le chemin), PG générera une nouvelle trajectoire, en utilisant le dernier plan fourni par le SLAM (qui a plus d'informations sur l'environnement que celui qui a été utilisé pour calculer la trajectoire précédente).

Le planificateur global utilise D^* et est exécuté sur la globalité du plan sauvegardé sur le robot. Une commande est générée et envoyée au module d'évitement d'obstacles. Le robot est également équipé de capteurs ultrasoniques (sonars). Les informations qu'ils fournissent sont utilisées pour faire un évitement réactif local. Ceci est nécessaire à cause du fait qu'il est possible que le plan acquis par le SLAM ne soit pas une représentation parfaite du monde réel (détails perdus pendant la discrétisation, êtres/objets en mouvement, etc.). La commande finale est ensuite envoyée aux actionneurs des roues du robot, pour être exécutée.

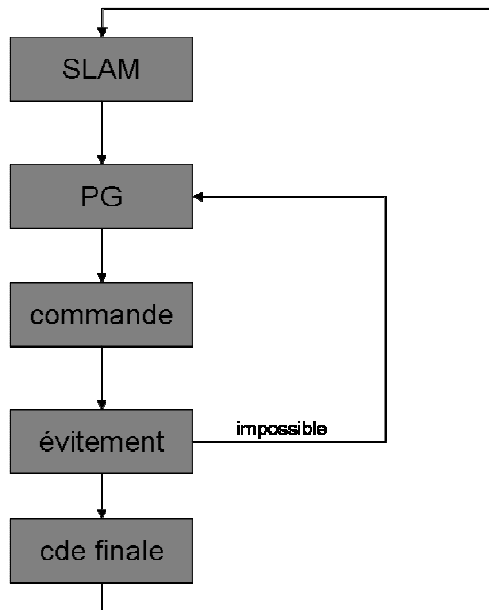


Figure 54 Situation de replanification

Si le robot arrive dans une impasse et que l'évitement d'obstacles ne peut pas générer, à partir de la commande qui lui a été transmise par PG, une commande pour déplacer le robot, PG sera informé et une nouvelle trajectoire globale sera planifiée (Figure 54). Cette trajectoire prendra en compte le problème que l'évitement n'arrivait pas à résoudre et, donc, celui-ci pourra générer une commande pour déplacer le robot (pour le cas d'une impasse, il est possible que la nouvelle consigne qui lui est envoyée est de faire marche arrière).

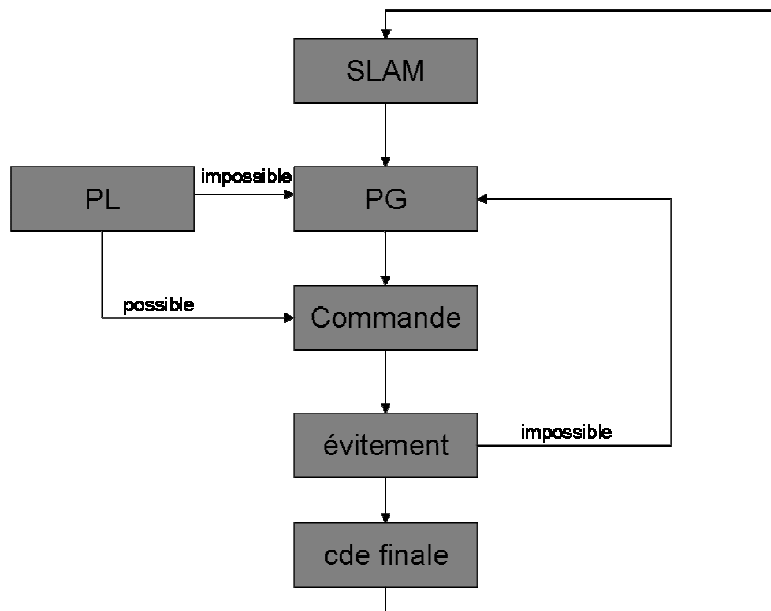


Figure 55 Le rôle du planificateur local

À cause du fait que la planification globale est coûteuse (en temps d'exécution), on a rajouté un planificateur local (PL - Figure 55). Le planificateur local opère dans une fenêtre locale. Celle-ci a une taille égale à la distance maximale de mesure du laser. Ainsi, on est sûr qu'on planifie en disposant des informations les plus récentes sur l'environnement (envoyées par le dernier scan laser). Si le planificateur local n'est pas capable de fournir une trajectoire adéquate (ce qui peut arriver même en début de l'exécution de l'algorithme, si la destination n'est pas dans la fenêtre locale), le planificateur global génère une trajectoire, qui passe par des zones qui n'ont pas encore été explorées. Une fois qu'une trajectoire est trouvée, le robot la suit en employant un module suiveur de trajectoire, pour générer des commandes de mouvement.

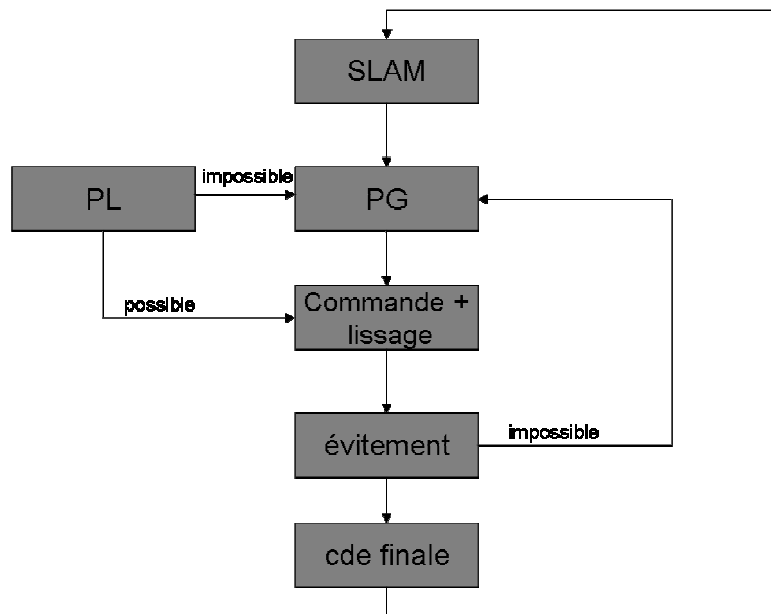


Figure 56 Lissage de la trajectoire

Une autre amélioration a été de modifier la commande, afin de lisser la trajectoire (Figure 56). PL et PG raisonnent sur une grille d'occupation, et la trajectoire générée est une succession de centres des cases. Le plus souvent (sauf pour un mouvement en ligne droite, parallèle avec la grille), la trajectoire est en zigzag. Le lissage vérifie s'il n'est pas possible de trouver une ligne droite, jusqu'au point le plus éloigné qui se trouvent encore dans la fenêtre utilisée par PL. Si un tel raccourci est possible, la commande est modifiée pour le prendre en compte. Un chemin vers le point intermédiaire suivant est calculé avant d'arriver au point intermédiaire courant, pour avoir le temps de tourner dans la nouvelle direction et pour que le déplacement du robot ne soit pas saccadé. Si la destination n'a pas été atteinte, on revient au début, à la détermination du plan et de la position du robot.

Pour l'implémentation pratique (que ce soit en simulation ou sur le robot réel), cette logique de déroulement a été adaptée afin d'améliorer les performances. Certaines parties se déroulent non pas de façon séquentielle, mais en parallèle, pour mieux profiter de la puissance de calcul offerte par les processeurs multi-cœur et pour satisfaire les contraintes de temps réel nécessaires pour un mode de commande. Par exemple, le SLAM et le planificateur local se déroulent en parallèle, avec une fréquence plus réduite pour le SLAM, qui met plus de temps à finir une boucle que le planificateur. Pour éviter l'utilisation des données incomplètes (une

moitié du plan actualisée et une moitié du plan provenant de l'ancienne position du robot n'est pas exploitable), des structures de données tampons sont utilisées. En ce qui concerne l'odométrie du robot, entre deux itérations de l'algorithme de SLAM, la position fournie par le SLAM est utilisée comme base de départ, et les mises à jour fournies par les codeurs du robot sont faites dessus.

L'algorithme de SLAM utilisé a tiré son inspiration du celui décrit dans [Steux 2010]. Les contextes se ressemblent beaucoup : la nécessité d'avoir l'algorithme embarqué sur le robot et des contraintes de temps réel (le robot doit se déplacer en permanence sur la trajectoire courante qui le relie avec son objectif). Une différence est que, pour accélérer les calculs, nous employons les quadtrees (Figure 43) comme structures de stockage des plans obtenus.

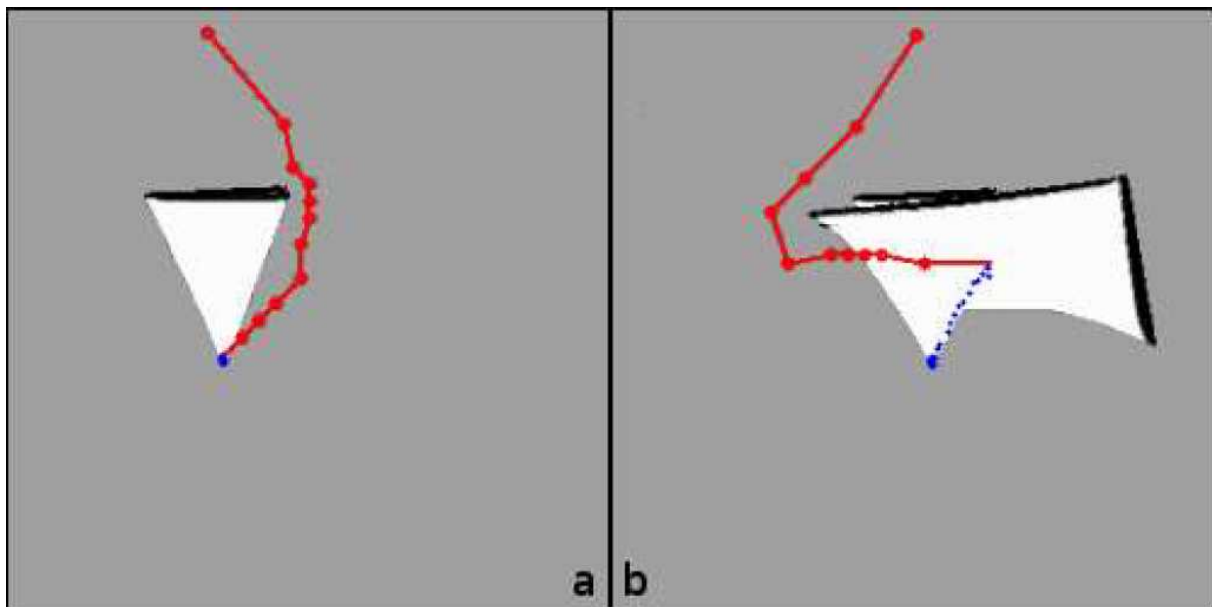


Figure 57 Replanification globale

Dans la Figure 57, à gauche, on peut observer la trajectoire initiale, proposée par le planificateur global, que le planificateur local essaiera de respecter. Le robot détecte bientôt l'impasse qui se trouve à droite. Parce que la distance entre cette position du robot et sa destination est trop grande pour que le planificateur local puisse proposer une alternative, le planificateur global est alors invoqué pour déterminer une trajectoire vers l'objectif (sur la figure de droite). Dans les deux figures, la trajectoire globale est représentée par des gros points reliés par des segments, avec les centres des cases marqués par ces gros points. La trajectoire effectuée par le robot, guidé par le planificateur local, est en pointillés.

Un avantage d'utiliser le planificateur local est que la trajectoire calculée par le planificateur local peut éviter des obstacles sur lesquels le planificateur global n'avait pas de connaissances au moment de la planification, tout en laissant la trajectoire globale inchangée. La Figure 58 illustre cette situation. La trajectoire globale est représentée par des gros points reliés par des segments, avec les centres des cases marqués par ces gros points. La trajectoire effectuée par le robot, guidé par le planificateur local, est en pointillés.

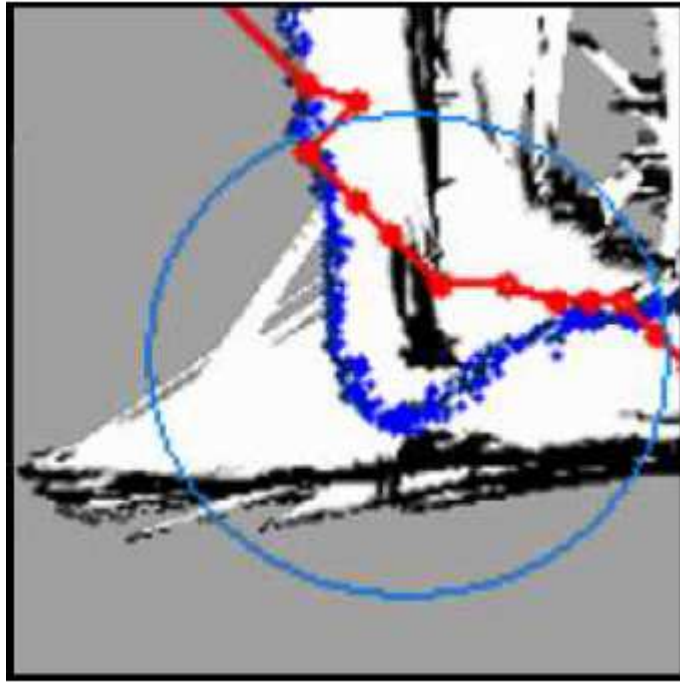


Figure 58 Replanification locale

Pour une trajectoire générale qui n'a pas été mise à jour depuis longtemps, on a de fortes chances que les cases soient assez larges, à cause d'un manque d'information sur ces zones de l'environnement et de l'utilisation des quadrees. La trajectoire générale fera probablement des zigzags, que le planificateur local arrivera, dans une certaine limite, à lisser (Figure 59). La trajectoire globale est représentée par des gros points reliés par des segments, avec les centres des cases marqués par ces gros points. La trajectoire effectuée par le robot, guidé par le planificateur local, est en pointilles. En fonction de la taille de la fenêtre locale, le planificateur local arrive à lisser, sur une certaine longueur, la trajectoire proposée par le planificateur global.

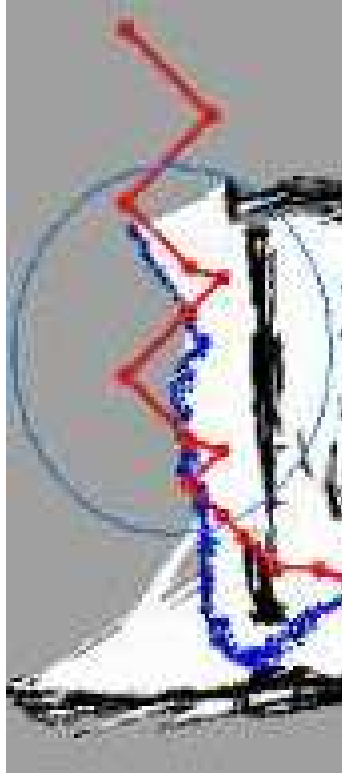


Figure 59 Lissage de la trajectoire

Le suiveur de trajectoire doit générer des commandes qui permettront de suivre la trajectoire générée par le planificateur local. L'évitement des obstacles est un évitement bas-niveau. Il est principalement chargé d'éviter les obstacles détectés avec les sonars, qui sont invisibles pour le laser. Il est basé sur un principe réactif, dès qu'il constate que le robot touchera un obstacle, il appliquera une vitesse angulaire qui fera que le robot va s'éloigner du danger. Une fois les vitesses linéaires et angulaires déterminées, elles seront transformées dans des commandes pour les actionneurs des roues par l'intermédiaire du module d'avancement.

3.2. Résultats

Ce mode de commande a été implémenté d'abord en simulation, ensuite sur le robot réel. Pour l'acquisition du plan, on a utilisé le laser.

Parce que en simulation l'odométrie du robot et son capteur laser sont parfaits, du bruit gaussien a été rajouté dessus. La Figure 60 montre un résultat typique. La trajectoire globale est représentée par des gros points reliés par des segments, avec les centres des cases marqués par ces gros points.. La trajectoire effectuée par le robot, guidé par le planificateur local, est en pointillés.

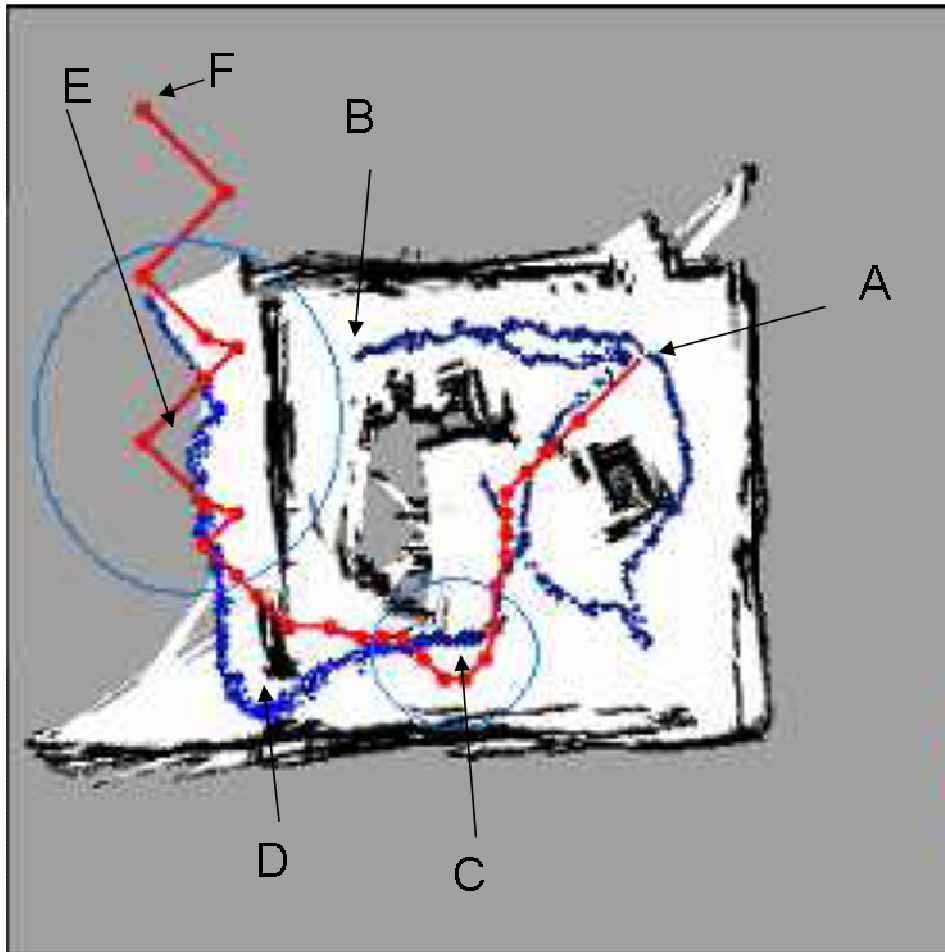


Figure 60 Exemple de parcours (en simulation)

Pour trouver la sortie, le robot est obligé, à cause de l'approche optimiste adoptée, d'explorer tout l'environnement avant de trouver une sortie. Le robot part du point A et doit se rendre dans le point F. Sans avoir de connaissances a priori sur l'environnement (comme c'est le cas pour la Figure 57), il se dirige d'abord directement vers son objectif (le point F). En arrivant au point B, il se rend compte qu'il ne peut pas passer et il cherche un nouveau chemin. La solution trouvée le ramène à essayer de sortir par la partie haute, droite (située autour du point A). Il descend ensuite, en longeant le mur droit, essaie d'aller vers l'objectif par le centre (impossible), et continue sa descente, pour trouver la sortie située en bas, à gauche (zone D). Dans les zones C et D on observe l'effet du planificateur local (la trajectoire globale n'a pas été régénérée). Dans la zone E on observe le comportement de lissage de la trajectoire.

Pour toutes les situations testées, le robot a pu atteindre l'objectif indiqué par l'opérateur, si un chemin existait. Sinon, il s'arrêterait et l'opérateur était informé par un message.

Parce que le robot utilisé en simulation a été modélisé en utilisant les mêmes paramètres que le robot réel, le code a pu être transféré tel quel de l'emploi en simulation à l'emploi en réel. La Figure 61 montre un essai avec ce mode de commande utilisé sur le robot réel. La trajectoire globale est représentée par des gros points reliés par des segments, avec les centres des cases marqués par ces gros points. La trajectoire effectuée par le robot, guidé par le planificateur local, est en pointilles. Dedans, on peut observer le comportement de replanification locale afin de passer le coin touché par la trajectoire globale et de lissage pour la fin du parcours.

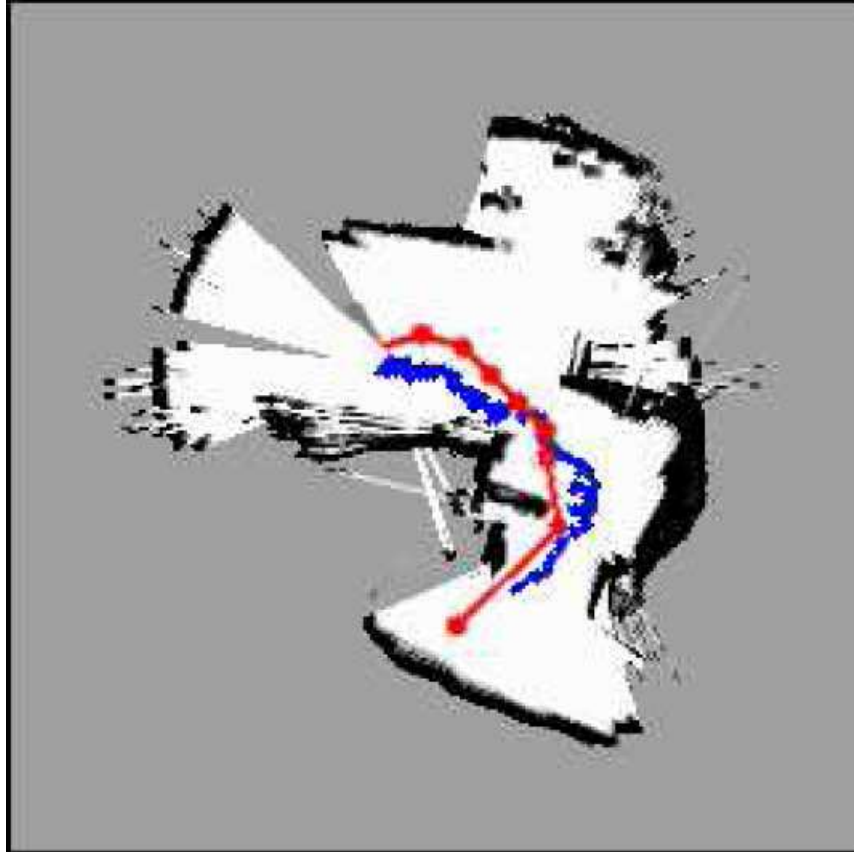


Figure 61 Exemple de trajectoire (avec le robot réel)

Dans ce cas, les capteurs sont à la base de quelques problèmes, un exemple étant la détection des tables : de loin, le laser n'observe pas bien les pieds et de près les sonars passent par en dessous du plateau.

3.3. Conclusions

Ce nouveau mode de commande semi-autonome a donné de très bons résultats en simulation et des résultats prometteurs avec un robot réel. Les capteurs du robot ont un rôle important dans le SLAM. Si on compare la Figure 60 (simulation) avec la Figure 61 (réel), on constate que les murs ont tendance à ne plus être perpendiculaires, même si l'algorithme de SLAM n'a pas été modifié. Le but de départ était de fournir un nouveau mode de commande à l'opérateur humain. On voulait que ce mode de commande utilise plus les capacités d'autonomie du robot que le mode du robot, chose qu'on a accompli. La description de ce mode de commande et les premiers résultats qu'on a obtenu ont été présentés dans [Devaux 2011].

Un problème potentiel est le fait que le robot peut, théoriquement, sortir du plan utilisé pour la planification de sa trajectoire. Les particularités présentées par notre situation font que ceci est peu probable. D'abord, le robot se trouve dans un bâtiment, fait qui nous permet de définir une taille du plan suffisante pour la majorité des immeubles. Ensuite, le robot est téléopéré, la construction de l'interface homme-machine (décrite de façon détaillée dans le chapitre suivant) ne permettant pas de définir des destinations extrêmement éloignées du robot.

3.4. Perspectives

Pour l'utilisation avec un robot réel, on s'est rendu compte du fait que la qualité des plans obtenus en simulation et avec le robot réel était différente (en employant le même algorithme de SLAM). On suppose que ceci est dû aux capteurs du robot réel. Si les capteurs ne fournissent pas des bons résultats, le SLAM n'est plus capable de fournir un bon plan et une localisation fiable du robot. On espère que l'utilisation du capteur Kinect du Microsoft et la fusion avec le laser et les sonars améliorera les performances du robot dans le monde réel. Pour généraliser ce mode de commande (à l'extérieur, par exemple), on envisage également d'utiliser une taille variable (dans le sens croissant) pour le plan de l'environnement.

Concernant le suivi de trajectoire, dans son implémentation actuelle on ne cherche pas à minimiser le temps de parcours du robot. Le lissage de la trajectoire peut réaliser ceci, mais c'est un comportement émergent. On peut envisager l'amélioration du temps de parcours en utilisant des informations sur l'environnement pour la génération des vitesses ou relaxer les conditions de suivi de trajectoire, si l'obtention d'un meilleur temps de parcours est envisageable.

Interface homme-machine et changement de mode de commande

Dans le premier chapitre de cette thèse on s'est concentré sur le mode de pilotage du robot manuel et on a introduit une aide lui permettant d'être utilisé en présence des délais entre le site de l'opérateur et celui du robot. Ce mode est un mode de bas niveau, basé sur les capacités de pilotage de l'opérateur. Dans le chapitre suivant on a introduit un mode semi-automatique, qui peut être utilisé avec un délai encore plus grand. Ce mode de commande se base plus sur les capacités du robot de détecter correctement les obstacles et de déterminer une trajectoire satisfaisante par rapport aux exigences de l'opérateur (l'objectif qu'il a indiqué). Dans ce chapitre, l'interface qui permet l'utilisation de tous les modes de commande disponibles est présentée au début. Avec la même interface, il est possible de piloter le robot dans un mode de bas-niveau, mais aussi dans un mode semi-automatique. Ainsi, l'opérateur n'a pas besoin de lancer une interface différente, s'il souhaite utiliser un autre mode de commande. A la suite, la question du changement de mode de commande est analysée. Même s'il a à sa disposition une interface qui lui permet de changer de mode, on a voulu analyser la possibilité de déterminer des conditions pour la réalisation de ce changement, pour qu'il soit réalisé en douceur pour l'opérateur, ainsi que pour le robot.

1. Interface Homme-Machine

L'équipe HANDS du laboratoire IBISC travaille depuis de nombreuses années sur la téléopération d'un robot mobile à roue par un opérateur distant. Pour mieux comprendre comment on envisage que l'opérateur utilise les différentes modes de commandes qui lui sont mis à disposition, cette section présente l'historique des interfaces développées par l'équipe HANDS et propose une évolution.

1.1. État de l'art initial

La téléopération se caractérise par une action à distance d'un opérateur sur un environnement par l'intermédiaire d'un outil, ici un robot. Ce dernier renvoie à l'opérateur des informations sur son état et celui de l'environnement grâce à différents types de capteurs. La Figure 62 illustre cette configuration de travail. L'objectif de la Coopération Homme-Machine est alors de déterminer la meilleure stratégie de partage des tâches entre l'opérateur et le robot.

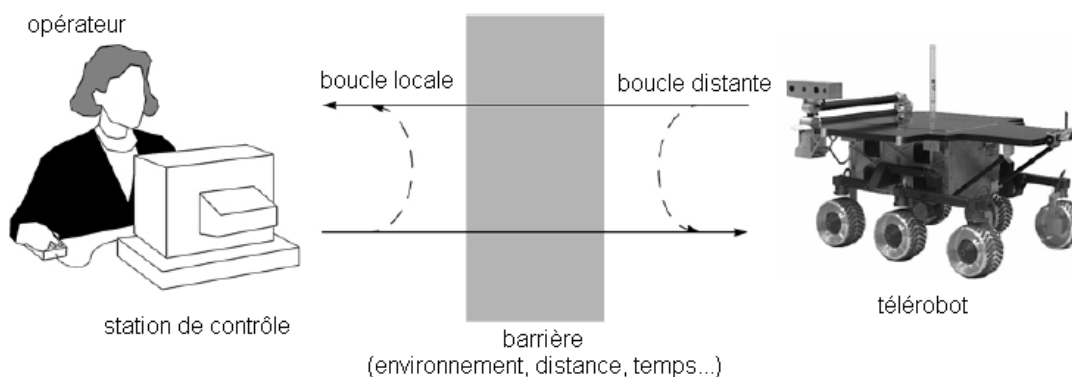


Figure 62 Situation de téléopération (adapté de [Fong 2001])

Le robot induit des contraintes techniques de deux ordres : les limites des capteurs et des actionneurs en incluant les capacités d'autonomies du robot et les délais de transmission entre les deux entités (question abordée dans le premier chapitre de cette thèse). La présence de

l'homme induit aussi des contraintes. En situation normale, l'être humain possède une grande richesse et une grande diversité d'informations sensorielles (visuelles, auditives, tactiles...). En situation de téléopération, certaines sont dégradées ou absentes. La modalité visuelle est la plus utilisée, même si la proprioception est aussi étudiée. La limitation du champ de vision, la difficulté à percevoir les distances, l'absence de modification du flux optique lors d'un mouvement de l'opérateur sont des éléments limitant les capacités de perception de l'opérateur.

1.1.1. Prise en compte d'éléments de sciences cognitives

La question de la coopération de l'opérateur avec le robot afin de réaliser une tâche est centrale dans ce travail. La thèse d'Yves Rybarczyk [Rybarczyk 2004] a porté sur ce sujet, encadrée en co-tutelle par Etienne Colle et Philippe Hoppenot pour IBISC et Daniel Mestre pour le LMP (Laboratoire Mouvement et Perception, Marseille), laboratoire de neurosciences. L'idée initiale a été de donner à la machine le plus d'autonomie possible afin de fiabiliser le système. Devant les limites de cette approche, il a fallu traiter le système homme-machine global et il a tout naturellement fallu s'intéresser aux interactions entre l'opérateur distant et la machine. En particulier, la répartition des tâches [Gaillard 1997] ou l'allocation des fonctions [Hoc 2000] entre l'homme et la machine est une question importante. La première idée est d'attribuer une fonction à l'agent le plus à même de la mener à bien ([Fitts 1951]). Le défaut principal est que l'opérateur humain reste responsable du système global, même lorsqu'une tâche attribuée à la machine est défaillante et donc que l'opérateur humain n'est plus dans la boucle de contrôle du système [Vanderhaegen 1994]. Pour y remédier, la coopération entre l'opérateur et le robot doit être renforcée et en particulier : "le système doit faciliter l'appropriation des réponses du système par l'utilisateur" [Karsenty 1995]. [Rabardel 1993] définit un instrument comme une entité mixte qui ne peut se réduire à l'artefact, c'est-à-dire sa composante physique : il tient à la fois de l'objet et du sujet. Outre l'artefact, des schèmes d'utilisation lui sont associés, provenant d'une construction propre du sujet ou d'une utilisation sociale. Pour comprendre comment ses schèmes sont construits, on se réfère à la conception piagétienne d'adaptation à l'environnement. Selon Piaget, l'intelligence est avant tout une capacité d'adaptation qui peut se décomposer en deux processus complémentaires [Piaget 1936]. Le premier est l'assimilation qui consiste en la généralisation de schèmes préexistants. Concrètement, des schèmes développés pour un outil donné peuvent être transférés sur un autre outil qui lui ressemble par sa forme ou la situation dans laquelle on l'utilise. Le second, l'accommodation, consiste en la différenciation des schèmes préexistants. Piaget prend l'exemple de l'apprentissage de l'utilisation du bâton par l'enfant [Piaget 1936]. Dans cette expérience, un enfant est placé face à un sofa sur lequel est posée une gourde et, à portée de sa main, est laissé un bâton avec lequel il a frappé des objets auparavant. Il commence par essayer de saisir directement la gourde, puis tape avec le bâton et la fait tomber par hasard. Une fois la gourde à terre, il continue à frapper avec le bâton et finit par la pousser pour la ramener vers lui. Plus tard, en l'absence du bâton, il saisit un livre pour rapprocher la gourde. L'enfant a donc commencé par mettre en œuvre un schème existant (taper avec un bâton), mais cette assimilation de la situation au schème ne permet pas de réussir à chaque fois. Le schème va alors progressivement être accommodé afin de gérer le déplacement de la gourde

(on peut aussi généraliser à tout autre objet de taille comparable) pour aboutir à un nouveau schème : pousser avec un bâton. Enfin, celui-ci sera généralisé à d'autres objets, ici un livre.

Cette approche a été appliquée à deux aspects du robot : le positionnement de la caméra sur le robot pour la saisie d'objet avec un bras manipulateur embarqué et l'anticipation visuo-motrice pour le pilotage de la base mobile. Pour le premier, il est important de voir comment l'organisation de l'espace est perçue par un opérateur humain. D'un point de vue neuropsychologique, on distingue un espace proche et un espace lointain. Des travaux ont montré que des sujets atteints de négligence de perception entre l'espace proche et l'espace lointain se trouvent dans l'incapacité à se représenter consciemment le secteur d'espace situé à proximité, alors que d'autres ont des difficultés à se représenter l'espace lointain [Covey 1998]. Le schéma corporel résultant a des propriétés dynamiques. [Iriki 1996] montre que lorsqu'un singe utilise un outil, son espace péricorporel s'étend jusqu'au domaine atteignable par cet outil. Dans le cas de négligence pour l'espace péripersonnel, celle-ci s'étend à la dimension de la baguette [Berti 2000]. De telles observations montrent que les frontières entre ces différents espaces sont souples. Une situation de téléopération peut être comparée à l'utilisation de la baguette dans cette dernière expérience.

Dans le cas de l'anticipation visuo-motrice, l'absence ou la mauvaise restitution de certains types d'informations perceptives contraint l'opérateur à privilégier certaines modalités sensorielles sur d'autres, en particulier la modalité visuelle [Mestre 1995]. En outre, on observe une baisse de la performance à cause de la réduction des indices permettant en situation "naturelle" la perception de la structure de l'espace d'action [Massimo 1989]. Les limitations visuelles sont de deux types : temporel et spatial. D'un point de vue temporel, le délai (étudié au premier chapitre) n'est pas seul en cause. La condition de vision indirecte entraîne une absence des indices liés au flux optique, dont le rôle est majeur dans le contrôle du déplacement ([Gibson 1986], [Warren 1991]). Sur le plan spatial, la téléopération pose le problème de la limitation du champ visuel. Pour pallier ces difficultés, l'anticipation visuo-motrice semble une bonne solution comportementale. Par exemple, lors d'un mouvement de capture d'une balle, le cerveau produit une contraction des muscles environ 300 ms avant que l'objet ne touche la main [Lacquaniti 1987]. Lors du contrôle d'un déplacement, le regard anticipe la trajectoire qui va être réalisée. Pour des trajectoires courbes, l'orientation de la tête du sujet est déviée vers l'intérieur de la courbe afin de percevoir plus d'informations dans la direction dans laquelle on se dirige. Il en va de même pour le contournement d'un repère [Grasso 1998]. Ces deux études mettent en évidence la même dynamique d'orientation de la tête. Plus généralement, des études ont montré l'existence d'un rapport constant entre le rayon de courbure et la vitesse linéaire d'un mouvement (gestes d'écriture et de dessin dans un espace 2D ou 3D). Ces mouvements ont été découverts comme obéissant à ce qu'on appelle la "loi de puissance deux tiers" ([Lacquaniti 1983], [Viviani 1991]). Cette loi dit que la vitesse tangentielle est proportionnelle à la racine cubique du rayon de courbure. Elle traduit que lors des courbures importantes, le geste ralentit. Cette loi s'applique aussi à la locomotion humaine [Vieilledent 2001].

Cette approche a donné des résultats intéressants. Sur le positionnement de la caméra sur le robot, l'étude a porté sur la position du référentiel visuel par rapport à l'organe préhenseur. L'objectif principal visait à savoir si la reproduction sur le robot d'une relation œil-bras anthropomorphe facilitait l'appropriation du système par l'opérateur. L'hypothèse est que si l'opérateur inclut l'espace d'évolution du robot dans son espace proche alors il existe une bonne adaptation. Le dispositif expérimental initial est constitué du bras manipulateur et de la caméra (Figure 63). Ils sont disposés suivant une relation anthropomorphe œil-bras.

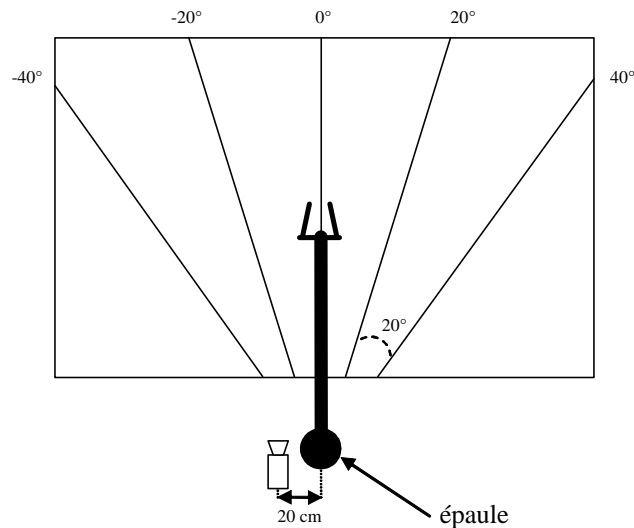


Figure 63 Caractéristiques du dispositif expérimental initial [Rybarczyk 2004].

Dans cette première expérience, le sujet doit évaluer la possibilité de saisir un objet avec le bras manipulateur (condition téléopération). Pour cela, on place l'objet à différentes distances et dans différentes directions. On en déduit la valeur D , correspondant à la distance jugée limite par l'opérateur pour saisir l'objet. En la divisant par la longueur R du bras, on obtient l'indice $\Pi = D/R$. Plus il est proche de 1, meilleure est l'estimation. Une condition de contrôle est créée en mettant un opérateur dans la même position et en lui demandant s'il peut saisir l'objet avec son propre bras. L'expérience ne révèle pas de différence statistique significative entre la valeur de Π mesurée en condition de téléopération et celle en condition de contrôle. De plus, cette précision représentationnelle s'obtient rapidement, ce qui milite en faveur d'une appropriation selon un processus d'assimilation.

Une seconde expérience a consisté à modifier la position de la caméra par rapport à celle du bras, dans la condition robotique (Figure 64). L'hypothèse est que, plus la caméra s'éloigne de la position anthropomorphe, plus l'appropriation est difficile.

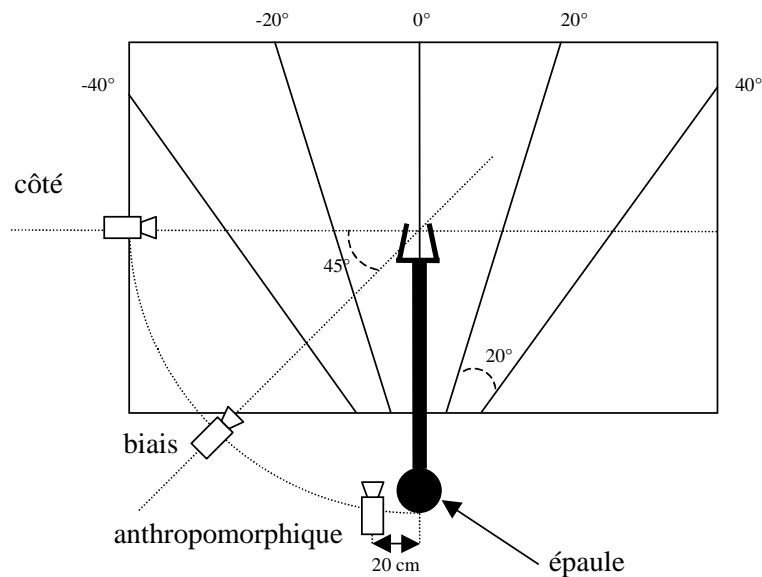


Figure 64 Caractéristiques du dispositif expérimental - Modulation de l'excentricité de la caméra [Rybarczyk 2004]

Les résultats montrent, dans un premier temps, que seules les conditions anthropomorphiques et biais fournissent une précision dans la délimitation de l'espace de saisie qui ne diffère pas significativement d'avec la condition naturelle. Il est intéressant de souligner que l'écart type de la condition biais ($\sigma=0.1$) est presque deux fois plus élevé que pour la condition anthropomorphe ($\sigma=0.06$).

Une troisième expérience analyse la dimension motrice du contrôle à distance. On demande à l'opérateur d'aller saisir des objets à différentes distances dans différentes directions dans les trois conditions de la Figure 64. Des critères de performance brute (taux de réussite et temps d'exécution) et des critères comportementaux (erreur spatiale et coordination entre ouverture de la pince et mouvements du bras) ont été retenus. Il ressort qu'une condition anthropomorphe offre à l'opérateur une performance en termes de vitesse et précision du contrôle du télémanipulateur qu'une vision de côté ne pourra jamais lui donner. Cette supériorité semble clairement due au fait que la configuration anthropomorphe fournit à l'opérateur des relations métriques connues, lui permettant d'exprimer une dynamique de mouvement de type humain et ainsi de s'appropriier la machine.

Pour l'anticipation visuo-motrice, on voit naturellement une analogie entre la direction du regard humain et l'orientation de la caméra. Deux implémentations ont été proposées : soit la caméra suit le mouvement du robot ("Je regarde là où je vais", appelée mode plate-forme), soit le robot avance dans la direction donnée par la caméra ("Je vais là où je regarde", appelée mode caméra). Deux hypothèses sont avancées : (i) les conditions avec anticipation sont plus performantes que lorsque la caméra est fixe et (ii) la condition caméra est plus naturelle (au sens appropriation par assimilation de Piaget) que la condition plate-forme. On demande à des opérateurs de piloter le robot à travers un labyrinthe et on évalue leurs performances à partir de performances brutes (temps d'exécution des parcours, nombre de collisions, nombre d'arrêts) et d'indices comportementaux (lissage des trajectoires, loi de puissance). Concernant les temps d'exécution moyen des parcours pour les trois conditions, les deux conditions

anticipatrices sont chacune significativement meilleures que la condition fixe. En ce qui concerne le nombre de collisions, seule la condition caméra donne significativement moins de collisions que la condition fixe. Quant au nombre d'arrêts, les deux conditions anticipatrices sont très significativement supérieures à la condition fixe. Le critère de lissage des trajectoires, calculé à partir des rayons de courbures, montre que la condition caméra a des conditions plus lisses que la condition plate-forme elle-même moins heurtée que la condition caméra fixe. Ces différences sont toutes significatives. Enfin, la loi de puissance, qui fait apparaître une relation entre le rayon de courbure et la vitesse linéaire lors des mouvements d'écriture ou de déplacement, n'est respectée que par la condition caméra. En définitive, les deux conditions anticipatrices sont meilleure que la condition fixe et la condition caméra est plus naturelle (loi de puissance) que la condition plate-forme.

1.1.2. Premières interface de contrôle

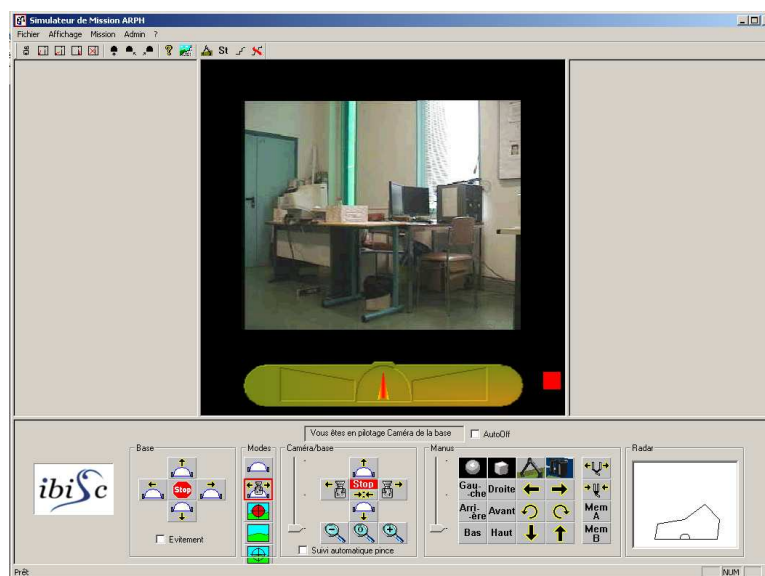


Figure 65 Première version de l'interface de contrôle [Delarue 2007]

Dans une première version de l'interface de contrôle (Figure 65), le contrôle du robot est réalisé par l'intermédiaire de la souris. Avec elle, l'opérateur appuie sur les différents boutons présents sur l'interface, pour interagir avec le robot. Cette interface permet de contrôler la position (déplacement) du robot, l'orientation de sa caméra (pan/tilt) et son bras (installé sur le robot comme dans la Figure 63). Trois modes de commandes différents sont possibles avec cette interface. Le premier est un mode manuel, où l'opérateur contrôle le déplacement de la base du robot. Le deuxième est le mode caméra, où l'opérateur contrôle le déplacement de la caméra, et le robot la suit (spécialement pour les déplacements avant/arrière). Pour le troisième mode de commande, l'opérateur contrôle le bras et, quand celui-ci est en train de dépasser certaines limites, le robot le suit, d'une façon similaire au mode caméra.

Ce qu'on peut rapidement observer est que chaque information est présentée individuellement. Même si les différents boutons sont groupés en fonction de leur type, il n'y pas de réduction de l'information affichée, par l'intermédiaire de la fusion, par exemple. La flèche qui se trouve en dessous de l'image reçue de la caméra du robot indique l'orientation de celle-ci par rapport à la base du robot. En bas et à droite de l'interface on a les mesures des

capteurs ultrason. Elles ont été agrégées autour du robot, pour offrir une vue à 180° sur les alentours du robot. La caméra est positionnée sur le robot et a un champ de vue inférieur à 180° (60° environ). De plus, si les obstacles sont trop proches du robot et s'ils ne sont pas assez hauts, ils ne seront pas visibles sur la vidéo. Étant donné aussi le fait que l'opérateur peut utiliser le bras du robot pour attraper des objets et que l'évaluation des distances dans un environnement non-structurés avec une caméra monoculaire est difficile, on peut considérer que la caméra et les capteurs de distance ultrason sont complémentaires. Si l'opérateur a besoin d'informations visuelles et de distances simultanément, il est obligé de déplacer systématiquement son regard entre la zone dédiée à la vidéo et celle pour les mesures de distance.

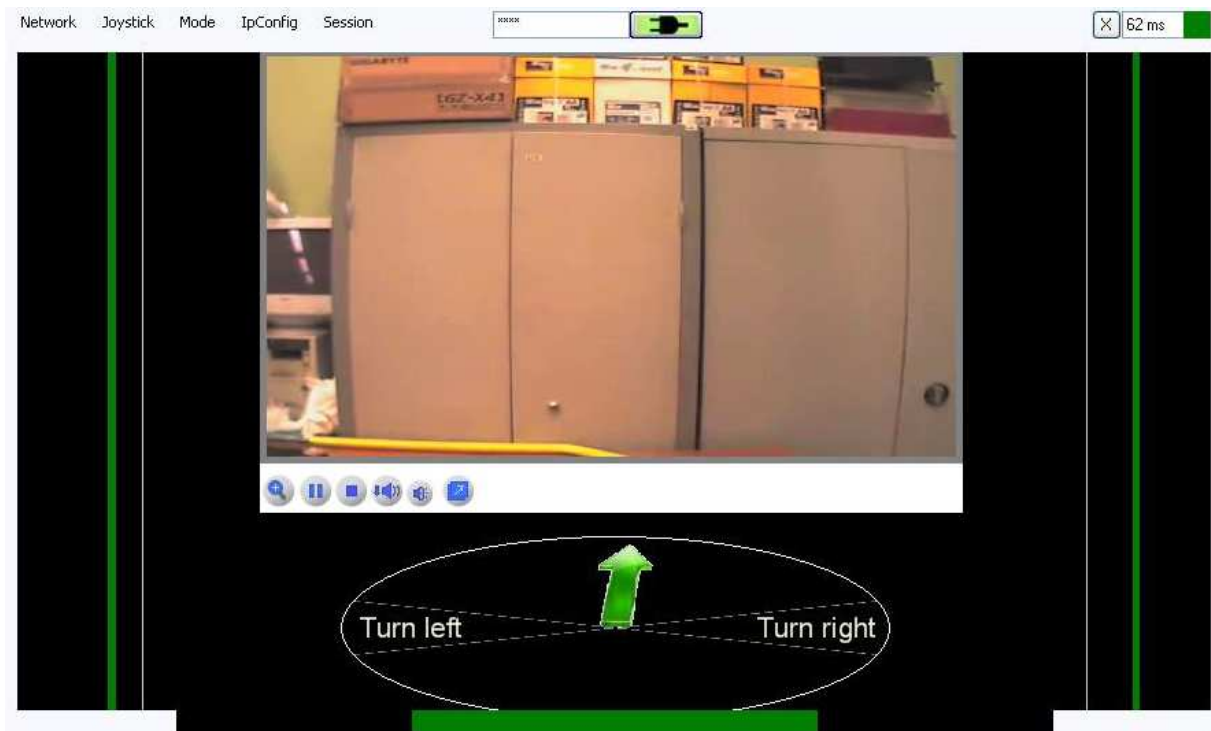


Figure 66 Interface avec barres comme indicateurs de distance

Une solution a été de fusionner les informations mises à disposition de l'opérateur par les capteurs de distance et de les présenter d'une façon synthétique, comme le montre la Figure 66. Les informations sur les distances ont été séparées en trois catégories : à l'avant du robot, à sa gauche et à sa droite. Elles sont affichées sous la forme de trois barres : les distances à gauche sont affichées à gauche de l'image, les distances à droite sont affichées à droite de l'image et les distances devant sont affichées sous l'image. Ainsi, l'opérateur doit centrer son champ visuel uniquement sur l'image, et il aura aussi les informations sur les distances accessibles. La couleur, l'épaisseur et la distance qui sépare les barres de l'image sont des indicateurs sur la proximité du robot des obstacles. Si le robot est loin d'obstacles, les barres latérales sont vertes, éloignées de la vidéo et fines. La barre pour la direction frontale est posée de façon horizontale sur l'interface. Plus le robot est proche des obstacles frontales, plus la longueur de cette barre croît. En plus, sa couleur passe de vert à rouge, suivant une transition. En ce qui concerne les barres latérales, au fur et à mesure que la distance entre le robot et les obstacles latéraux décroît, la couleur des barres change de vert vers rouge, elles se

rapprochent de la vidéo et grossissent. Bien sûr, seule la barre se trouvant dans la direction des obstacles sera affectée par ces changements.

Cette façon d'afficher les informations liées aux distances facilite leur visualisation par l'opérateur (qui n'est plus obligé à faire de longues saccades pour changer entre la vidéo et les distances). Un souci est qu'il y a une perte de détails, à cause la fusion des données en trois catégories. La partie suivante détaille une approche écologique pour la création d'une interface de contrôle du robot qui gardera au maximum les détails fournis par le capteur de distance et qui facilitera l'interprétation et l'intégration de ces distances avec les images fournies par la caméra du robot.

1.2. Approche écologique

1.2.1. Présentation

La synthèse précédente, assez longue, a permis de montrer qu'une approche anthropomorphique appliquée à la téléopération permet d'obtenir des modes de commandes naturels et faciles à utiliser. Elle a aussi permis d'introduire le mode de commande caméra que nous allons utiliser par la suite. Pour prolonger cette étude et dans la suite du travail réalisé pour prendre en compte les délais de transmission, l'objectif était de permettre à l'opérateur de mieux appréhender l'environnement du robot. L'interface de commande utilisée à ce stade est celle présentée dans le premier chapitre de ma thèse, dans la Figure 21. Il nous a semblé intéressant de prolonger le travail ci-dessus en intégrant mieux les informations disponibles pour l'opérateur, en particulier les mesures du laser et la vidéo. Le travail de thèse de Bruno Mantel ([Mantel 2009]) fournit une étude très large et complète de l'approche écologique. Pour [Gibson 1966], à l'origine de cette approche dans les années cinquante, les questions "Que perçoit-on ?", "Pour quoi faire ?" passent avant la question "Comment perçoit-on ?". Il existe une dépendance forte entre ce qui est perçu et la finalité de l'action en cours. Et vice-versa, toute action modifie notre perception.

D'après la théorie écologique, on ne perçoit pas des propriétés de notre environnement (taille, orientation, etc.), mais des possibilités d'action offertes [Gibson 1977]. C'est le concept d'affordance, formalisé dans [Warren 1984]. Il est bien entendu qu'elles dépendent de l'environnement mais aussi de l'acteur. Mantel propose l'illustration des affordances par la Figure 67. Dans le cas (a), l'obstacle à franchir est suffisamment petit pour être aisément enjambé. Dans les autres cas, l'affordance enjambement n'est plus disponible. Le franchissement pourra s'effectuer en sautant (b) ou en passant en dessous (c), cela en fonction de la taille de l'individu. Enfin, dans le cas d, le fait de porter un sac à dos lourd oblige la personne à prendre appui sur l'obstacle pour le franchir. Ainsi, bien que la personne ait la même taille que dans le cas b, les conditions particulières liées au port du sac à dos modifient ses capacités d'action et donc les affordances liées au franchissement d'obstacle. Il est important de noter que les affordances existent indépendamment du fait d'être perçues. Ce ne sont que des opportunités d'action qui doivent être perçues et faire l'objet d'une décision pour être transformées en actions.

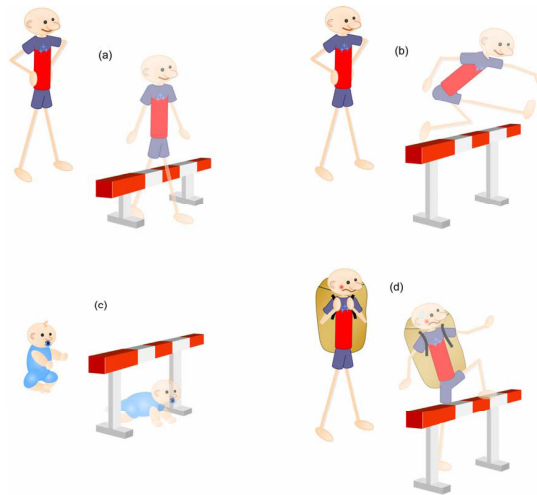


Figure 67 Illustration des affordances tirée de [Mantel 2009]

La perception d'une affordance dépend non seulement de son existence mais de la façon dont l'information est perçue par l'opérateur. Par exemple, la perception de la passabilité d'une ouverture dépend de la hauteur des yeux.

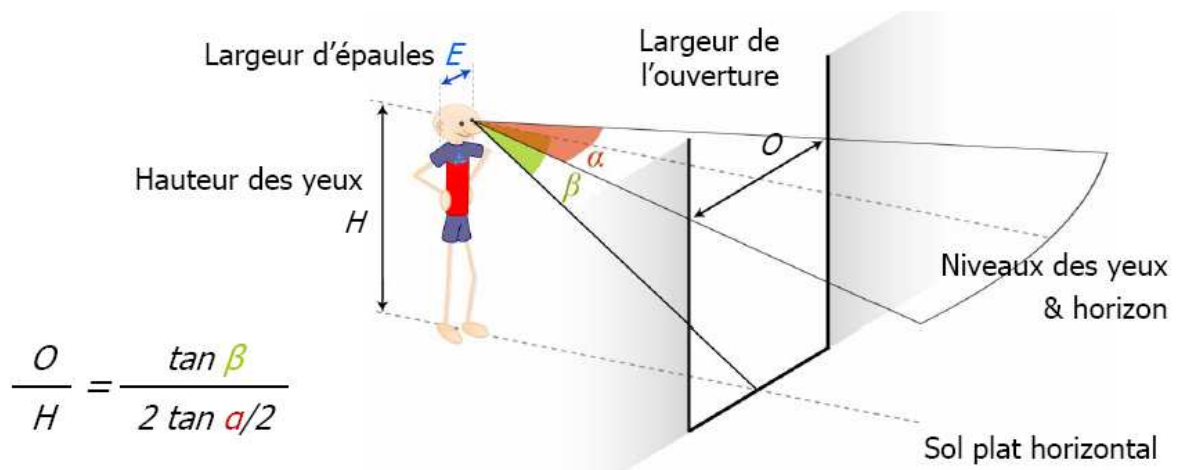


Figure 68 Perception de la passabilité d'une ouverture d'après [Mantel 2009].

Concernant la perception elle-même, l'approche écologique soutient qu'elle est directe, c'est-à-dire exploitable naturellement par l'animal [Shaw 1977]. On ne voit pas la lumière mais des objets, on n'entend pas un son mais le cri d'un animal. L'évolution de notre système de perception nous permet d'accéder directement à l'information sans passer par une phase de traitement.

1.2.2. Application à la téléopération d'un robot mobile

1.2.2.1. Démarche

L'objectif est de proposer des solutions afin de rendre les affordances liées au déplacement intelligibles par l'opérateur. Quatre approches non exclusives permettent d'en rendre compte :

- **L'anthropomorphisme.** Cette première approche est en fait celle qui a été préconisée par Yves Rybarczyk. Plus le robot a un comportement semblable à l'homme, plus la perception de son action sera facilitée à l'opérateur. Ainsi, il sera plus à même de percevoir les affordances lui permettant de réaliser une action. Par exemple, il a été montré que la passabilité d'une porte est jugée d'autant plus facilement par l'opérateur que le rapport entre la hauteur de la caméra et la largeur du robot est proche du rapport de la hauteur des yeux de l'opérateur et de sa largeur d'épaules [Moore 2009].
- **L'augmentation d'information.** C'est la plus intuitive. Elle consiste à ajouter sur l'écran de nouveaux retours d'information. En robotique téléopérée, les retours prédictifs apparaissent comme de bons exemples d'augmentation d'information [Ricks 2004], en donnant à l'opérateur des informations sur les conséquences à venir de son action. Cependant, pour que l'on puisse parler d'augmentation d'information au sens écologique du terme, l'information doit être de type fonctionnel (directement lié au contrôle du robot). Un point important consiste à rendre la notion d'échelle. Une solution est de représenter le corps du robot. Une alternative est de représenter directement des affordances à l'écran. Par exemple, il est possible de marquer sur l'interface les passages trop étroits pour le robot.
- **L'augmentation d'action.** Elle consiste à donner à l'opérateur la possibilité d'agir pour accéder à une information. Dans ce cas, l'action n'est pas directement liée à l'objectif fonctionnel de la mission. Cela peut être le cas en ajoutant une caméra mobile que l'opérateur pourrait piloter indépendamment du mouvement du robot. L'avantage de cette approche est de ne pas surcharger l'écran avec de trop nombreuses informations dont l'opérateur n'a pas forcément besoin à tout instant. C'est lui qui décide de l'information dont il a besoin et qui crée le retour utile pour la percevoir.
- **La délégation.** Elle offre une alternative différente au retour d'information. L'idée est de donner au robot une certaine autonomie d'action. Elle peut pallier une information non disponible pour l'opérateur ou simplement le décharger d'une tâche qui peut être répétitive ou difficile à réaliser par l'opérateur. C'est le cas de l'évitement d'obstacle : le robot peut s'acquitter de cette tâche sans renvoyer à l'opérateur toutes les mesures de distance qu'il utilise.

1.2.2.2. Interfaces proposées

En prenant en compte les préceptes de l'approche écologique, une nouvelle interface de contrôle a été développée. Les informations disponibles sont celles provenant de la caméra vidéo et du laser embarqués sur le robot.

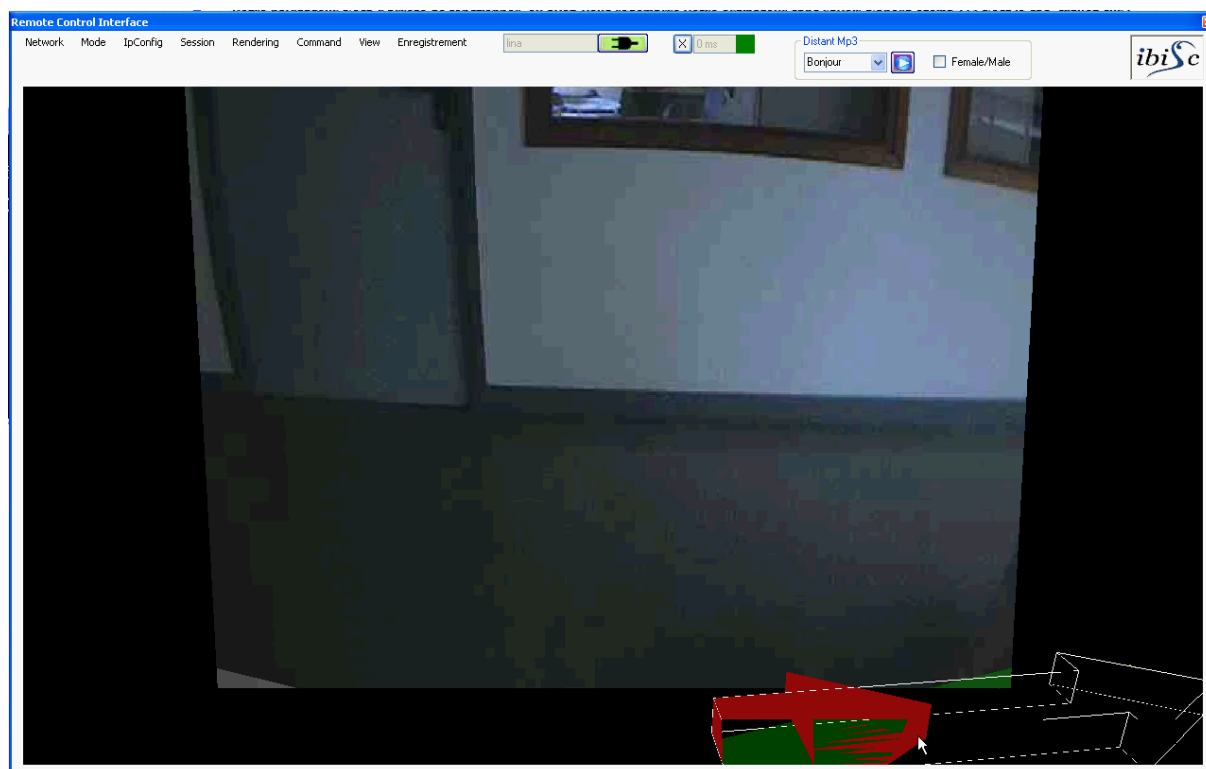


Figure 69 Image initiale

La Figure 69 montre le retour vidéo provenant de la caméra du robot. Ses inconvénients principaux sont :

- le manque d'information sur l'échelle de la scène
- le manque d'information sur les propriétés du robot
- l'étroitesse du champ de vision

Afin de donner des informations d'échelle, l'idée a été de développer une interface basée sur une reconstruction 3D, en situant l'opérateur à la place du robot (reconstruction à la première personne). Elle adopte la première approche proposée ci-dessus à savoir l'anthropomorphisme. On y voit à la fois l'image vidéo et un avatar du robot (Figure 70). Pour renforcer la notion d'échelle, une grille a été dessinée sur le sol virtuel et une ligne d'horizon a été ajoutée sur l'écran.

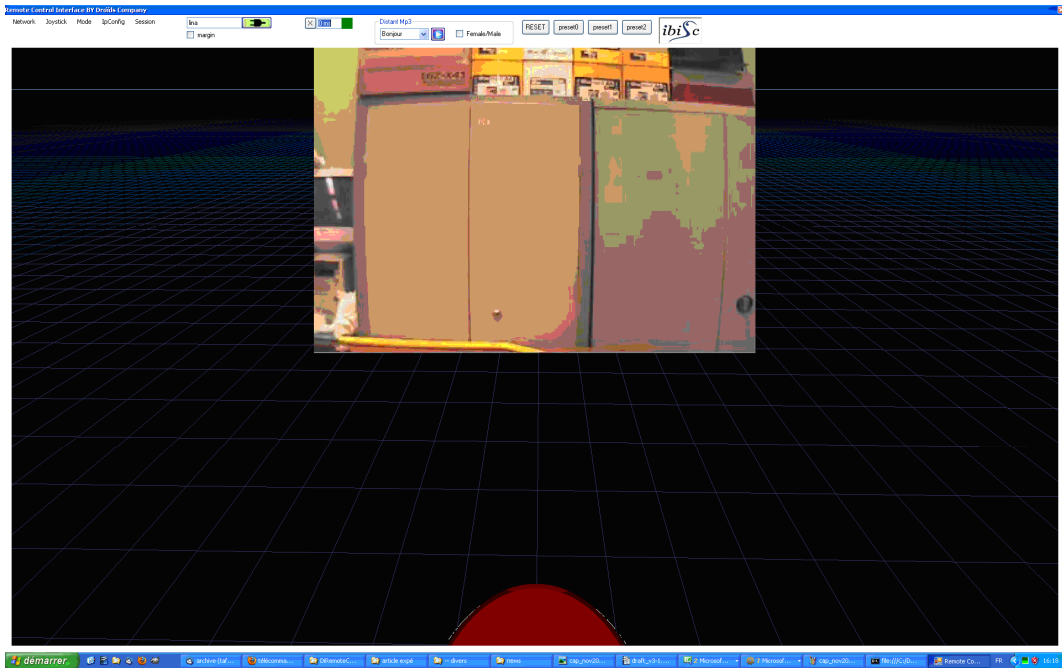


Figure 70. Vue perspective

Une connaissance précise de la position et de l'orientation de la caméra mobile est indispensable à ce niveau. On pourra ainsi extraire des paramètres permettant de garder une cohérence pendant l'affichage de l'image envoyée par le robot dans l'environnement virtuel (où la caméra virtuelle peut avoir un champ de vue supérieur à celui de la caméra réelle). Deux façons sont possibles pour déterminer ces paramètres : la première consiste à savoir exactement la position de la camera sur le robot (cinq coordonnées : x , y , z , angle de pan et de tilt – la caméra ne peut pas changer son orientation par rapport au reste du robot) et ses paramètres optiques (angle de vision vertical et horizontal, dimensions du capteur, déformations induite par l'objectif utilisé). Une autre façon est d'utiliser la boîte à outils MATLAB écrite par Jean-Yves Bouguet, qui permet l'automatisation de ce processus. Une mire doit être utilisée, comme celle de la Figure 71.

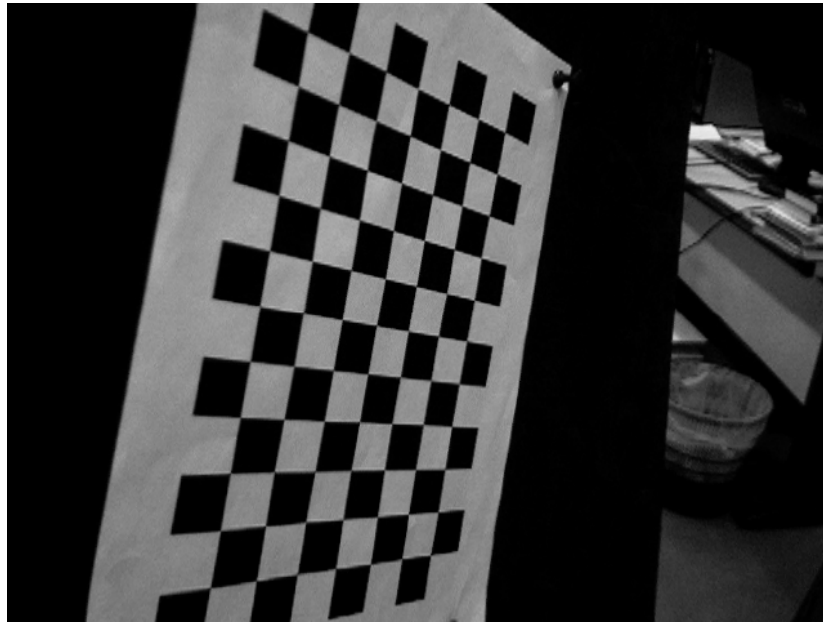


Figure 71 Mire utilisée pour la calibration de la caméra

Pour avoir une bonne estimation de ces paramètres, des captures d'image d'au moins trois positions différentes de la caméra doivent être prises. Pour chaque position de la caméra, on a besoin d'au moins cinq images, avec la mire dans des différentes positions et orientations par rapport à la caméra du robot. Les résultats obtenus à la suite de la calibration sont décrits dans l'annexe.

Après cette première étape, afin de donner à l'opérateur une vue plus large de la scène, la suite a été de représenter dans ce même référentiel les mesures du plan laser (Figure 72). On a ainsi une vision cohérente des informations disponibles pour représenter l'environnement du robot.

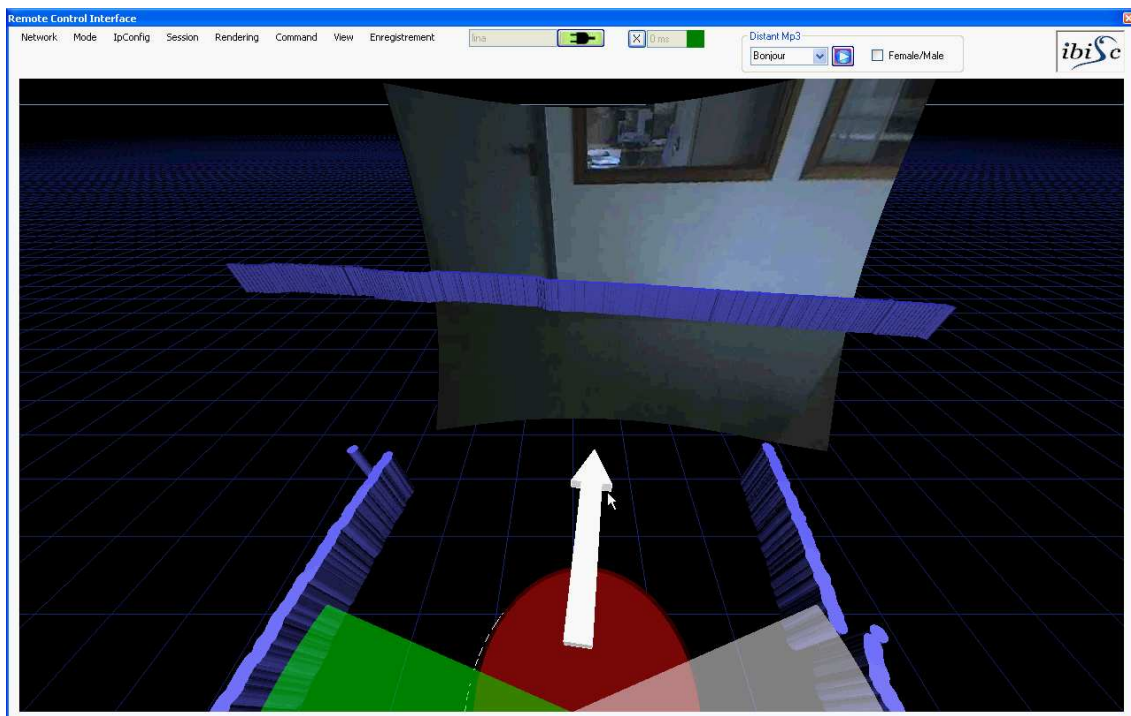


Figure 72 Représentation des mesures laser

On adopte ici la seconde approche décrite ci-dessus, à savoir l'augmentation d'information. D'autres auteurs ont utilisé une telle représentation (par exemple [Keyes 2010], [Nielsen 2007b]), mais seulement deux d'entre eux ont rendu les deux sources d'information cohérentes ([Ferland 2009], [Keskinpala 2004]). Comme pour le cas précédent, on a deux façons de faire afin d'obtenir une mise en correspondance réaliste de la caméra du robot et du capteur de distance laser, pour avoir un affichage réaliste. La première est d'utiliser les paramètres de la caméra déterminés précédemment et de mesurer l'emplacement du capteur laser par rapport à la caméra. Le souci avec cette façon de détermination est que des petites erreurs peuvent se transformer en des écarts de mise en correspondance importants, en fonction de la distance des objets détectés par le laser par rapport au robot. La deuxième façon de faire cette calibration laser-caméra est l'utilisation de la boîte à outils MATLAB écrite par Abdallah Kassir, qui permet l'automatisation de ce processus. Une mire doit être utilisée, comme celle de la Figure 73.



Figure 73 Mire utilisée pour la calibration caméra-laser

Des contraintes supplémentaires sur la taille de cette mire sont cette fois imposées, par rapport à la calibration de la caméra uniquement. La mire doit être détectée (visible) par la caméra et par le laser simultanément. Pour le cas du robot Lina, par exemple, pour une orientation horizontale (parallèle au sol) de la caméra, pour que la partie basse d'une mire qui fait un mètre sur un mètre soit visible sur l'image, celle-ci doit se trouver à quatre mètres (environ) devant le robot, ce qui impose des contraintes sur la taille minimale des cases de la mire, pour qu'elle soit utilisable par la boîte à outils MATLAB, à cause de l'objectif utilisé par la caméra

et sa résolution. Pour avoir une bonne estimation de ces paramètres, des captures d'au moins trois positions différentes de la caméra doivent être prises. Pour chaque position de la caméra, on a besoin d'au moins cinq images, avec la mire dans des différentes positions et orientations par rapport à la caméra du robot. Les résultats obtenus sont présentés dans l'annexe du manuscrit.

Sur cette base, deux interfaces ont été développées, concernant deux modes de commande différents. Le premier est un mode manuel. L'opérateur déplace une flèche blanche, parallèle au sol, pour contrôler les vitesses linéaire et angulaire du robot, comme dans l'interface présentée sur la Figure 72. L'idée originale de cette interface est que cette flèche pilote aussi une projection virtuelle de la trajectoire future du robot (Figure 74). Il s'agit d'un cas d'augmentation d'information vraiment intéressant car cela donne à l'opérateur une information fonctionnelle sur la faisabilité de la trajectoire en cours. Pendant des tests, certains opérateurs l'ont même utilisée comme une augmentation d'action au sens où ils ont piloté cette trajectoire prévue du robot sans piloter le robot lui-même, seulement pour percevoir si un passage était passable ou non par le robot.

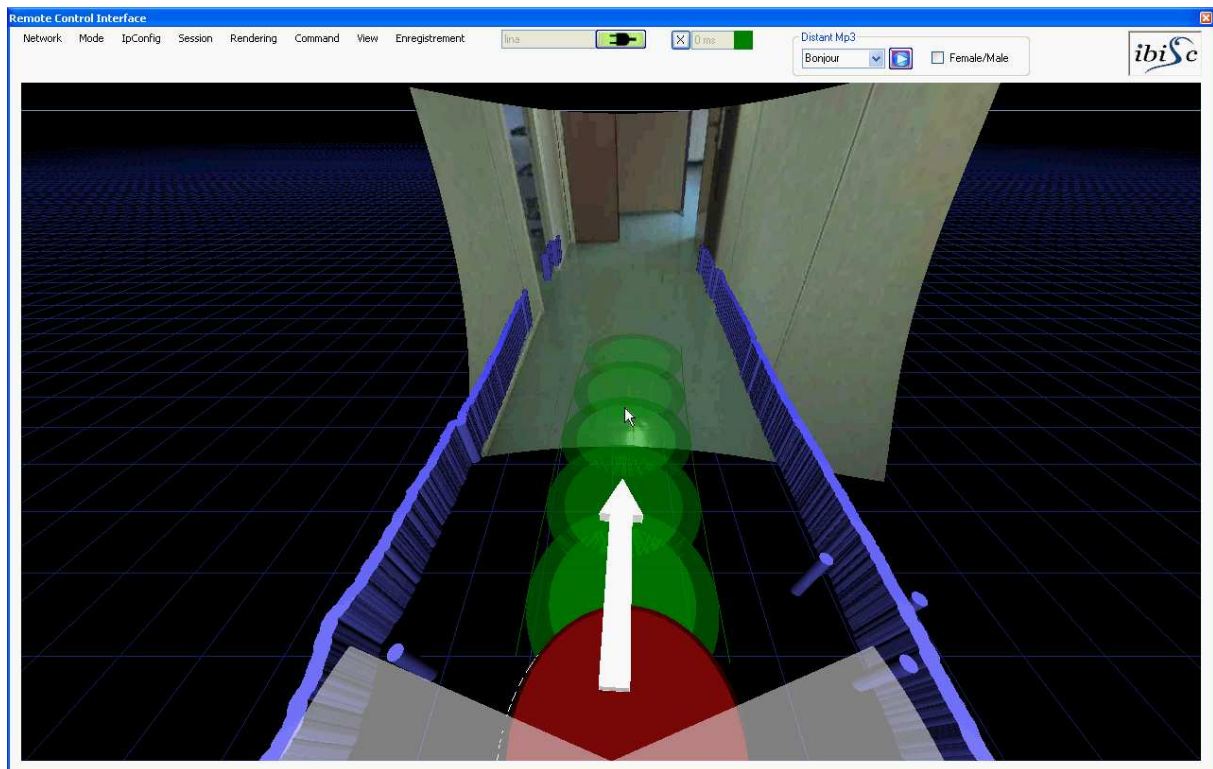


Figure 74 Projection de la trajectoire en mode manuel

Un autre mode de commande implémenté est le mode caméra. L'opérateur contrôle la caméra du robot et le robot suit l'orientation donnée par sa caméra. Du point de vue de l'affichage des informations sur l'interface de contrôle utilisée, la différence majeure est le fait que la position de l'image reçue à partir du robot n'est plus fixe. Pour le mode manuel, l'image reçue de la caméra embarquée sur le robot gardera la même position par rapport à l'interface de pilotage du robot. Pour le mode de commande caméra, les modifications de l'orientation de la caméra sur le robot réel seront traduites par des modifications de la position et l'orientation de

l'image reçue à partir du robot dans l'environnement virtuel, comme on peut l'observer dans la Figure 75.

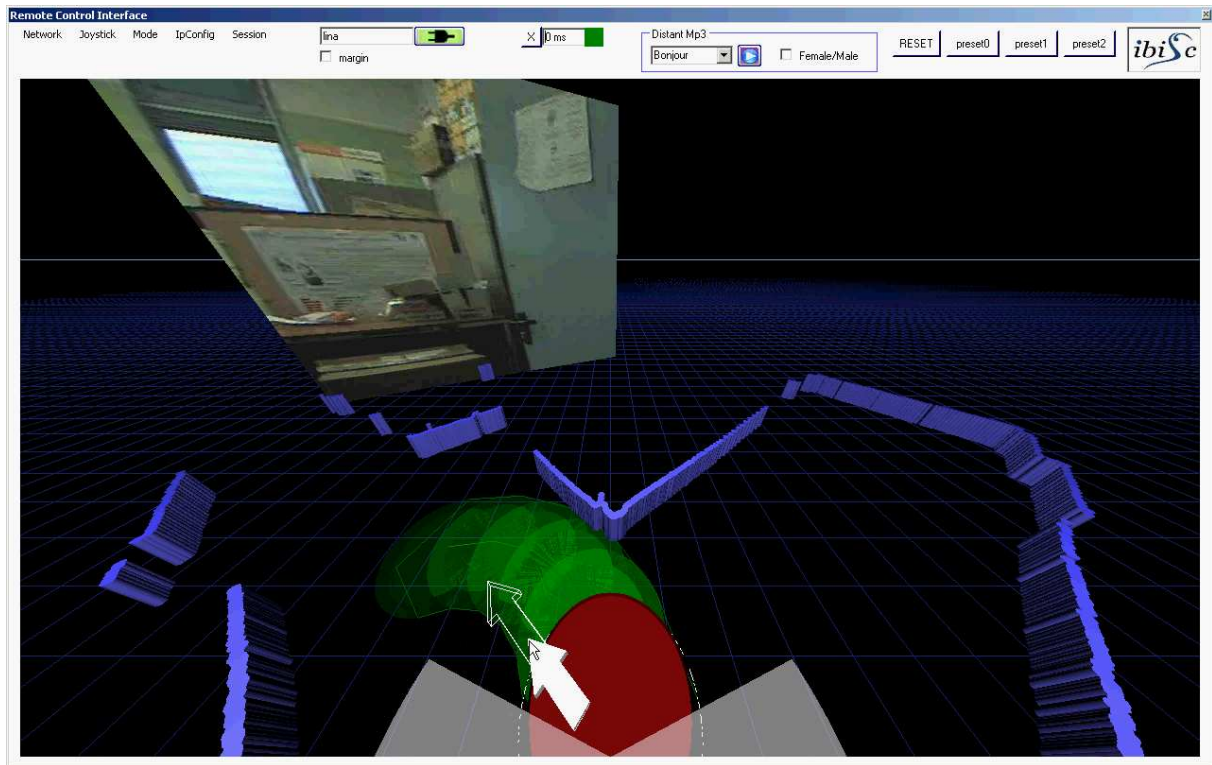


Figure 75 Mode de commande caméra

Un troisième mode de commande est le mode point à point dans lequel l'opérateur désigne un but et le robot s'y déplace automatiquement, en ligne droite. La partie commande du robot est un exemple de délégation de la fonction de déplacement au robot. Pour indiquer un but au robot, l'opérateur déplace un avatar du robot sur le sol virtuel de l'interface 3D. Cela lui permet de générer des informations à propos de la trajectoire qu'il souhaite exécuter et les affordances de passabilité disponibles (Figure 76). Une fois un premier point choisi, l'opérateur peut en choisir d'autres.

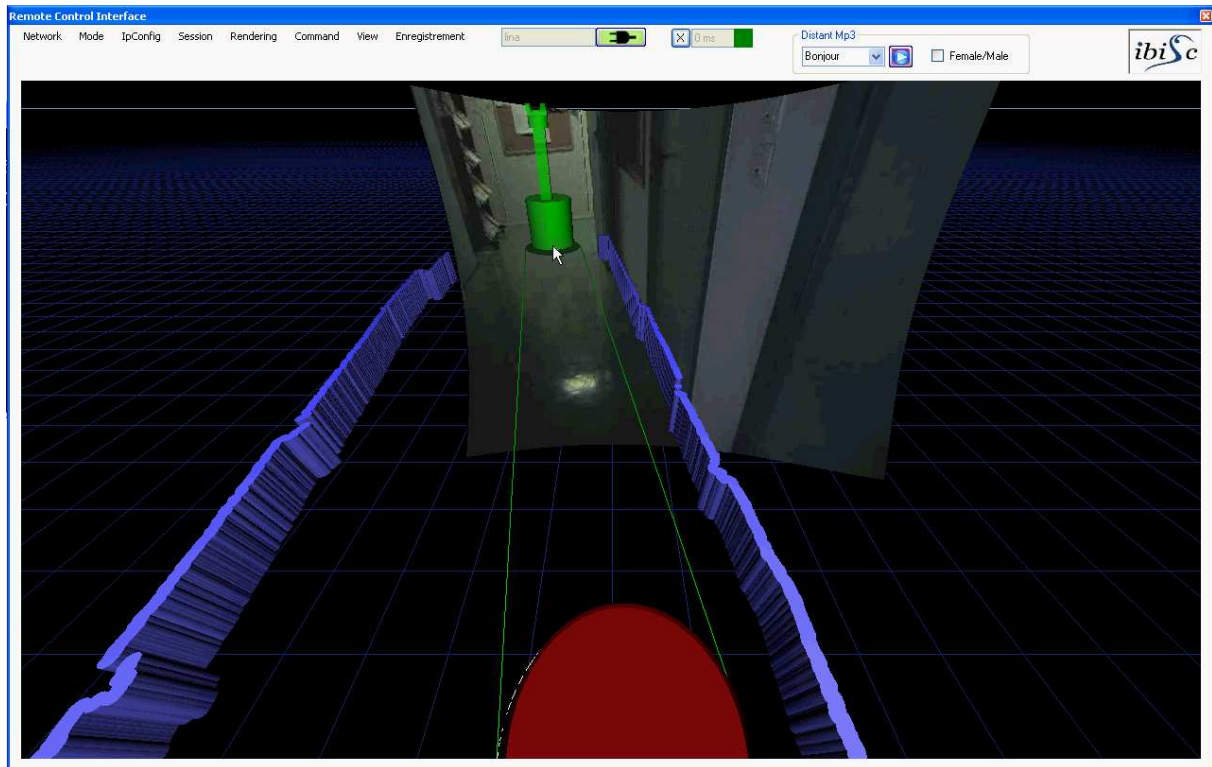


Figure 76 Mode de commande Click'n'Go

Il est aussi possible d'inclure l'aide pour les délais (décrite dans la première partie de cette thèse) sur cette interface. Si le délai est le même pour les données laser et pour les images, les deux seront modifiées : l'image sera modifiée selon la méthode décrite dans le premier chapitre et les données laser seront modifiées en fonction des commandes envoyées au robot, mais qui n'ont pas encore été exécutées. Si le délai n'affecte pas la réception des données laser, alors seule l'image sera corrigée.

Une facilité rapportée par cette interface est le fait que l'opérateur peut déplacer la caméra virtuelle, pour obtenir un point de vue qui lui convient mieux (pour observer des données laser qui sortent du champ de vue de la caméra virtuelle, par exemple). Par conséquent, la position, l'orientation et la taille de l'image réelle changeront sur l'interface présente sur l'écran de l'opérateur.

1.3. Évaluations des interfaces proposées

Cette nouvelle interface a été évaluée dans le cadre du laboratoire IBISC, afin de la comparer avec la précédente et d'observer l'influence de l'approche écologique sur les résultats obtenus par les opérateurs.

1.3.1. Protocole

L'environnement où le robot a évolué est montré dans la Figure 77. Chaque case a une longueur de 20 cm. Les trois interfaces utilisées ont été :

- L'interface où les indications concernant la proximité des obstacles sont présentées sous la forme de barres (présentée dans la Figure 66) ; le mode de commande utilisé par l'opérateur est le mode manuel

- L'interface écologique en mode manuel (présentée dans la Figure 74)
- L'interface écologique avec le mode Click'N'Go modifié (présentée dans la Figure 76) - il n'y pas d'autres points de passage pour le robot (à l'exception de la destination) et le robot s'y dirige directement (on ne fait pas de planification de trajectoire)

Douze sujets, âgés entre 21 et 33 ans, ont participé aux expériences. Chaque permutation possible de l'ordre de présentation des interfaces a été réalisée deux fois par le groupe des sujets (pour chaque permutation, il y a eu deux sujets qui l'ont eu). Avec chaque interface (et mode de contrôle), chaque sujet devait conduire le robot du point A au point B, identifier la lettre qui s'y trouvait, aller au point C, identifier la lettre qui s'y trouvait et revenir au point A, pour identifier la dernière lettre.

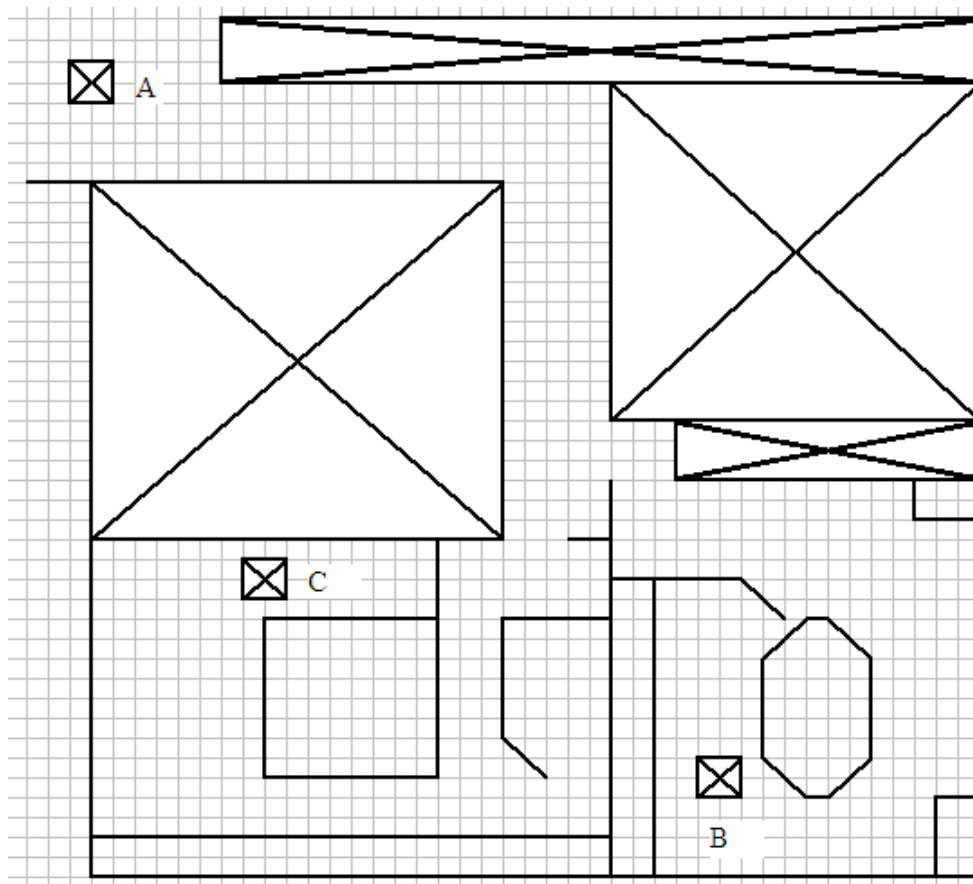


Figure 77 Environnement pour l'évaluation.

Pendant les essais, le robot n'a pas été muni de l'évitement d'obstacles. Néanmoins, pour des raisons de sécurité, il était capable de réduire sa vitesse linéaire (jusqu'à l'arrêt complet) pour éviter de toucher son environnement.

Pour comparer les résultats des opérateurs, les variables suivantes ont été enregistrées et analysées :

- La durée d'un parcours ($A \rightarrow B$, $B \rightarrow C$ ou $C \rightarrow A$) ; on suppose que plus ce temps est court, plus l'opérateur est à l'aise avec l'interface, il va plus vite et il y a moins de collisions pour ralentir le robot

- Le nombre de collisions ($v=0$) ; plus il y en a, plus la difficulté de l'opérateur à positionner le robot dans son environnement par rapport aux obstacles et à le commander de façon efficace est grande
- Le pourcentage de fonctionnement du système anticollision (potentiellement sans arrêter le robot) ; cet indicateur est utilisé pour examiner la facilité globale de piloter le robot ; plus ce pourcentage est grand, plus l'opérateur a utilisé le robot dans des zones dangereuses (trop près des obstacles) ; ce système anticollision n'est pas un évitement d'obstacle, son but n'est pas de faciliter le pilotage du robot par l'opérateur ; ce système modifie la vitesse linéaire, pour que le robot ne touche pas l'obstacle, il ne prend pas de mesures pour orienter le robot (afin qu'il passe à côté de l'obstacle)
- Le pourcentage de temps pendant lequel le robot n'a pas bougé ; cet indicateur est utilisé pour examiner les difficultés rencontrées par les opérateurs après une collision (le temps qu'ils mettent pour revenir dans une situation normale)
- Le rayon de courbure de la trajectoire ; cet indicateur donne une idée sur le confort et l'efficacité de l'interface ; s'il est trop petit, ça signifie que les opérateurs effectuent beaucoup de virages avec le robot, ce qui ne devrait pas être le cas avec les parcours qu'on leur a proposés

A la fin de chaque séance d'expériences, un questionnaire subjectif a été présenté aux participants. Ils ont été priés de classer les trois interfaces selon leurs préférences et, pour chaque interface, de mettre un trait sur une ligne ayant zéro et dix comme valeurs extrêmes, pour les quatre affirmations suivantes :

- L'interface est facile à utiliser
- L'interface est pratique
- L'interface est agréable
- L'interface est stressante

Plus le trait que les sujets tiraient sur la ligne était vers la droite (vers dix), plus la propriété en question de l'interface était importante.

1.3.2. Résultats

Les résultats ont été obtenus de deux façons : en observant les performances des sujets pendant le contrôle du robot et en examinant les questionnaires qu'ils ont remplis à la fin de leur séance. La première façon est objective, alors que la deuxième est subjective, mais elle permet de capturer les opinions des sujets par rapport à chaque interface, même s'il serait théoriquement possible que l'interface qu'un sujet a mieux aimé a donné les pires résultats.

1.3.2.1. Résultats objectifs

Les conclusions qu'on a tirées ont été basées sur les mesures suivantes : la durée des parcours, la courbure des trajectoires, le pourcentage de temps pendant lequel le robot n'a pas bougé et les collisions (leurs nombre et le pourcentage de temps pendant lequel le système anticollisions a fonctionné).

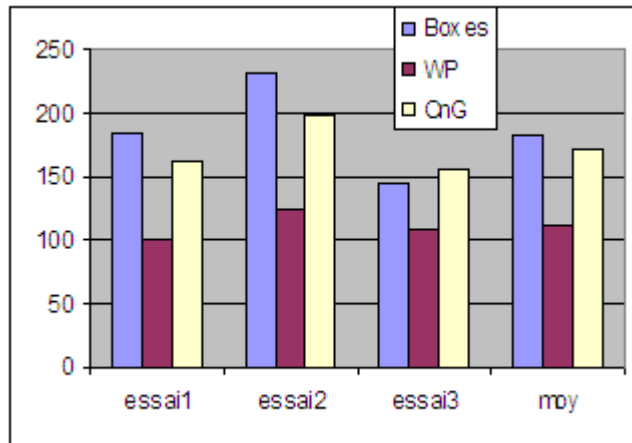


Figure 78 Durées des parcours (secondes)

Dans la Figure 78, le premier essai correspond au parcours A-B de la Figure 77, le deuxième essai au parcours B-C, et le troisième au parcours C-A. Le dernier groupe de colonnes est la moyenne. La première colonne (Boxes) représente l'interface classique (qui a, comme indication pour les distances, des barres latérales). La deuxième colonne (WP) correspond à l'interface présentée dans la Figure 74. La dernière colonne (CnG) correspond à l'interface présentée dans la Figure 76. L'interface WP obtient le meilleur temps moyen (111,02 s), suivi par CnG (171,68 s) et par Boxes (184,15 s).

A la suite d'une ANOVA, on arrive à la conclusion que l'interface WP est statistiquement différente par rapport aux autres. Par rapport à Boxes, $p=0,0002$ et par rapport à CnG, $p=0,0006$. La différence entre Boxes et CnG n'est pas significative ($p=0,5167$), en utilisant le temps comme critère.

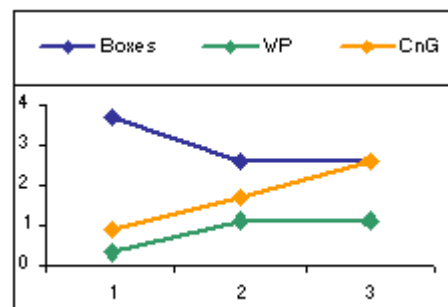


Figure 79 Nombre de collisions

La Figure 79 montre le nombre de collisions, par parcours. Pour l'interface Boxes, le premier parcours pose des problèmes particuliers aux opérateurs, surtout par rapport à l'interface WP ($p = 0,037$), la différence par rapport à l'interface CnG n'étant pas significative.

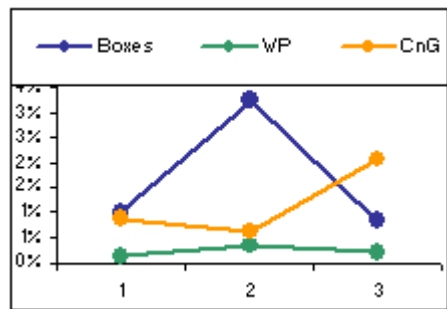


Figure 80 Robot arrêté

Dans la Figure 80 on observe le pourcentage de temps pendant lequel le robot n'a pas bougé. Le deuxième parcours (B-C) a été plus difficile pour l'interface Boxes, mais la différence n'est pas significative.

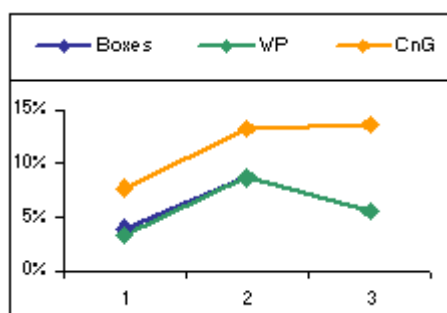


Figure 81 Fonctionnement anticollisions

La Figure 81 montre le pourcentage de temps pendant lequel le système d'évitement de collisions a fonctionné (il est possible pour le robot de continuer son déplacement, mais avec une vitesse linéaire réduite ou nulle). L'interface CnG a demandé des interventions plus longues (troisième parcours – $p = 0,0006$) à cause de son mode de fonctionnement (interventions ponctuelles de l'opérateur) et du fait que le robot, à cause d'un problème mécanique (pression différente dans la roue droite par rapport à la gauche, sans possibilité de correction), était biaisé vers la droite. En permettant l'utilisation de l'évitement d'obstacles et en corrigeant les soucis mécaniques du robot, les problèmes liés aux collisions disparaîtront. L'idée principale de cette campagne d'évaluations a été d'évaluer les évolutions de la nouvelle interface par rapport à la précédente, ce qui signifie qu'on s'est concentré sur le mode de commande manuel. Le mode de commande CnG a été surtout testé du point de vue de sa facilité d'utilisation avec la dernière version de l'interface (conçue selon l'approche écologique). Pour rappel, le but principal de cette série d'évaluations a été de comparer la nouvelle interface avec la précédente. L'évitement d'obstacles a été remplacé par une sécurité, pour éviter d'endommager le robot. Cette sécurité modifie uniquement la vitesse linéaire du robot, pour l'arrêter avant que le robot touche un obstacle. Ceci nous a permis de comparer les interfaces entre elles (l'interface précédente avec l'interface écologique), en non pas le couple évitement-interface précédente avec le couple évitement-interface écologique. Ceci signifie que les résultats obtenus pendant cette campagne d'évaluations avec ce mode de commande (surtout en ce qui concerne les collisions) n'illustrent pas les performances qu'on s'attend à obtenir avec une implémentation complète de ce mode.

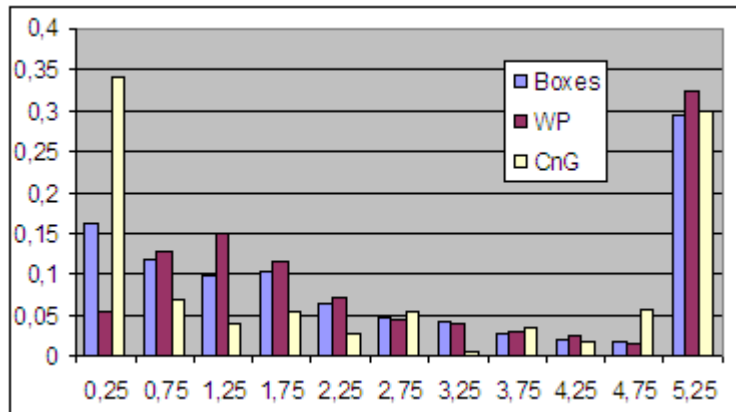


Figure 82 Courbure de la trajectoire

La courbure de la trajectoire est présentée dans la Figure 82. L'hypothèse qu'on utilise pour son interprétation est qu'une courbure trop petite est une conséquence de l'inconfort de l'opérateur avec l'interface. Sur l'abscisse on a des intervalles pour les rayons de courbures (espacés de 0,5 m) et sur l'ordonnée on a le pourcentage de la trajectoire totale. Pour les rayons inférieurs à 0,25, on a l'interface CnG très présente (à cause de son mode de fonctionnement, on ne peut pas spécifier l'orientation finale du robot, on est obligé à le tourner sur place), suivi par l'interface Boxes et par WP. Pour l'intervalle 0,75 – 1,25, on observe une domination de l'interface WP ($p = 0,01$). Ceci est proche de la largeur moyenne des espaces que le robot a eu à traverser, les opérateurs ont donc eu moins de mal (moins de corrections à effectuer) avec ce mode qu'avec le mode Boxes.

1.3.2.2. Résultats subjectifs

Les résultats subjectifs ont été obtenus en analysant les questionnaires auxquels les sujets ont répondu à la fin des séances d'expériences. La Figure 83 synthétise leurs réponses.

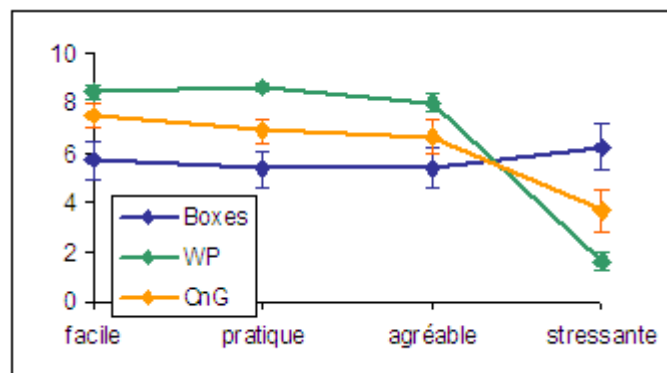


Figure 83 Résultats subjectives

L'interface WP est préférée aux deux autres sur tous les critères, à l'exception de la facilité d'utilisation par rapport à l'interface CnG ($p = 0,075$). Pour les trois autres critères, la différence est significative. L'interface CnG n'est pas significativement différente de l'interface Boxes, à l'exception du stress engendré ($p = 0,0369$).

Si on regarde le classement réalisé par les sujets, en ordre de préférence des interfaces, l'interface WP est classée en premier par deux tiers des sujets, l'interface CnG est classée en

deuxième par deux tiers des sujets et l'interface Boxes est classée en troisième par deux tiers des sujets.

1.3.3. Conclusions

La préférence manifestée par les sujets pour l'interface WP confirme les performances obtenues avec elle. C'est cette interface qui enregistre le temps le plus court. Un rayon de courbure modéré (plus important que pour Boxes) est également produit par cette interface, qui semble indiquer le fait que les sujets sont plus à l'aise avec elle et n'ont pas besoin de modifier en permanence, avec des valeurs importantes, la trajectoire du robot. Si on regarde les collisions, c'est toujours l'interface WP qui est meilleure que l'interface Boxes.

L'interface CnG a été incluse dans cette étude dans une étape préliminaire de son implémentation, surtout pour observer son acceptabilité par les sujets. Les résultats sont encourageants, elle est considérée comme étant moins stressante que l'ancienne interface. Comme suite de ceci, il est très probable que le temps moyen diminuera. Le rayon de courbure réduit de la trajectoire est dû au fonctionnement de ce mode. On compte réaliser une deuxième campagne d'évaluations avec une nouvelle version de ce mode (similaire à celle décrite dans le deuxième chapitre), qui nous permettra de tirer de conclusions plus précises sur son utilisation par les opérateurs.

2. Changement de mode de commande

Dans la partie précédente de ce chapitre on a analysé la genèse d'une nouvelle interface de commande, qui offre la possibilité d'observer plus facilement l'environnement du robot (les obstacles qui se trouvent autour de lui) sans faire bouger la caméra ou le déplacer. On a vu également un nouveau mode de commande (point-destination – Click'n'Go) et comment son utilisation était facilitée par cette nouvelle interface. Dans la suite on essayera de trouver comment faire pour passer d'un mode de commande à un autre, en utilisant la même interface de pilotage du robot par l'opérateur humain. Des brefs rappels sur le fonctionnement du système avec délais et sur la commande en environnement inconnu seront faits. Une présentation de l'automatisation du système homme-machine, des différentes façons utilisées pour classer les niveaux d'automatisation et des schémas pour le changement du niveau d'automatisation sera faite à la suite. Un exemple typique d'utilisation du système qu'on utilise est donné, pour qu'ensuite on analyse les façons de changer de niveau d'automatisation pour notre cas.

2.1. Téléopération avec délais

Dans la première partie on a pu voir une méthode pour aider le téléopérateur à piloter le robot en cas de délai de communication entre le site maître (celui du l'humain) et le site esclave (là où le robot se trouve). Pour rappel, cette méthode a employé un modèle du robot, appelé robot virtuel, pour que l'interface de commande puisse faire une prédiction concernant la distance et l'angle que le robot décrira, à la suite de l'exécution des ordres envoyés par l'interface et qui sont en train d'arriver au robot. Cette distance et cet angle sont traduits, sur l'interface de l'opérateur, par des modifications de l'image. La correspondance linéaire est traduite par un

zoom sur l'image et la composante angulaire α , comme correspondance, un mouvement de translation de l'image. La Figure 84 montre une situation-type. L'opérateur souhaite que le robot avance et tourne à gauche. Normalement, il verrait l'image de droite, parce que le robot n'a pas encore reçu les commandes qu'il a envoyées. Mais, grâce à l'utilisation de l'aide, l'image de gauche lui sera affichée à l'écran. Pour les virages, comme le robot ne possède pas une caméra capable de percevoir une partie importante de son environnement, une bande noire sera affichée dans la partie correspondante à ce manque d'information.



Figure 84 Exemple d'image (droite) modifiée par l'aide (gauche)

Ce mode de commande, comme le mode caméra, représente essentiellement une évolution du mode de commande manuel. Tous ces modes de commande sont des modes de commande continus, où l'intervention de l'opérateur est nécessaire à chaque moment. Si l'opérateur cesse d'intervenir, alors le robot arrête son déplacement.

2.2. Commande en environnement inconnu

Comme on l'a déjà décrit dans le deuxième chapitre de cette thèse, l'opérateur indique une destination au robot, que celui-ci doit atteindre par ses propres moyens. Le robot est donc tenu d'utiliser ses propres moyens de planification de la trajectoire et d'évitement d'obstacles pour atteindre la destination qui lui a été indiquée. Dans sa version la plus avancée, le robot doit aussi faire de la replanification, s'il se rend compte que le chemin qu'il a pris n'est pas le bon, pour des raisons comme un éloignement trop important de son objectif ou l'arrivée dans une impasse.

Plusieurs déclinaisons de ce mode ont été réalisées. Dans la plus basique on n'a pas la partie replanification. Cette version est utile pour l'opérateur novice, qui n'a pas beaucoup de compétences dans l'utilisation du mode manuel et qui n'a pas non plus trop de confiance dans l'évitement d'obstacles du robot. Un exemple typique pour notre contexte est le passage d'une porte. Un seul point-destination est utilisé dans ce cas, et le robot s'y dirige en ligne droite (s'il est orienté dans la bonne direction, sinon, le mouvement commencera sur un arc de cercle plus ou moins important, pour que le robot soit orienté d'une bonne manière). Une évolution est la possibilité de donner une liste de points au robot, qui seront parcourus en ordre par celui-ci. Il est également possible d'avoir la replanification activée pour ces deux modes d'organisation d'ordres.

Par rapport à la famille du mode de commande manuel, ce type de commande est un type de commande discret. L'intervention de l'opérateur n'est nécessaire qu'au début (quand il indique le point-destination, avec ou sans des points supplémentaires de passage). La plupart

du temps, les capacités de l'opérateur ne sont pas utilisées (même s'il peut intervenir à tout moment pour arrêter le robot ou pour lui indiquer une nouvelle destination).

2.3. Automatisation du système homme-machine

Quand le système homme(s)-machine(s) devient trop complexe pour être facilement gérable par la partie humaine sans qu'elle soit aidée, l'automatisation du système peut apporter une réponse à la question de comment faire pour que le système continue à accomplir ses fonctions.

L'automatisation du système a comme but son maintien dans un état fonctionnel. Ce problème est attaqué par deux côtés, pas nécessairement exclusifs. La première façon est d'améliorer l'autonomie du système, ses capacités d'autoanalyse et d'autocorrection. La deuxième est d'aider l'opérateur dans sa prise de décision et dans son implémentation.

2.3.1. États d'automatisation d'un système homme-machine

Une des façons les plus utilisées pour classifier l'état d'automatisation d'un système homme-machine est celle qui a été proposée par [Sheridan 1992]. Dix niveaux ont été proposés :

- Niveau manuel – l'opérateur humain est responsable pour toutes les décisions et les actions, la machine n'offre pas d'assistance
- La machine présente un ensemble complet de décisions/actions alternatives
- La machine diminue la liste de choix à quelques uns
- La machine exécute l'action si l'opérateur humain l'approuve
- La machine laisse un certain temps à l'opérateur de s'opposer et, sinon, exécute l'action qu'elle a sélectionnée
- La machine exécute l'action de façon automatique et informe l'opérateur
- La machine exécute l'action de façon automatique et informe l'opérateur uniquement si celui-ci le lui demande
- La machine exécute l'action de façon automatique et informe l'opérateur uniquement si elle le décide
- Niveau automatique – la machine décide tout, actionne de façon autonome, l'opérateur humain est réduit au rôle d'observateur

Plus tard, un niveau supplémentaire (6,5) a été rajouté par [Inagaki 1998]. Celui-ci consiste à informer l'opérateur et, après, l'automatisation exécute l'action. Ceci a deux raisons : prévenir l'opérateur pour réduire les surprises générées par l'automatisation du système et garder l'efficacité de l'automatisation dans des cas d'urgence.

A la classification proposée par [Sheridan 1992], [Endsley 1999] rajoutent quatre axes concernant les informations et les actions. Ces axes sont les suivantes :

- Acquisition de l'information
- Analyse de l'information
- Sélection de l'option (action à implémenter)
- Implémentation de l'action (option)

Ainsi, ils caractérisent chaque niveau proposé par [Sheridan 1992] selon ces quatre axes. « H » indique le fait que l'opérateur humain est en charge de la tâche, et « A » indique le fait que l'automatisation du système est responsable de la tâche en question.

	Acquisition	Analyse	Sélection	Implementation
1 (manuel)	H	H	H	H
2	H/A	H	H	H/A
3	H/A	H	H	A
4	H/A	H/A	H	H/A
5	H/A	H/A	H	A
6	H/A	H/A	H/A	A
7	H/A	A	H	A
8	H/A	H/A	A	A
9	H/A	A	A	A
10 (automatique)	A	A	A	A

Tableau 2 Niveaux d'automatisation [Sheridan 1992]

[Billings 1997] considère que les dix niveaux que [Sheridan 1992] avait proposés pour un système homme-machine ne sont pas parfaitement adaptés pour un système homme-robot plus réduit et propose les niveaux suivants :

Niveau	Rôle de la machine	Rôle de l'opérateur
Contrôle manuel direct	L'automatisation n'est pas utilisée	L'opérateur contrôle toutes les fonctions et tâches
Contrôle manuel assisté	Contrôle manuel principalement, avec un peu de support de la part de l'automatisation	L'opérateur contrôle manuellement, avec assistance de la part de l'automatisation partielle
Contrôle partagé	Contrôle automatique de quelques tâches	L'opérateur contrôle quelques fonctions/tâches
Opération par délégation	Contrôle automatique quand il est indiqué par l'opérateur	L'opérateur donne des commandes de supervision que l'automatisation suit
Opération par consensus	Contrôle automatique complet, avec une supervision de près de la part de l'opérateur	L'opérateur supervise de près, approuve les actions et peut intervenir
Opération par exception	Fonctionnement autonome, essentiellement, sauf si des circonstances spécifiques sont rencontrées	L'opérateur approuve les décisions critiques et peut intervenir
Opération autonome	Opération autonome complète ; l'opérateur n'est pas informé, habituellement, ; le système est, ou pas, capable d'être désactivé	L'opérateur n'a pas, en règle générale, un rôle dans le système, et la monitorisation est limitée

Tableau 3 Niveaux d'automatisation [Billings 1997]

Pour toutes les classifications proposées, une remarque générale est qu'elles peuvent s'appliquer à tout le système homme-machine, mais il est également possible de décomposer le système dans des sous-parties, et de les classifier séparément, ce qui est parfois plus pertinent, surtout pour ceux qui ont une dimension importante [O'Hara 2010].

2.3.2. Changement du niveau d'automatisation par la machine

Quand la machine est capable de changer le niveau d'automatisation du système, on dit que l'automatisation du système homme-machine est adaptative (« adaptive », en Anglais). Dans la littérature, le changement peut se faire à partir de mesures physiques, à partir de mesures de performance, ou les deux.

Le changement basé sur des mesures physiques est de deux types : intrusif (avec contact) ou non. Un exemple de capteur de mesures physiques non-intrusives est l'oculomètre, qui est capable de déterminer, après une étape initiale de calibration, le point où l'opérateur humain regarde sur l'écran. A partir de ce type d'information le système peut déterminer s'il a pris connaissance des nouveaux éléments, s'il les traite ou s'il est en train de chercher des informations sur eux. Il a aussi été établi le fait qu'il y a une relation entre le nombre de clignements des yeux et la charge de travail visuelle. Quand la charge de travail augmente, le taux de clignotement se réduit. De l'autre côté, une fréquence de clignotements accrue, ainsi que leur durée, peut signifier que l'opérateur est fatigué et qu'il peut avoir besoin de dormir.

Les capteurs intrusifs (avec contact) sont utilisés pour déterminer, par exemple, le pouls de la personne, la conductance de sa peau (la transpiration), l'activité cérébrale ou musculaire, la respiration, etc.

Les mesures de l'activité cérébrale sont obtenues en utilisant des électrodes placées sur le scalp. La majorité des mesures ont une fréquence comprise entre 0-400 Hz. Il est possible d'isoler l'activité d'une certaine région (visuelle, auditive, régions exécutives frontales). Aussi la fatigue est détectable en analysant des changements dans la puissance des certaines fréquences (delta, thêta, alpha, beta et gamma). Un problème avec l'acquisition de ces signaux est le fait que le cerveau n'a pas exactement la même organisation pour toutes les personnes et que des différentes stratégies et styles sont associés avec différents modèles d'activités cérébrales. Un souci supplémentaire est le fait que des mouvements des yeux et des clignements des yeux produisent des potentiels de champs éloignés qui sont détectés par les électrodes placés sur le scalp. Il est possible d'améliorer la qualité de ces signaux, mais des méthodes intrusives devront être employés (perçement du crâne).

En ce qui concerne le pouls, des incréments dans la charge de travail ont été associés avec des pouls plus importants. Aussi l'intervalle entre deux pulsations a été mis en relation avec des changements dans la charge cognitive.

Pour la respiration, des cycles peu profonds et rapides ont été mis en relation avec des augmentations du stress et de la charge cognitive. La respiration a aussi des effets importants sur la fréquence cardiaque et sa variabilité. La décomposition des cycles respiratoires permet de vérifier le degré avec lequel les changements cardiaques sont des artefacts de la variabilité respiratoire.

L'activité musculaire est associée avec des fréquences supérieures à 10 Hz, et elle est très présente entre 50 et 150 Hz. De tels mesures sont en liaison directe avec des charges de travail physiques, plus l'amplitude des mesures est importante, plus les actes physiques demandés sont onéreux. Par conséquent, des altérations dans la charge périphérique (associée avec l'utilisation des dispositifs de contrôle, comme les joysticks), peuvent être analysées en temps

réel. Des mesures musculaires sont aussi associées avec la fatigue. Pendant des moments de fatigue, ou pendant le sommeil, il y a une diminution progressive de l'activité musculaire. Ainsi, une diminution dans le tonus musculaire peut indiquer des niveaux dangereusement bas d'attention. Au contraire, des états de forte excitation sont associés avec un tonus augmenté.

L'activité cérébrale est utilisée par [Russell 2005] pour aider les opérateurs de drones. [John 2003] utilise plusieurs mesures : l'activité cérébrale, l'oculométrie, la pression d'appui sur la souris, le pouls et la réponse galvanique de la peau. Dans [Prinzel 2000] le système essaye de maintenir un niveau optimal de la charge cognitive de l'opérateur, en utilisant des mesures d'activité cérébrale. La performance du groupe expérimental a été améliorée de façon significative par rapport au groupe de contrôle. [Prinzel 2003] utilisent plusieurs mesures (cérébrales, pouls) pour l'automatisation adaptative. Les résultats montrent que ces mesures peuvent être exploitées pour réaliser une automatisation adaptative. [Prendinger 2007] utilise l'oculomètre pour évaluer l'utilité des agents d'interface. Les gestes de l'agent sont plus efficaces pour diriger l'attention de l'opérateur vers des objets utiles, avec un petit coût lié à l'inspection visuelle de l'objet.

Une question, abordée par [Dekker 2010], concerne le fondement théorique du choix de ces indicateurs (charge de travail cognitive). Des questions auxiliaires qu'on pourrait poser concernent l'unité de mesure pour la charge de travail, comment s'assurer que les mesures obtenues n'ont pas une autre source que l'utilisation du système et comment choisir les seuils pour le déclenchement de l'automatisation. En ce qui concerne l'unité de mesure, des mesures obtenues à partir de capteurs de type différents devraient être ramenées au même système de notation pour pouvoir garder une certaine généralité pour une architecture proposée, par exemple. En ce qui concerne la source des mesures, idéalement il faudrait avoir aussi une connaissance sur l'environnement de l'opérateur, pour pouvoir prendre en compte les facteurs perturbateurs externes qui pourrait intervenir et déranger l'opérateur humain pendant qu'il est en train d'utiliser le système. Les facteurs internes à l'opérateur peuvent aussi perturber les mesures, mais ils sont beaucoup plus difficiles à évaluer. La question du seuil est importante à cause de la variation humaine. Une calibration du système pour un opérateur pourrait s'avérer fautive pour un autre opérateur. L'opérateur peut présenter des seuils variables aussi (à cause de la fatigue ou d'une sollicitation multitâche, par exemple).

Par rapport à la charge de travail, les performances du système sont plus simples à mesurer. Il faut, bien sûr, définir les indicateurs qui fourniront les données nécessaires pour calculer les performances.

Les performances du système ne sont pas indépendantes de la charge de travail de l'opérateur. Un exemple de dépendance entre les deux est montré dans la Figure 85 :

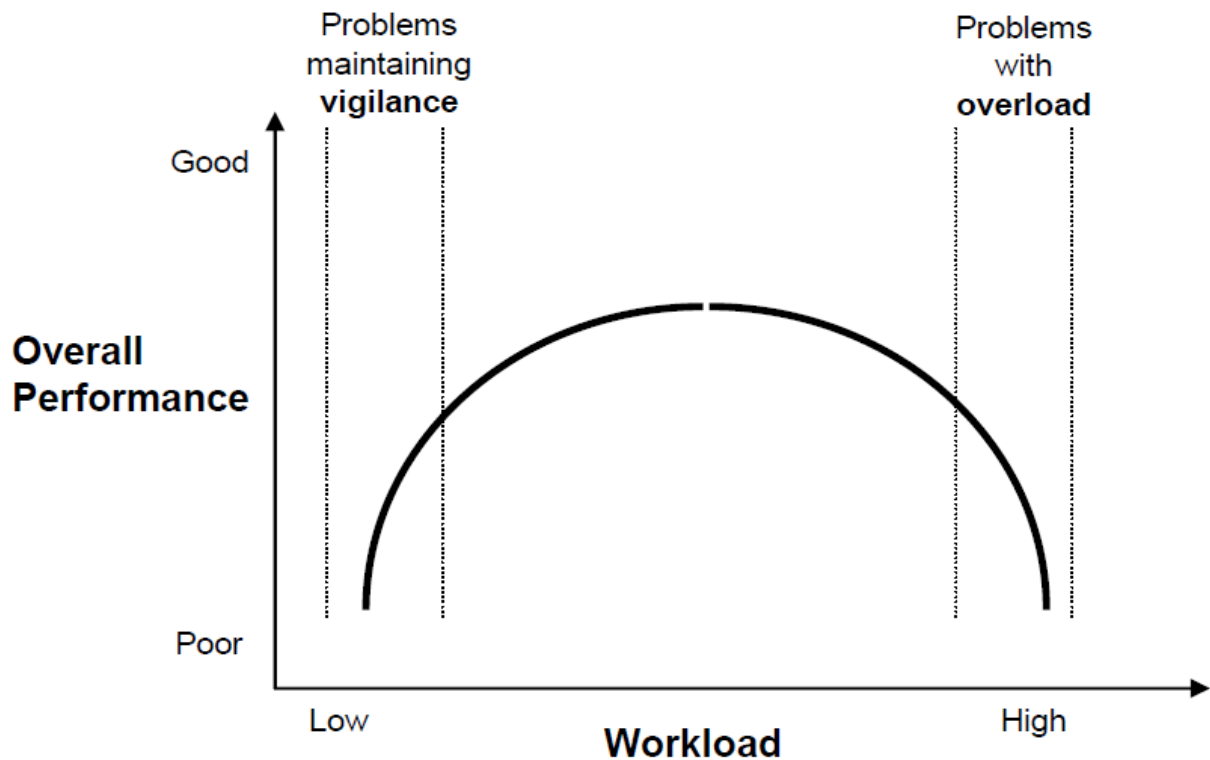


Figure 85 Dépendance entre la charge de travail et la performance du système [O'Hara 2010]

Comme on peut le constater en examinant la Figure 85, une charge de travail trop petite génère des problèmes venant de la vigilance que l'opérateur a sur le fonctionnement du système, donc les performances du système sont susceptibles de baisser. Au contraire, une charge de travail trop importante donnera une surcharge de l'opérateur, qui ne pourra plus analyser toutes les données qui lui seront communiquées. Ainsi, les performances du système sont à nouveau susceptibles de baisser. Une charge de travail moyenne résultera dans une bonne performance, l'opérateur étant capable d'ajuster le système pour qu'il garde des bonnes performances.

Une autre façon que la machine a pour décider quand il faut changer le niveau d'automatisation est d'utiliser des modèles de l'opérateur. En examinant la façon dont il interagit avec elle, la machine peut décider s'il a besoin d'assistance.

Une telle proposition d'architecture est faite par [Wood 2004]. Le schéma est fait pour des avions de chasse, mais les auteurs affirment qu'elle peut être utilisée avec n'importe quelle interface adaptative.

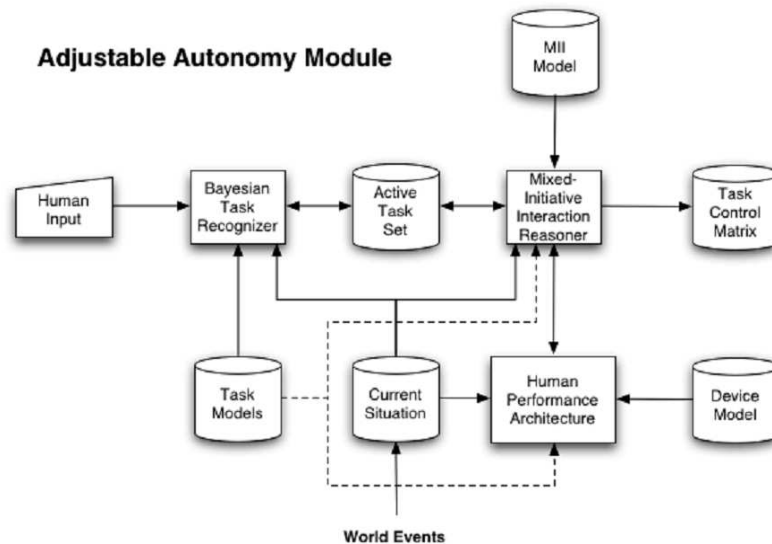


Figure 86 Module pour l'autonomie adaptative [Wood 2004]

Dans [Alpert 2003] l'attitude des opérateurs par rapport à une interface adaptative utilisée par un site de commerce électronique est examinée. Le modèle implicite de l'opérateur (comportement) est basé sur des visites précédentes sur le site en utilisant la même interface. Le modèle explicite est basé sur des suggestions directes de la part des opérateurs.

On peut supposer qu'un système qui examine l'opérateur en utilisant ses interactions (commandes) avec le système, mais aussi des mesures physiques, doit fournir de très bons résultats, malgré sa difficulté de mise en œuvre. [Duric 2002] proposent un tel système. Le point du regard, ainsi que les émotions exprimées par l'opérateur par l'intermédiaire du son visage et la partie supérieure de son corps sont utilisés. Les touches appuyées (choix et fréquence) ainsi que les mouvements et les clics de la souris sont examinés.

COGMON [Willis 2005] est un des systèmes combinés les plus complets. Les données captées fournissent des informations sur l'état d'alerte de l'opérateur et quelles inférences faire sur la nature de l'activité cognitive courante. Les saccades sont également utilisées par le système. Le pouls, la respiration, la conductivité et la température des extrémités sont analysés. Ces mesures donnent des informations sur le stress. Les facteurs environnementaux (bruits, lumière, température), ainsi que les vocalisations de l'opérateur sont enregistrés. Beaucoup de logiciels de statistiques et analytiques sont utilisés pour interpréter ces signaux. Ils doivent également effectuer cette analyse de façon convergente (plus d'une source de signaux). Ces logiciels peuvent aussi identifier des profils particuliers. Ces profils sont personnalisés en fonction de l'utilisateur et du type d'activité qui est effectuée, à cause du fait que l'interprétation des mesures varie d'un contexte à un autre et d'un opérateur à un autre.

Plusieurs systèmes pour changer le mode de commande ont été proposés dans la littérature. Dans ces systèmes, il est possible que l'opérateur humain soit à l'origine du changement, mais aussi que l'automatisation le fasse.

[Brunelis 2008] propose un schéma pour l'allocation dynamique des tâches. Ceci est fait dans un souci d'équilibrer la charge de travail (workload) de l'opérateur. Si on utilise longtemps de hauts niveaux d'automatisation du système, sans utilisation du mode de manuel, on aura une

dégradation de la conscience de la situation (situation awareness). Dans le temps, une perte des compétences et de l'expertise acquissent en utilisant des modes de commande moins automatisés est inévitable. Le schéma proposé est le suivant :

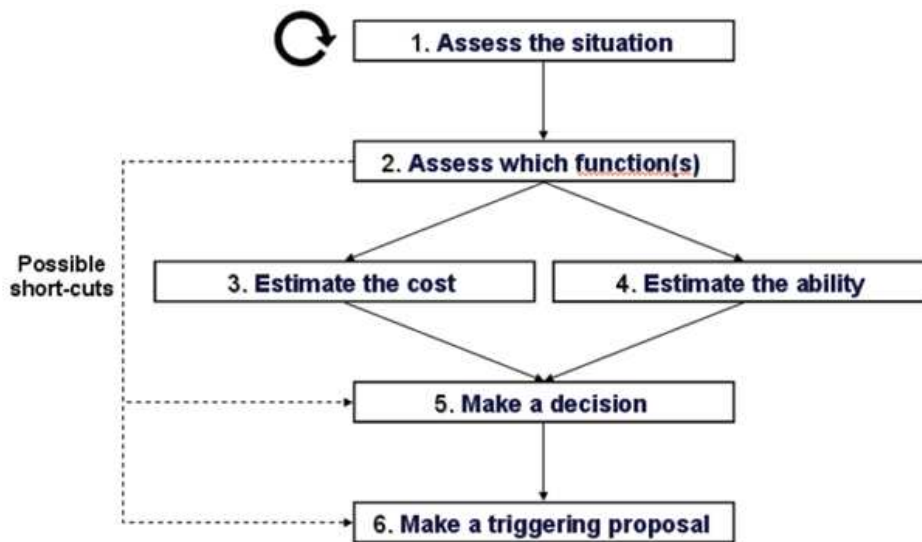


Figure 87 Changement de niveau d'automatisation [Brunelis 2008]

La machine évalue la situation, ensuite évalue les fonctions à affecter, estime les coûts d'affectation des fonctions, estime les capacités de l'opérateur pour effectuer l'activité, prend une décision concernant la proposition de réaffectation et fait une proposition de déclenchement (avec son timing) et déclenche la réaffectation de la fonction. Comme on peut l'observer, il est possible d'éviter certaines de ces étapes.

[Dongen 2005] propose l'architecture suivante (Figure 88) pour une allocation dynamique des tâches cognitives. L'analyse de l'information est abrégée par IA, et la prise de la décision est abrégée par DM. L'opérateur humain doit prendre le cycle extérieur de la figure, mais la machine peut aussi utiliser la boucle intérieure. La machine peut faire sa propre analyse de l'information et prendre la décision sans l'intervention de l'opérateur, pour une implémentation plus rapide de l'action.

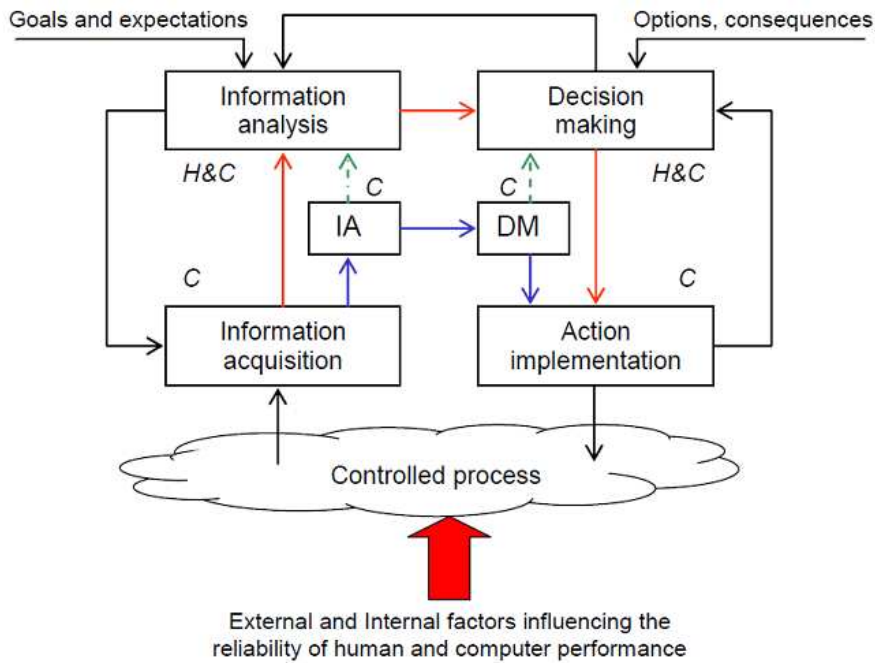


Figure 88 Allocation dynamique des tâches cognitives [Dongen 2005]

[Taylor 2003] compte utiliser son système pour le pilotage d'un avion de chasse. Dans ce contexte, il a besoin d'utiliser un moniteur cognitif (Figure 89).

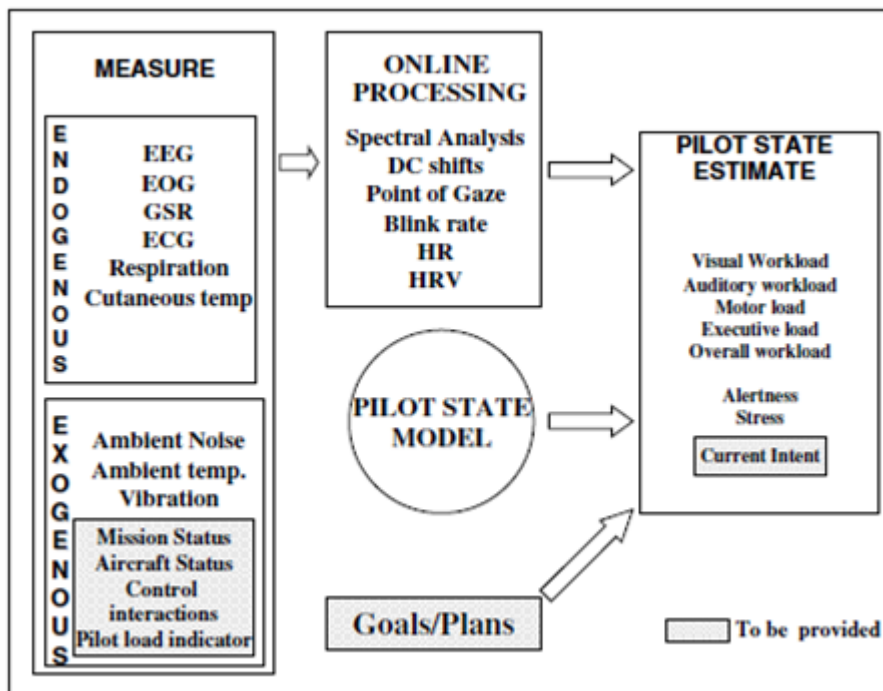


Figure 89 Moniteur cognitif

Ce moniteur utilise différentes mesures du pilote et de son environnement pour estimer son état. Le système a six états d'automatisation possibles, mais uniquement quatre sont utilisables dans le cadre de l'automatisation adaptative (Figure 90).

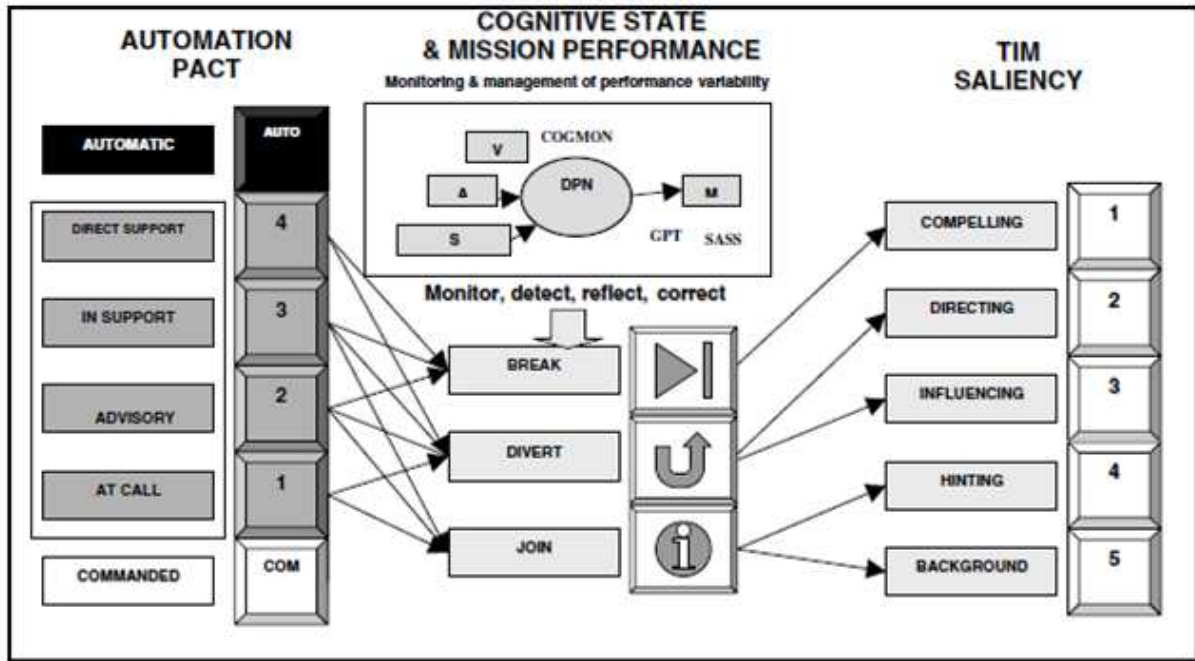


Figure 90 - PACT (Pilot Authorisation and Control of Tasks)

Une architecture générique pour les systèmes intelligents adaptatifs a été mise en avant par [Edwards 2004]. Elle est basée sur CommonKADS – Common Knowledge Acquisition and Documentation Structuring.

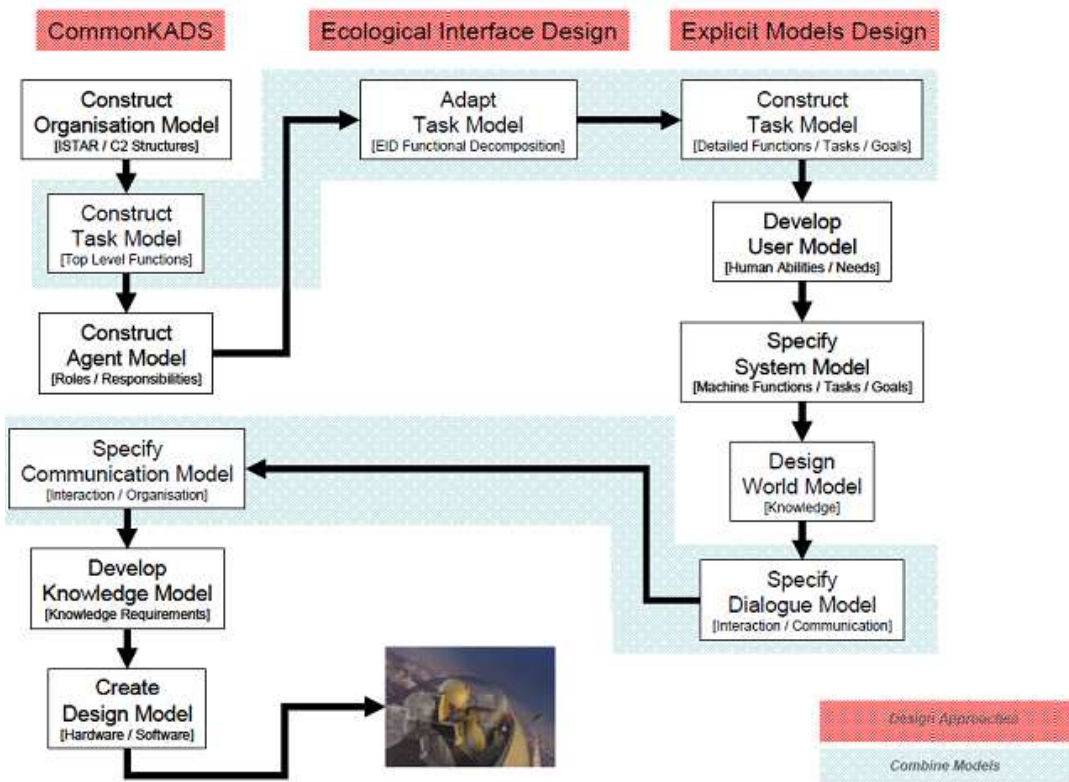


Figure 91 Architecture générique proposée par [Edwards 2004]

Le système utilisé par [Johnson 2007] doit réaliser des tâches différentes, comme le contrôle de l'armement, l'analyse des images, l'allocation des tâches (« va à », « protège », etc.) et

l'auto-routage des drones. Plusieurs façons pour l'adaptation sont investiguées, basées sur la charge de travail, les performances de l'opérateur et sur le temps. L'architecture du système utilisé est celle décrite dans la Figure 92.

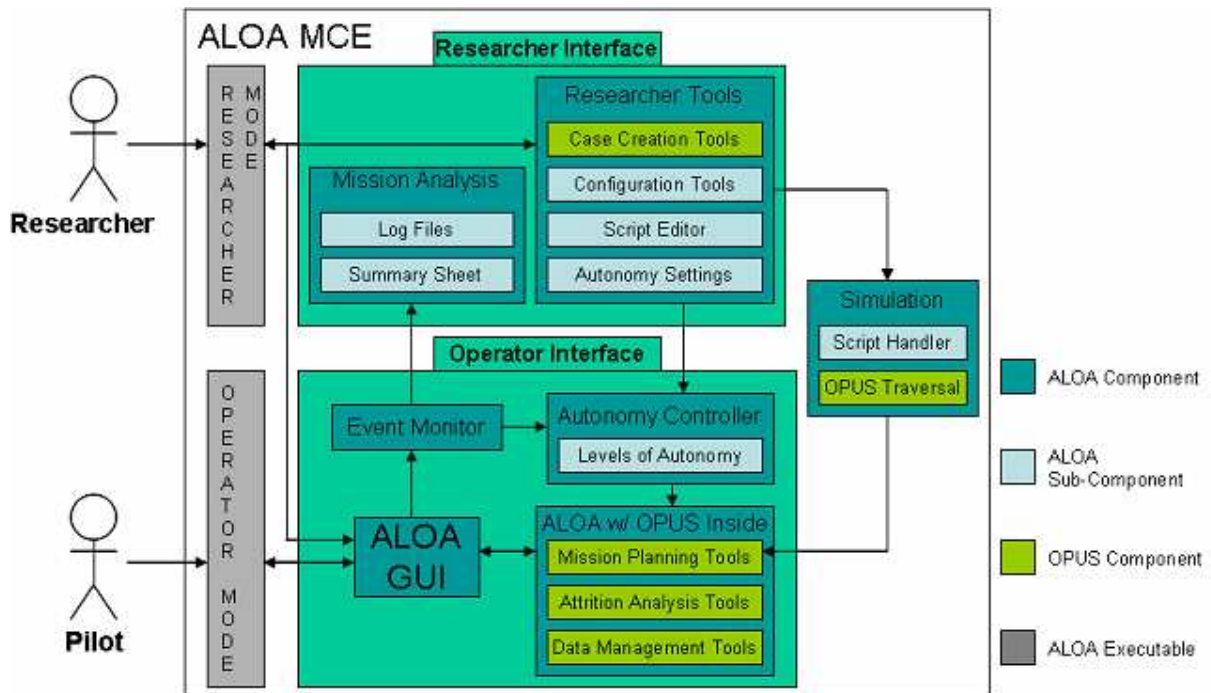


Figure 92 Architecture utilisée par [Johnson 2007]

2.3.3. Conséquences du changement du niveau d'automatisation

L'opérateur humain n'est pas capable de suivre les changements du mode opératoire du système de façon instantanée. On peut supposer que, si le changement est initié par lui alors il saura à l'avance comment gérer la nouvelle façon de fonctionner du système. Ceci sous-entend le fait que le changement du niveau d'automatisation n'est pas la conséquence d'une faute de l'opérateur (à cause d'une connaissance imparfaite des menus de l'interface de contrôle, par exemple) et que le schéma mental qu'il a sur le nouveau mode est bon. Ce schéma est construit par deux méthodes complémentaires : la documentation du mode d'opération (le manuel d'utilisation - sa lecture et compréhension) et l'entraînement avec le système, pour comprendre comment il réagit sous différentes conditions de fonctionnement.

Le changement du niveau d'automatisation initié par le système lui-même est plus difficile à suivre par l'opérateur. À l'exception des situations expérimentales, le changement est effectué uniquement quand le système estime que l'opérateur n'est plus capable de bien le gérer (ses performances deviennent inacceptables) et il prend donc la main. La charge de travail de l'opérateur est devenue trop importante et le système a décidé d'intervenir. Ainsi, l'opérateur aura une chance de récupérer l'écart entre l'état présent du système et sa représentation mentale sur cet état et d'interpréter toutes les informations pertinentes en temps réel. Une fois sa compréhension de la situation actualisée, il pourra utiliser le système dans le mode de commande précédent, s'il le décide.

Même dans le cas où l'opérateur se trouve à l'origine du changement du mode commande, il devra faire un effort cognitif pour s'adapter au nouveau mode. On peut parler ainsi du coût lié

au passage d'un mode de commande à un autre. Si les avantages du nouveau mode sont inférieurs à ce coût, alors, pour un certain temps après le changement, les performances du système seront inférieures aux performances antérieures. Si le système décide de faire le changement, il est probable que, pour un niveau inférieur d'automatisation, ce changement soit perturbant [Di Nocera 2007]. L'existence de ce coût est mise en évidence aussi par [Di Nocera 2005], où les systèmes d'une station spatiale doivent être gérés. Le changement de niveau n'est pas sous le contrôle de l'opérateur humain, donc il est raisonnable de supposer que, s'il l'était, ce coût serait inférieur, mais leurs conclusions restent toujours valables pour une automatisation adaptative.

On peut voir pourquoi il est souhaitable d'avoir une automatisation adaptable (initiée par l'opérateur) au lieu d'une automatisation adaptative (initiée par la machine). Bien sur, dans le cas des systèmes critiques il y aura toujours cette partie adaptative, pour les événements qui dépassent les capacités de l'opérateur humain (quand la charge de travail qui lui attribuée n'est plus raisonnable à accomplir avec les contraintes temporaires imposées). Ainsi, comme le trouve [Zieba 2009], un niveau moyen d'automatisation génère moins de changements qu'un niveau trop bas ou trop haut. [Millot 2008] trouve que, pour un système de contrôle aérien, les performances sont en corrélation directe avec le niveau d'automatisation (invocation d'une aide) employé. Deux types de coopération homme-machine sont aussi décrites : coopération verticale (l'agent supérieur peut invoquer l'aide de l'agent inférieur) et une coopération horizontale (ils sont indépendants si leur tâches le sont ; sinon, ils doivent soit s'entendre, soit faire appel à un arbitre/distributeur de tâches). [Hardin 2008] trouvent des résultats similaires, un mode où l'opérateur gère le comportement du système (composé par des agents) donne des performances meilleures qu'un système avec des agents complètement indépendants. Pour une commande à un niveau moyen [Endsley 1999] sont aussi d'accord. Ils trouvent (pour des tâches de type contrôle aérien) que les performances sont meilleures et que la compréhension de la situation (situation awareness) est plus facile pour un niveau moyen que pour un niveau avancé d'automatisation.

2.3.4. Conséquences d'une automatisation imparfaite

Il y a trois principales raisons pour lesquelles l'automatisation d'un système peut ne pas bien fonctionner. La première est liée à une mauvaise implémentation par rapport aux spécifications. La deuxième raison est liée aux défaillances des capteurs (ou des systèmes se trouvant en amont). Même si l'automatisation a été correctement implémentée, si les données qui lui sont fournies ne sont pas bonnes, les décisions prises seront, pour un observateur externe, mauvaises. La troisième raison pour laquelle l'automatisation peut ne pas bien fonctionner est sa mauvaise utilisation par l'opérateur humain. Si le contexte n'est pas bon, l'utilisation d'un certain niveau d'automatisation, qui n'est pas adapté à la situation, peut avoir des conséquences graves, comme pour le vol 965 American Airlines, où le mode de commande activé n'était pas le même que celui que les pilotes pensaient être activé. Ainsi, à cause d'une mauvaise interprétation de la part des pilotes d'une information présentée par le système, qui lui a été ensuite envoyée comme consigne, en plus de l'utilisation d'un mode de commande inadéquate, la tragédie n'a pas pu être évitée.

Une première conséquence est le fait que les performances du système vont se dégrader [Meyer 2003]. Dans ces situations, il est possible que l'utilisation d'un niveau inférieur d'automatisation améliore les performances du système automatisé. Si l'opérateur est informé ou s'il se rend compte du fait que l'automatisation du système n'est pas parfaite (à cause des deux premières raisons), il peut perdre sa confiance dans le système. Ceci a été observé par [Cassidy 2009], par exemple. Dans ce cas, les opérateurs ont eu tendance à ignorer les suggestions du système.

Si l'opérateur ne se rend pas compte du fait que l'automatisation ne fonctionne pas parfaitement, il est possible que les performances du système baissent. La situation devient plus floue dans le cas d'une automatisation adaptative (quand c'est le système qui décide de changer de niveau d'automatisation). [Moray 2000] trouve qu'il est préférable d'avoir une automatisation imparfaite dans des situations contraintes temporellement (qu'il faut traiter rapidement). Pour leur cas, le manque de fiabilité de l'automatisation n'affecte pas de façon notable les performances du système, si elle est de 90%. Si la fiabilité est de 70%, alors un mode plus bas d'automatisation donne des performances meilleures.

2.4. Le cas du système CompanionAble

Comme on l'a vu, l'automatisation du système est appliquée à beaucoup de domaines. Par rapport à la surveillance d'une centrale nucléaire ou le contrôle et la coordination d'un essaim de drones, notre cas est assez simple. Une situation typique d'utilisation de notre système est décrite ci dessous. Nous sommes dans le cas d'une personne âgée, habitant seule dans un appartement. Les modes de commandes disponibles sur le robot sont : le mode manuel, le mode caméra, le mode CnG et un fonctionnement automatique. Comme c'est le cas pour un fonctionnement normal de notre système, le robot fonctionne en mode autonome. La personne est de train de regarder la télévision. L'heure du dîner arrive et la personne se met debout, pour aller dans la cuisine pour le préparer. Soudainement, on détecte, par l'intermédiaire d'un petit holter³ muni, en plus, de capteurs de mouvement, que la personne est tombée. Le robot, fonctionnant en mode automatique, la cherche, pour vérifier son état. A cause du fait que la personne est tombée derrière le canapé, le robot n'arrive pas à la retrouver. Une alarme est alors envoyée et une personne d'un centre de télévigilance prend le contrôle du robot. Elle pilote le robot en mode manuel et cherche la personne. Elle arrive à la trouver et vérifie l'état de la personne. Après s'être assurée que la personne peut se lever seule, la session de téléopération est finie et le robot retourne en mode automatique.

Comme on peut le voir dans cet exemple de scénario, notre système présente quelques particularités. Le temps typique d'une session de téléopération, par exemple. On ne s'attend pas qu'il dépasse quinze minutes. Ainsi, on peut supposer que la fatigue de l'opérateur pendant la session augmentera de façon négligeable. Les utilisateurs du système ne feront pas ça de façon régulière. Ils auront des entraînements avec le système, mais ils ne sauront pas comment il fonctionne de façon précise. Enfin, la quantité d'informations que l'opérateur aura à traiter sera raisonnable. Ceci ne signifie pas qu'il ne faut pas l'aider, avec des

³ Dispositif portable permettant l'enregistrement en continu du pouls et de la tension artérielle pendant au moins 24 h.

représentations unifiées des informations, par exemple. Mais on ne s'attend pas que l'évolution du système soit trop rapide pour qu'elle entraîne une surcharge cognitive de l'opérateur.

2.5. Proposition

Le changement de mode de commande est une question qui a été abordée pour aider l'opérateur (côté adaptable) ou pour maintenir les performances de systèmes acceptables (côté adaptatif). La façon dont la commande est envoyée par la partie humaine du système à la machine reste, fondamentalement, inchangée. Il s'agit des commandes discrètes (une commande est suffisante pour un but). Elles sont envoyées une à une à la machine. D'un point de vue pratique, il faut noter que dans ce contexte, discret signifie à une fréquence inférieure à quelques Hz (cliquer sur des boutons, par exemple). En contraste avec ceci, les commandes envoyées en continu sont envoyées avec une fréquence supérieure à quelques Hz, afin de surmonter le seuil de perception imposé par l'opérateur. Ce qu'on propose de faire est d'aider l'opérateur dans sa tâche de téléopération, afin qu'il soit le plus à l'aise possible pendant qu'il contrôle le robot. Si possible, on veut lui cacher le fait qu'un changement a eu lieu. Ceci implique l'utilisation d'une automatisation adaptative du système.

Le mode manuel est un mode continu. L'opérateur est impliqué à tout moment dans la conduite du robot. Ceci ne veut pas dire que le robot ne peut pas se trouver dans un état d'automatisation supérieur à un (pas de tout automatisé). Mais, le fait est que si l'opérateur n'indique pas au robot de se déplacer, celui-ci ne se déplace pas. En utilisant la classification proposée par [Endsley 1995], sa conscience de la situation (situation awareness – SA) peut se limiter au deuxième niveau (intégration et compréhension des éléments de l'environnement). Ceci ne gêne pas l'opérateur pour acquérir une conscience de la situation de troisième niveau.

Le mode de planification en environnement inconnu (CnG) est un mode discret. Pendant que le robot se déplace, l'opérateur peut ne pas se focaliser sur son évolution dans l'environnement, comme dans le cas précédent. La conscience de la situation (CS) de l'opérateur doit, obligatoirement, se situer au troisième niveau (projection des états futurs de l'environnement, basée sur la compréhension [Endsley 1995]).

Comme on l'a vu, on a deux sources principales d'information pour aider l'automatisation adaptative à prendre une décision en ce qui concerne l'opérateur : des informations concernant sa charge de travail (workload) et des informations concernant sa performance en utilisant le système.

La charge de travail (CT) de l'opérateur est évaluée principalement en examinant celui-ci pendant l'utilisation du système. Habituellement, les mesures proviennent des capteurs comme un EEG (électroencéphalographe), un oculomètre ou un ECG (électrocardiographe). A l'exception de l'oculomètre, les autres capteurs sont assez intrusifs. Comme on envisage l'utilisation du système par un particulier, même l'utilisation d'un oculomètre est hors de question. Ceci est dû au fait que on souhait garder le système le plus générique que possible, pour des raisons de facilités d'utilisation (sans personnel surveillant qualifié dans l'installation et l'utilisation de tels systèmes) et pour maintenir les coûts raisonnables.

Une autre façon de déterminer la charge de travail supportée par l'opérateur est d'examiner les informations que celui-ci aura à traiter. Si le système estime que leur volume est trop important, l'automatisation sera invoquée. Étant donné que l'utilisateur typique du notre système n'aura pas des capteurs dédiés à la mesure de sa charge de travail, on pourra se baser uniquement sur l'analyse des informations que celui-ci aura à traiter. Ceci est à dire que, pour notre système, on peut se servir des observations sur les commandes envoyées par l'opérateur et des observations sur l'environnement. Les modes de commande mis à la disposition de l'opérateur sont regroupés dans deux catégories : manuelle et semi-automatique.

On suppose que l'opérateur a des bonnes raisons pour le choix d'un certain mode de commande. C'est à dire que l'automatisation adaptative essayera de choisir un nouveau mode, le plus proche de celui choisi par l'opérateur que possible, afin que celui-ci soit le moins troublé que possible dans la réalisation de sa tâche. La proximité entre deux modes est d'abord reliée au mode de contact entre l'opérateur et le système. Si l'opérateur utilise des touches, l'automatisation doit proposer un mode qui utilise des touches aussi (si possible). Si les touches doivent être maintenu appuyées, l'automatisation doit proposer un mode qui utilise des touches qui doivent être maintenu appuyées aussi, et ainsi de suite. La proximité entre deux modes est aussi reliée à la CS demandée. Si l'opérateur a une CS inférieure à celle requise par le nouveau mode, la transition ne sera pas nette, l'opérateur devant faire un effort supplémentaire pour arriver à la CS nécessaire pour bien utiliser le nouveau mode.

Une critique de la notion de CS a été faite par [Dekker 2010]. Un autre problème est qu'elle ne peut pas être évaluée en ligne, des questionnaires pour l'établir étant utilisés [Prinzel 2003], lesquels sont remplis pendant que le système est en pause (ou à la fin de l'utilisation du système).

Dans la catégorie manuelle on a deux modes de commande (le mode manuel et le mode caméra) et une aide contre les délais (la modification des images). Pour des raisons de sécurité, l'évitement d'obstacles est actif tout le temps. Si le système détecte qu'un délai nuisible est en train d'intervenir dans la liaison opérateur-robot, la modification de l'image est activée, pour aider les deux premières niveaux de la CS (perception des éléments de l'environnement et leur intégration et compréhension).

En ce qui concerne les performances du système, si le mode de commande est manuel et que le système constate que l'opérateur fait trop d'ajustements sur l'orientation du robot, il passera dans le mode caméra, pour lui faciliter le contrôle. Ceci sera décidé en analysant les rayons de courbure du robot. Les expériences que nous avons menées vont dans cette direction. Un problème reste quand même : le fait qu'un rayon de courbure optimal pour un environnement (maison) peut ne pas l'être pour un autre (entrepôt), à cause d'une architecture et d'une disposition des objets différente. Mais, pour des environnements typiques (habitations), il est possible de déterminer une bonne valeur initiale.

Ces deux types de changement ont la qualité de ne pas avoir besoin d'une pause dans l'envoi des commandes de l'opérateur au robot.

Le passage d'une catégorie de mode de commande (manuelle, semi-automatique, automatique) à une autre est plus compliqué. Si l'opérateur se trouve à l'origine de ce

changement, alors le robot sera à l'arrêt (l'opérateur utilise la souris pour indiquer le changement de mode, donc il ne peut pas l'utiliser en parallèle pour piloter le robot).

En ce qui concerne le changement automatique, pour un délai au-delà des possibilités de la modification des images, le mode semi-automatique sera utilisé, pour que la performance du système soit affectée dans un degré minimal. La CS demandé à l'opérateur est supérieure dans ce cas de mode de commande, par rapport à un mode continu (plus de planification est nécessaire, qui implique une analyse sur une période plus longue de temps des états futurs du robot par l'opérateur). Le système arrête le robot, informe l'opérateur du changement qui suivra et change de mode de commande. Ainsi, pendant cet arrêt, l'opérateur aura l'opportunité de se former une CS suffisante pour bien utiliser le nouveau mode. La Figure 93 résume ceci.

Le passage automatique du mode manuel au mode caméra (d'un mode de commande continu à un autre) se fait sans arrêter le robot. De même, l'aide pour la téléopération avec délai est activée sans être obligé d'arrêter le robot.

En ce qui concerne le passage d'un mode continu à un mode discret, il est réalisé de façon automatique si le délai est devenu trop important pour être traité avec une aide pour un mode continu. Le robot est arrêté, pour les raisons exposées au début de ce paragraphe. Pour revenir dans un mode de commande discret, il faut que l'opérateur le demande. Si le robot est à l'arrêt, il change simplement de mode. Sinon, on propose que l'opérateur arrive à lui donner une consigne similaire à sa propre commande (reprise), en lui affichant l'objectif courant du robot et ses vitesses. Si le délai est trop grand même pour un mode discret de commande, le robot passe en mode automatique. Il doit arriver à son objectif avant de faire ce changement.

Pour le passage du mode automatique vers un mode continu, il faut que le robot soit à l'arrêt. Il serait envisageable de traiter ce changement (automatique - continu) comme un changement discret-continu, mais le robot peut avoir des objectifs variés : « aller dans la cuisine » est envisageable à être représenté sur l'interface courante. « Chercher les lunettes » nécessite une nouvelle réflexion sur l'interface pour être représenté au téléopérateur.

Le passage du mode automatique vers un mode discret n'a pas absolument besoin que le robot soit à l'arrêt. Si le robot est en mouvement (avec les mêmes difficultés de représentation de son objectif comme pour un passage automatique-continu), la réception d'une destination à atteindre fera qu'il s'y dirigera sans s'arrêter (il est possible que sa vitesse varie, en fonction des virages qu'il doit faire). Deux choses font que ceci est possible. D'abord, l'évitement d'obstacles est exécuté en parallèle avec la planification de la trajectoire, donc le robot peut suivre la consigne du mode précédent, en attendant qu'une trajectoire soit calculée, sans rentrer en collision avec l'environnement. Ensuite, le calcul d'une trajectoire est rapide, donc le temps de passage d'un mode à l'autre ne sera pas long. Pendant les évaluations du mode Click'n'Go avec la nouvelle interface écologique, on a eu des opérateurs qui envoyaient un nouveau ordre pendant que le robot était en mouvement. Ils attendaient que le robot s'approche de sa destination et, avant qu'elle soit atteinte, ils envoyaient une nouvelle, en continuation. Pendant ces évaluations, le robot ne s'est jamais arrêté pour calculer une nouvelle trajectoire.

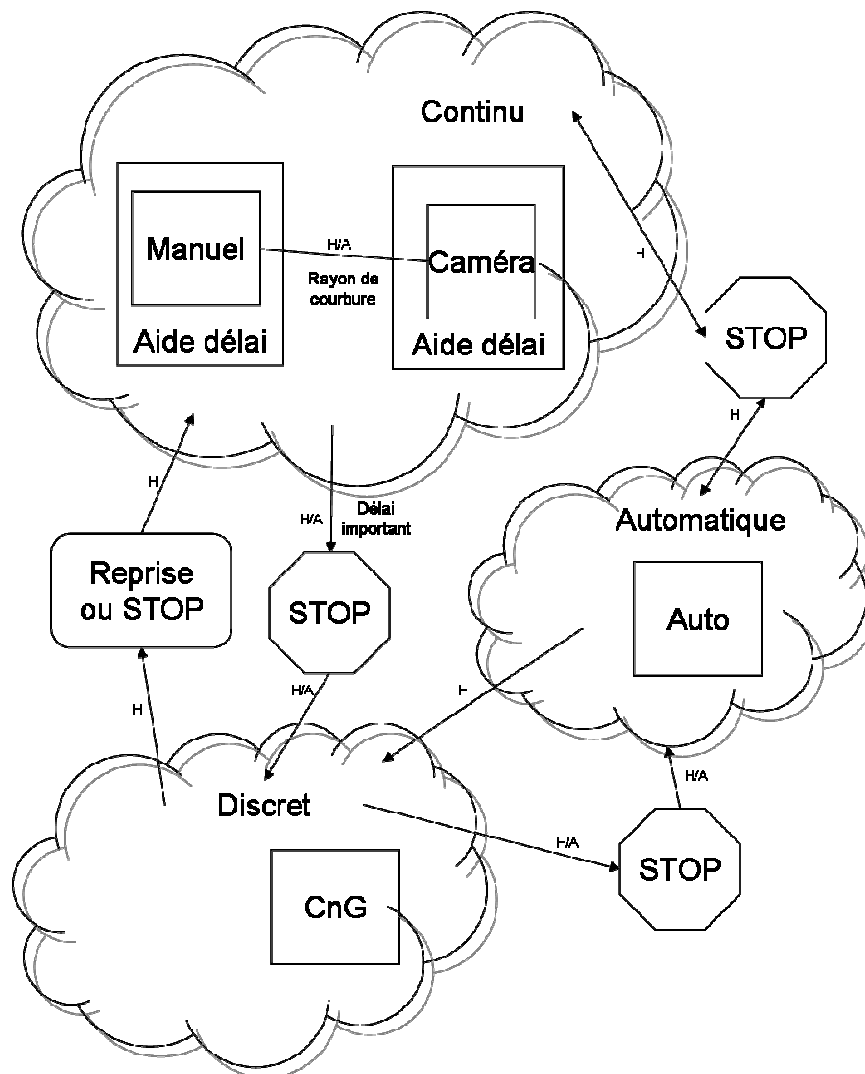


Figure 93 Changement de mode de commande

2.6. Perspectives

La mise en place des évaluations permettra de vérifier l'efficacité de l'aide apporté par chaque type de changement. Les résultats seront utilisés pour raffiner certains changements, afin que le retour des opérateurs soit intégré dans la conception du système.

Cette partie a présenté une première réflexion sur les changements de mode de commande pour notre système de téléopération. Cette réflexion a été centrée sur l'opérateur humain. La suite devrait se focaliser sur la possibilité que le robot se retrouve dans l'incapacité de fonctionner à 100% de ses capacités ou dans une situation imprévue, qu'il ne saurait pas gérer.

Pour une situation imprévue, une première piste serait de faire le choix d'un mode de commande nécessitant une CS minimale pour aider le robot, pour que l'opérateur puisse tout de suite la réaliser.

Si le robot peut faire un auto-diagnostic, une solution serait de choisir un mode de commande qui n'a pas absolument besoin de la fonctionnalité défaillante pour fonctionner et qui aurait besoin d'une CS élevé, afin que l'opérateur ne pilote pas comme il a l'habitude, mais qu'il soit contraint de réfléchir aux états futurs du système.

L'utilisation des capteurs pour des mesures physiques sur l'opérateur pourrait aider à améliorer l'évaluation de sa CT, malgré l'encombrement et spécificité du système de commande demandés par ceux-ci.

Conclusions et perspectives

Ce travail de thèse a eu pour but de faciliter la téléopération d'un robot, dans un contexte spécifique - le maintien à domicile des personnes âgées, atteintes de troubles cognitifs légers. De plus, ces personnes habiteront seules. Le robot, en collaboration avec la maison (munie de capteurs) de la personne âgée, veillera à son bien-être et pourra être utilisé pour communiquer avec elle. Le robot doit aider et accompagner cette personne dans sa vie quotidienne. On espère que la personne âgée finira par considérer le robot comme un compagnon (comme un chien ou un chat), ce qui fera qu'elle soit plus impliquée, parmi d'autres activités, dans la réalisation des exercices de stimulation cognitive (plus d'attention, moins d'abandons), utiles pour retarder la détérioration des capacités cognitives. Le robot peut fonctionner de façon autonome, ainsi qu'être téléopéré. Pour des raisons de facilité d'installation du système d'aide (maison intelligente et robot), l'Internet public est utilisé comme milieu de communication. Avec ce type de liaison, on s'attend que la qualité de la connexion ne permette pas toujours d'avoir un délai de communication faible (300 ms pour notre cas concret). La première chose qu'on a essayée de faire a été de réduire l'influence de ce délai sur l'opérateur. L'idée sur laquelle l'aide que nous avons proposée dans le premier chapitre de cette thèse se base est de modifier l'image, afin qu'elle se rapproche le plus possible de l'image que l'opérateur verrait sur son interface de commande si le délai était extrêmement faible, voire nul. Deux séries de tests ont été menées pour évaluer son utilité. La première série a été réalisée avec des volontaires exerçant dans le cadre du laboratoire IBISC. Pour la deuxième série d'évaluations, on s'est déplacé en Espagne, à l'université de la Corogne, dans le cadre du projet européen CompanionAble. Là-bas, les sujets qui ont participé aux évaluations étaient des professionnels de la santé, plus exactement de la gériatrie. On s'attend que des tels professionnels utilisent le système CompanionAble, quand celui-ci sera opérationnel. Les résultats obtenus montrent que, en employant l'aide proposée, le temps de parcours est comparable avec celui obtenu dans des conditions normales (sans délai) et que le nombre d'arrêts du robot (« move-and-wait ») est réduit.

La suite des travaux a été d'examiner d'autres modes de commande du robot. Le mode manuel, même avec une aide qui le rend plus résistant face aux perturbations induites par les délais, est un mode qui implique beaucoup l'opérateur dans son utilisation. Celui-ci doit être impliqué de façon continue dans le contrôle du robot : il faut qu'il analyse son évolution et la corrige, si besoin, en permanence, pour que le robot se dirige vers l'endroit souhaité par l'opérateur. A part ce mode, le robot peut fonctionner en mode automatique. Le fait que l'opérateur doive être beaucoup impliqué dans son opération, ainsi que le fait que l'autre mode d'évolution du robot (autonome) soit à l'autre bout du spectre s'étalant entre un mode de fonctionnement purement manuel et un mode purement autonome, font qu'on a mené une réflexion sur un nouveau mode de commande qui nécessite, à la fois, l'implication de l'opérateur dans un moindre degré que le mode manuel et qui se dirige vers une utilisation plus importante des capacités d'autonomie du robot qu'un simple évitement d'obstacles, par exemple. Ce mode constitue le sujet du deuxième chapitre de cette thèse. Une fois la destination envoyée au robot, celui-ci doit utiliser ses propres capacités pour la joindre, même si le chemin est compliqué et nécessite des retours en arrière. Pendant ce temps, l'opérateur peut poursuivre d'autres activités, comme parler avec la personne âgée ou observer

l'environnement distant. Des tests effectués en simulation et avec un robot réel ont prouvé que ce nouveau mode de commande est capable, avec une seule consigne (les coordonnées de la destination, envoyées par l'opérateur), d'effectuer la planification et le suivi de la trajectoire. Cette trajectoire, déterminée à l'avance, peut comporter des zigzags. Le suivi de trajectoire les lisse, afin d'obtenir une trajectoire plus directe. Quand il a été confronté à des situations plus complexes (des impasses), le robot a pu retrouver tout seul le bon chemin vers son objectif.

Une fois qu'on a augmenté le choix de l'opérateur en ce qui concerne la façon de piloter le robot, il fallait réunir ceci dans un cadre commun. La troisième partie de cette thèse vient relier les chapitres précédents. Dans une première partie, une interface commune aux différents modes de commande est présentée. Cette interface augmente le retour vidéo, en utilisant les informations issues du capteur laser. Ces données laser sont mises en correspondance avec les objets visibles sur l'image et continuent à être affichées au-delà de l'ouverture horizontale de la caméra. A titre d'exemple, pour le robot Lina, l'ouverture horizontale de la caméra est de 60°, pendant que le laser a un champ de 240°, six fois plus que la caméra. Des évaluations de cette interface avec différents modes de commande, et par rapport à celle qui était utilisée auparavant avec le mode de commande manuel, ont été réalisées. On a une interface qui permet l'utilisation de plusieurs modes de commande. Le mode WP et le mode CnG ont été testés avec la même interface, mais pas le passage de l'un à l'autre, ce qui dépassait le cadre qu'on s'était fixé pour cette série d'évaluations. Par rapport à l'interface précédemment utilisée, on obtient des temps de parcours meilleurs, moins de collisions avec l'environnement et une trajectoire moins courbée (moins d'ajustements). Une évaluation subjective de la part des utilisateurs a montré également leur préférence pour la nouvelle interface. La même interface a été aussi utilisée avec le mode de commande semi-automatique. Dans la suite de la troisième partie, la question du passage d'un mode de commande à un autre est examinée. On a proposé des façons de faire ce passage pour que pour l'opérateur soit assisté dans sa tâche de téléopération, mais aussi pour que les performances du système restent bonnes. On essaye de respecter le choix de l'opérateur en ce qui concerne le type de commande choisi. S'il souhaite utiliser un mode continu, on privilégie le passage (initié par le système) vers un autre mode de commande de type continu. Le passage d'un type à un autre n'est fait que si les conditions empêchent le bon fonctionnement de tous les modes d'une même catégorie. En ce qui concerne le passage initié par l'opérateur, on veut s'assurer qu'il comprenne l'état du robot. Soit le robot n'est pas en mouvement, auquel cas on n'impose pas de restrictions, soit le robot est en mouvement et il faut que l'opérateur montre qu'il a compris quelles étaient les objectifs du robot.

On compte réaliser des tests de la nouvelle interface à la fin de l'année 2011, en collaboration avec des partenaires européens, dans le cadre du projet CompanionAble. Les opérateurs seront des personnes impliquées dans le domaine de la télémédecine. Plus tard, on souhaite installer le système complet (le robot et la maison intelligente) et tester l'utilisation des différents modes de commande disponibles, dans des situations réelles (l'opérateur et le robot seront situés à des distances de l'ordre de centaines de kilomètres l'un de l'autre). Cette validation dans des conditions réelles nous permettra de raffiner le changement de mode de commande, ainsi que de commencer à réfléchir à l'utilisation de l'information ambiante (fournie par la maison) dans la téléopération du robot.

Une bonne connaissance du délai entre le site de l'opérateur et celui du robot est nécessaire pour que l'image résultante à la suite de la modification en utilisant l'aide proposée soit la meilleure correspondance possible. Si la connaissance de la valeur du délai ou du déplacement du robot réel n'est pas exacte (mais qu'elle reste raisonnable), les opérateurs peuvent toujours profiter de l'aide, comme le montre les expériences menées avec l'évitement d'obstacles activé. En pratique, si on a à faire à un délai constant, il est possible de le mesurer au début de la session de téléopération. L'étude bibliographique avait montré que la partie constante était beaucoup plus présente (le délai total avait cette valeur la majorité du temps) que la partie variable, et que celle dernière avait des valeurs plus importantes (Figure 3). Pour que le système de téléopération soit le plus générale possible, il faudrait traiter cette variation de façon explicite et l'intégrer dans l'aide à la téléopération en présence des délais proposés. Pour cela, des techniques de prédiction pourraient être utilisées. Le schéma proposée dans la Figure 19 resterait le même, la prédiction impacterait uniquement la valeur du délai, sans conséquence pour le reste du système. Pour mieux connaître les spécificités du système de maintien à domicile des personnes âgées, il serait souhaitable que des statistiques sur les délais soient réalisées. Ceci implique une campagne de mesures importante, dans plusieurs pays européens. Une autre amélioration à faire est d'utiliser la qualité de service mise à disposition par l'utilisation de l'IP v6. En ce qui concerne l'architecture du système d'aide, l'inclusion des commandes générées par l'évitement d'obstacles dans la modification du flux vidéo est souhaitable. En pratique, le robot aura l'évitement d'obstacle activé par défaut, pour des raisons de sécurité et de responsabilité. Pour des faibles interventions de l'évitement d'obstacles, dans la structure actuelle du système, l'opérateur n'est pas trop gêné, comme l'a montré la deuxième campagne d'évaluations de cette aide. Dans un environnement beaucoup plus encombré, par contre, on s'attend à ce que l'intervention de l'évitement d'obstacle soit plus perturbatrice pour l'opérateur, à cause de son intensité. Pour ces raisons, il est souhaitable de prendre en compte l'évitement (réalisé sur le robot) sur l'interface de pilotage. Ceci signifie qu'il faudra prédire son résultat sur le robot virtuel. Il est possible d'envisager l'utilisation d'une autre piste pour cette aide, afin de contourner l'utilisation de la valeur du délai dans la modification des images. Dans la version courante, la dernière image reçue par l'interface homme-robot est modifiée en fonction du délai. Le robot virtuel reçoit et exécute les mêmes ordres que le robot réel. Avant d'afficher l'image, on regarde de combien le robot virtuel (et donc le réel) a avancé pendant l'intervalle de temps donné par le délai et on effectue, le plus souvent, un zoom sur l'image, afin de prédire une image ressemblante à celle qui sera reçue par le poste de l'opérateur, mais après le délai utilisé pour la modification. Une autre façon d'obtenir l'indication sur la distance parcourue par le robot réel serait que celui-ci envoie sa position avec l'image. Ceci éviterait le besoin de connaître le délai entre le site maître et le site esclave. Ainsi, en faisant la différence entre la position courante du robot virtuel et celle du robot réel, l'image sera modifiée pour cacher les délais. Le principal défi de cette approche consiste dans la synchronisation de la position donnée par l'odométrie du robot virtuel avec celle du robot réel. Ceci peut se faire chaque fois que le robot s'arrête pendant une durée supérieure au délai de transmission.

En ce qui concerne le mode de commande décrit dans le deuxième chapitre de cette thèse, la planification dans un environnement inconnu se base beaucoup sur le plan fourni par le

SLAM. La technique qu'on a choisie pour la première implémentation est basée sur les mesures de distance fournies par un capteur laser. Une amélioration qui est en cours d'investigation dans le sein de notre équipe est l'utilisation d'un capteur Kinect, produit par Microsoft pour la console Xbox 360. Celui-ci permet d'avoir des informations de distance à l'intérieur d'un angle solide. Dans l'architecture proposée pour ce mode de commande, il est possible de remplacer l'algorithme de SLAM utilisé par un autre, sans avoir besoin de changer le reste du système (à condition que le plan construit soit de même type que celui fourni par l'ancien algorithme). Il serait intéressant que, dans le temps, des comparaisons entre différents algorithmes, qui utilisent des capteurs différents, soient faites, pour trouver le meilleur adapté à notre environnement-type. Comme la puissance de calcul qu'il est possible d'embarquer sur un robot est en constante croissance, il serait intéressant d'évaluer les algorithmes multi-capteurs, pour exploiter leur complémentarité. Un problème logistique particulier réside dans les évaluations qui devraient être effectuées. Il faut utiliser des environnements le plus proches de ceux réels que possible. Le plus simple serait d'installer le système dans différentes habitations, ce qui est très coûteux de point de vue du temps nécessaire. A court terme (2013), la Maison de l'Innovation, des Sciences et de la Société (MISS), résultat d'une initiative commune entre des différents organismes de recherche et industriels dont l'Université d'Evry fait partie, devrait fournir un environnement adapté à ce type de campagne d'évaluations.

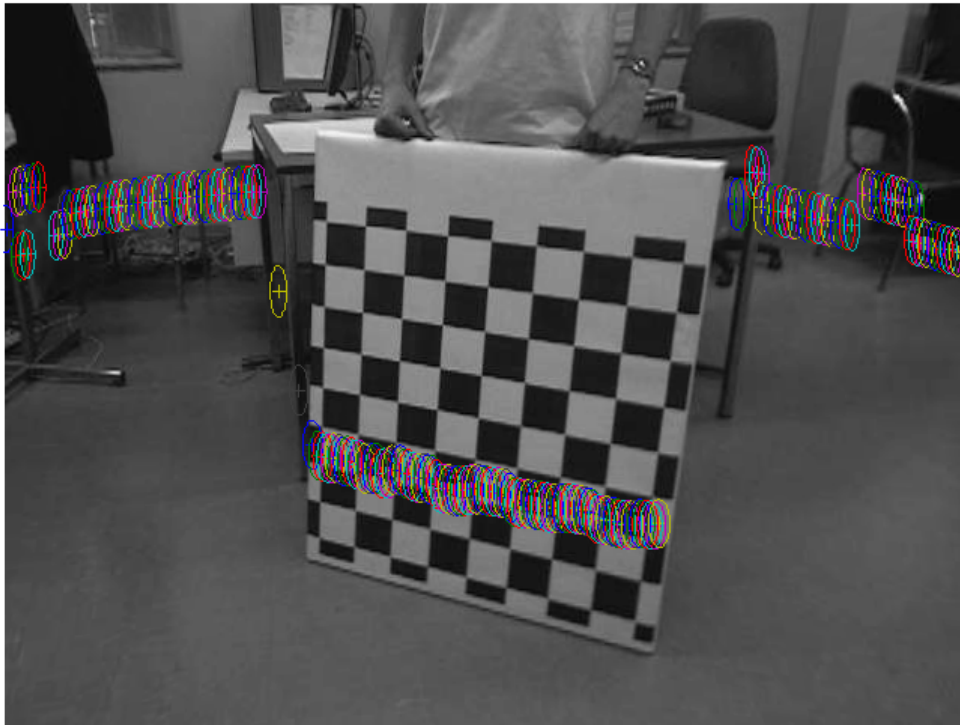
Des évaluations du changement de mode de commande sont à prévoir. Une amélioration du système proposé serait d'utiliser des capteurs physiques pour évaluer l'état de la personne. A titre d'exemple, les oculomètres fabriqués par la compagnie suédoise Tobii Technology sont très répandus. Ainsi, il serait possible d'évaluer, en plus de l'état de l'opérateur, les zones qu'il est en train de regarder sur l'interface. Il est possible que d'autres catégories de commande soient définies et incluses dans le modèle proposé, mais ceci nécessiterait l'implémentation de nouveaux modes de commande sur l'interface et sur le robot. Si on se rendait compte que l'interface pose des limites à l'utilisation de ces nouveaux modes, sa mise à jour devrait être entreprise.

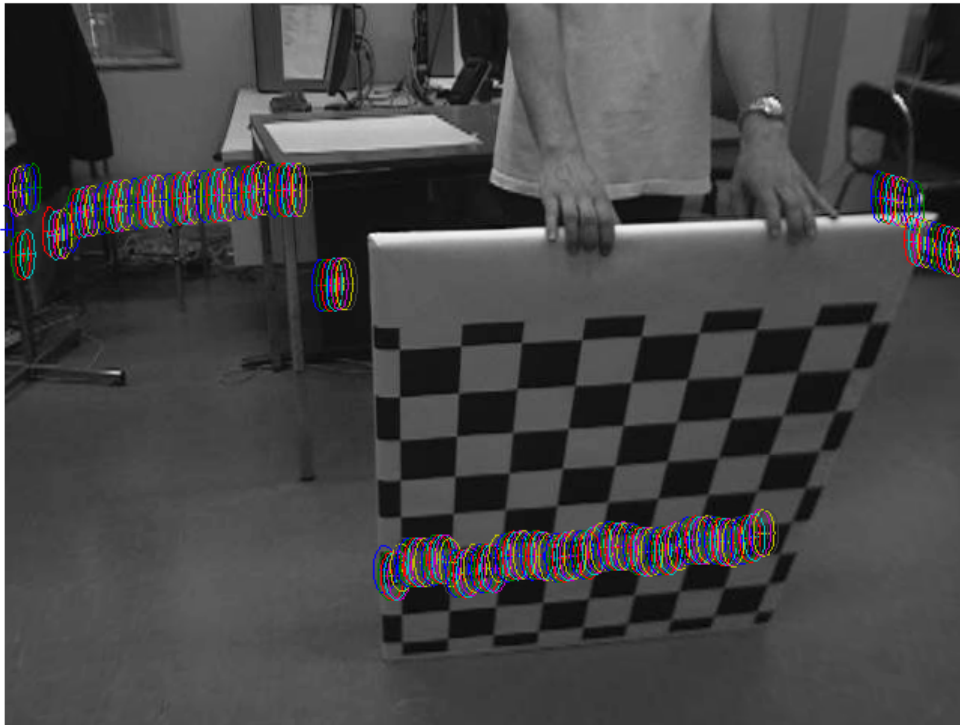
En ce qui concerne la mise en production du système de maintien à domicile des personnes âgées, une partie qui n'a pas été traitée par la présente thèse est l'entraînement des opérateurs à utiliser le système. Une étude en cette direction, se déroulant sur une période moyenne-longue, pourrait montrer les points forts et les points faibles rencontrés dans la pratique et offrir des nouvelles pistes de réflexions pour l'amélioration du système de téléopération. Les travaux décrits dans cette thèse étant dans un contexte de recherche, on n'a pas donné une grande attention à l'aspect financier. Les coûts principaux proviennent de l'achat du robot et de l'abonnement aux services fournis par le centre de télémédecine. Une réflexion sur la réduction de ces coûts devra être menée, pour pouvoir offrir la possibilité de bénéficier de ce système à un public le plus large possible.

Annexes

Les images suivant montrent des données utilisées pour la calibration laser-caméra effectuée pour le robot Lina. Les ellipses représentent l'incertitude associée à chaque point du laser qui apparait dans l'image.





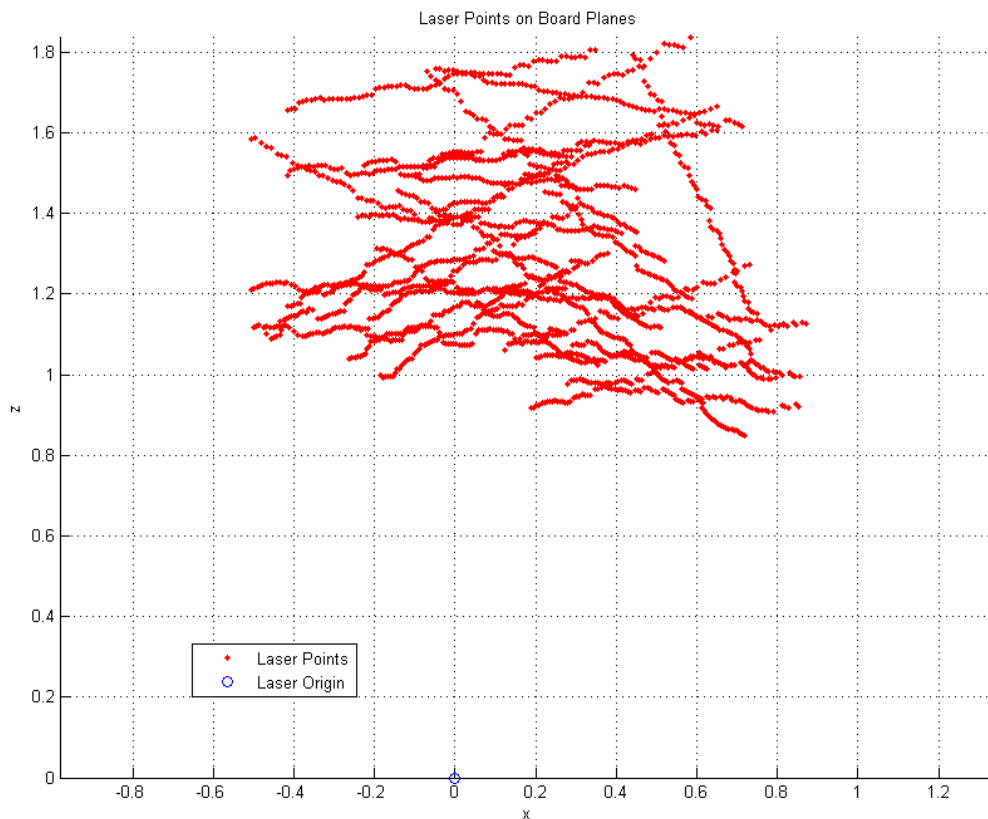


Un scan laser est une suite de nombres, en mètres. Chaque nombre est espacé de 0,5° de ses voisins. Un exemple typique de scan laser est le suivant :

1.6760 2.9060 5.0000 2.9200 2.9200 2.7970 2.7720 2.7730 2.7730 2.7720 2.7420 2.7350
2.6990 2.6980 2.6810 2.6690 2.6630 2.6360 2.6240 2.6080 2.5920 2.5760 2.5680 2.5480
2.5270 2.5220 2.5070 2.4870 2.4810 2.4810 2.4670 2.4530 1.0070 3.4180 1.0070 1.0170
1.0630 2.3500 2.3350 2.3350 2.3350 2.3200 2.3200 2.3120 2.2880 2.2780 2.2760 2.2580
2.2520 2.2410 2.2400 2.2290 2.2160 2.2140 2.2070 2.1890 2.1790 2.1660 2.1660 2.1660
2.1620 2.1500 2.1430 2.1360 2.1230 2.9700 2.1230 2.1070 2.0990 2.0980 2.0980 2.0800
2.0790 2.0740 2.0740 2.0680 2.0590 2.0550 2.0500 2.0420 2.0350 2.0320 2.0210 2.0200
2.0200 2.0150 2.0130 2.0130 2.0120 2.0040 1.9930 1.9910 1.9900 1.9870 1.9840 1.9780
1.9740 3.6100 1.9740 1.9660 1.9650 1.9650 1.9640 1.9500 1.9500 1.9490 1.9460 1.9440
1.9420 1.9420 1.9420 1.9420 1.9380 1.9380 1.9360 1.9360 1.9350 1.9300 1.9300 1.9290
1.9250 1.9210 1.9210 1.9240 1.9190 1.9180 1.9180 1.9180 1.9170 3.3540 1.9170 1.9150
1.9150 1.9150 1.9150 1.9150 1.9150 1.9160 1.9150 1.9160 1.9160 1.9150 1.9150 1.9140
1.9140 1.9140 1.9200 1.9140 1.9200 1.9170 1.9230 1.9260 1.9270 1.9290 1.9290 1.9290
1.9310 1.9370 1.9340 1.9360 1.9360 1.6260 1.9360 1.9360 1.9360 1.9410 1.9440 5.0000
5.0000 5.0000 1.9630 1.9620 1.9620 1.9630 1.9680 1.9680 1.9710 1.9740 1.9780 1.9780
1.9840 1.9840 1.9860 1.9880 1.9960 1.9990 2.0020 2.0130 5.0000 5.0000 5.0000 5.0000
5.0000 3.5460 5.0000 5.0000 2.0590 2.0590 2.0590 2.0620 2.0710 2.0730 2.0800 2.0810
2.0910 2.0950 2.0950 2.0950 2.0890 2.0660 2.0650 2.0400 2.0130 1.9890 1.9630 1.9240
1.9090 1.8860 1.8640 1.8640 1.8640 5.0000 5.0000 5.0000 1.6910 3.0980 1.6740 1.6580
1.6450 1.6210 1.6070 1.5820 1.5690 1.5580 1.5370 1.5360 1.5100 1.5020 1.4790 1.4660
1.4570 1.4380 1.4360 1.4180 1.4160 1.3950 1.3840 1.3820 1.3670 1.3560 1.3520 1.3430
1.3270 1.3080 1.3080 1.2970 1.2800 2.1380 1.2700 1.2650 1.2540 1.2510 1.2410 1.2320
1.2220 1.2190 1.2090 1.2080 1.2020 1.1990 1.1750 1.1740 1.1690 1.1670 1.1610 1.1530
1.1450 1.1380 1.1380 1.1200 1.1060 1.1030 1.1030 1.1030 1.1060 1.1100 1.1100 1.1100
1.0850 0.7940 1.0850 1.0880 1.0880 1.0880 1.0870 1.0810 1.0800 1.0790 1.0670 1.0640
1.0630 1.0500 1.0440 1.0380 1.0380 1.0350 1.0340 1.0280 1.0280 1.0270 1.0160 1.0120
1.0090 1.0090 1.0070 1.0010 0.9990 0.9930 0.9880 0.9850 0.9730 5.0000 0.9730 0.9730
0.9720 0.9710 0.9700 0.9670 0.9670 0.9590 0.9550 0.9550 0.9580 0.9550 0.9510 0.9480
0.9470 0.9440 0.9420 0.9360 0.9350 0.9320 0.9310 0.9280 0.9260 0.9260 0.9250 0.9250
0.9210 0.9190 0.9150 0.9130 0.9120 1.6900 0.9100 0.9100 0.9090 0.9090 0.9040 0.9090
0.9030 0.9030 0.9010 0.9010 0.8970 0.8970 0.8960 0.8920 0.8920 0.8950 0.8930 0.8930
0.8930 0.8920 0.8920 0.8920 0.8920 0.8940 0.8940 0.8910 0.8890 0.8900 0.8900 0.8900
0.8900 0.0900 0.8910 0.8930 0.8910 0.8910 0.8910 0.8900 0.8900 0.8940 0.8940 0.8940
0.8940 0.8940 0.8880 0.8840 0.8880 0.8880 0.8930 0.8930 0.8960 0.8960 0.8960 0.8950
0.8920 0.8880 0.8880 0.8950 0.8900 0.8980 0.9010 0.9010 0.9010 0.6660 0.9010 0.9020
0.9020 0.9070 0.9070 0.9070 0.9070 0.9110 0.9110 0.9120 0.9130 0.9140 0.9160 0.9160
0.9190 0.9220 0.9220 0.9230 0.9250 0.9260 0.9250 0.9280 0.9340 0.9340 0.9340 0.9370
0.9380 0.9390 0.9470 0.9470 0.9510 1.3700 0.9550 0.9560 0.9580 0.9590 0.9630 0.9630
0.9640 0.9640 0.9700 0.9720 0.9750 0.9760 0.9820 0.9850 0.9870 0.9950 0.9960 0.9980
1.0000 1.0020 1.0070 1.0070 1.0020 0.9970 0.9970 0.9970 1.0020 1.0040 1.0040 1.0040
1.0020 1.3060 1.0020 1.0060 1.0140 1.0270 1.0380 1.0430 1.0450 1.0520 1.0540 1.0560

1.0580 1.0630 1.0670 1.0710 1.0720 1.0750 1.0900 1.0910 1.0930 1.0990 1.1000 1.1130
 1.1190 1.1200 1.1310 1.1360 1.1370 1.1450 1.1540 1.1560 1.1700 1.1140 1.1730 1.1850
 1.1910 1.2050 1.2090 1.2100 1.2220 1.2250 1.2290 1.2430 1.2480 1.2570 1.2580 1.2660
 1.2820 1.2840 1.2850 1.2930 1.3000 1.3140 1.3160 1.3370 1.3440 1.3440 1.3630 1.3670
 1.3870 1.4030 1.4060 1.4130 1.4410 0.7940 1.4590 1.4600 1.4730 1.4900 1.5000 1.5210
 1.5330 1.5380 1.5580 1.5740 1.5920 1.6160 1.6240 1.6370 1.6560 1.6750 1.6980 1.7090
 1.7250 1.7460 1.7630 1.7850 1.8050 1.8290 1.8550 1.8680 1.8710 1.8880 1.9230 1.9370
 1.9510 3.4180 1.9710 2.0330 2.0410 2.0410 2.0440 2.0580 2.0580 2.0580 2.0770 2.0810
 2.0920 2.1160 2.1340 2.1670 2.1800 2.2330 2.2460 2.2960 2.3460 2.3770 2.4310 2.4860
 2.5330 2.6050 2.6550 2.7340 2.8520 2.9830 3.0050 5.0000 5.0000 3.0340 5.0000 5.0000
 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000
 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000 5.0000
 4.0950 4.0950 4.0950 4.0950 4.0950 1.1140 5.0000 5.0000 4.0950 4.0950 4.0950 4.0950
 4.0950 4.0950 1.4220 1.3700 1.3400 1.3320 1.3260 1.3160 1.3050 1.1560 1.1190 1.1140
 1.0950 1.0680 1.0560 1.0170 0.9920 0.9730 0.9410 0.9170 0.8670 0.5780 0.5750 0.5730

La figure suivante montre les différentes positions de la grille de calibration par rapport au laser. Son inclinaison a aussi été variée pendant l'acquisition des mesures.



Les paramètres de la caméra sont les suivants :

Distance focale $f_c = [603.22429 \ 602.46458] \pm [2.02103 \ 1.90465]$

Point principal $c_c = [283.70511 \ 234.00588] \pm [2.52172 \ 2.75242]$

Déformation $\alpha_c = [0.00000] \pm [0.00000] \Rightarrow$ angle des axes des pixels = 90.00000 ± 0.00000 degrees

Distorsion $k_c = [-0.46989 \ 0.22048 \ 0.00061 \ 0.00655 \ 0.00000] \pm [0.00710 \ 0.02227 \ 0.00101 \ 0.00083 \ 0.00000]$

Erreur des pixels $err = [0.21132 \ 0.25274]$

Champ de vue horizontal = 55.66°

Champ de vue vertical = 43.13°

Les valeurs obtenues à la fin de la calibration laser sont les suivantes :

Delta:[0.0291 0.815 0.695] \pm [0.00942 0.0186 0.00841]

Phi (in degrees):[24.3 -5.57 0.234] \pm [0.942 0.206 0.459]

Total rms error:0.00915

L'origine du système est le centre de la base du robot, au niveau du sol.

Bibliographie

- [Nadrag 2008] Nadrag, P., Maseda, A., Etxeberria, I., Delarue, S., Villanueva, M., Yanguas, J.J., Colle, E., Millán, J.C., Hoppenot, P., Home Robotic Companion to Assistance Living for Mild Cognitive Impairment Elderly, IROS 2008.
- [Nadrag 2010] Nadrag, P., Maseda, A., Grosdemouge, C., Delarue, S., Villanueva, M., Millán, J.C., Hoppenot, P., Remote control of a real robot taking into account transmission delays, IFAC 2010.
- [Maseda 2011] Maseda, A., Nadrag, P., Suárez, P., Delarue, S., Hoppenot, P., Millán-Calenti, L.C., A Mobile Robot to Offer Cognitive Intervention in Elderly with Memory Problems, IAGG 2011.
- [Devaux 2011] Devaux, J.C., Nadrag, P., Colle, E., Hoppenot, High level assisted control mode based on SLAM for a remote controlled robot, ICAR 2011.
- [Arioui 2003] Arioui, H., Kheddar, A., Mammari, S., A model-based controller for interactive force reflecting virtual environment under time delay, Journal of Intelligent and Robotic Systems, vol. 37, n° 2, pp. 193-207, 2003.
- [Armitage 2003] Armitage, G., An Experimental Estimation of Latency Sensitivity in Multiplayer Quake3, IEEE International Conference on Networks (ICON), 2003.
- [Baldwin 1999] Baldwin, J., Basu, A., Zhang, H., Panoramic video with predictive windows for telepresence applications, Robotics and Automation, vol. 3, pp. 1922-1927, 1999.
- [Bares 1997] Bares, J., Wettergreen, D., Lessons from the Development and Deployment of Dante II, Field and Service Robotics, pp. 72-79, 1997.
- [Becker 2006] Becker, M., Dantas, C.M., Macedo, W.P., Obstacle Avoidance Procedure for Mobile Robots, ABCM Symposium Series in Mechatronics, vol. 2, pp. 250-257, 2006.
- [behardware 2006] <http://www.behardware.com/articles/632-4/lcds-images-delayed-compared-to-crts-yes.html>, dernier accès: 1er septembre 2011.
- [Berti 2000] Berti, A., Frassinetti, F., When far becomes near: Remapping of space by tool use, Journal of Cognitive Neuroscience, vol. 12, n° 3, pp. 415-420, 2000.
- [Bluethmann 2003] Bluethmann, W., Ambrose, R., Diftler, M., Askew, S., Huber, E., Goza, M., Rehnmark, F., Lovchik, C., Magruder, D., Robonaut: A robot designed to work with humans in space, Autonomous Robots, vol. 14, n° 2, pp. 179-197, 2003.
- [Borenstein 1991] Borenstein, J., Koren, Y., The vector field histogram-fast obstacle avoidance for mobile robots, IEEE Transactions on Robotics and Automation, vol. 7, n° 3, pp. 278-288, 1991.
- [Brock 1999] Brock, O., Khatib, O., High-speed navigation using the global dynamic window approach, Robotics and Automation, vol. 1, pp. 341-346, 1999.
- [Brunelis 2008] de Brunelis, T.C., Le Blaye, P., Bonnet, P., Maille, N., Distant mission management and dynamic allocation of functions, Proceedings of Humans Operating Unmanned Systems (HUMOUS), pp. 3-4, 2008.

- [Burke 2004] Burke, J.L., Murphy, R.R., Covert, M.D., Riddle, D.L., Moonlight in Miami: A field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise, *Human-Computer Interaction*, vol. 19, n° 1, pp. 85-116, 2004.
- [CARMEN 2008] - <http://carmen.sourceforge.net>, dernier accès: 1er septembre 2011.
- [Cassidy 2009] Cassidy, A. M., Mental models, trust, and reliance: Exploring the effect of human perceptions on automation use, Master Thesis, 2009.
- [Chong 2003] Chong, N.Y., Kotoku, T., Ohba, K., Komoriya, K., Tanie, K., Oaki, J., Hashimoto, H., Ozaki, F., Maeda, K., Matsuhira, N., A collaborative multi-site teleoperation over an ISDN, *Mechatronics*, vol. 13, n° 8-9, pp. 957-979, 2003.
- [Civera 2010] Civera, J., Grasa, O.G., Davison, A.J., Montiel, J.M.M., 1-Point RANSAC for EKF Filtering. Application to Real-Time Structure from Motion and Visual Odometry, *Journal of Field Robotics*, vol. 27, n° 5, pp. 609-631, 2010.
- [Cobzas 2005a] Cobzas, D., Jagersand, M., Zhang, H., A Panoramic Model for Remote Robot Environment Mapping and Predictive Display, *International Journal of Robotics and Automation*, vol. 20, n° 1, pp. 25-33, 2005.
- [Cobzas 2005b] Cobzas, D., Jagersand, M., Tracking and Predictive Display for a Remote Operated Robot using Uncalibrated Video, *International Conference on Robotics and Automation*, pp. 1847-1852, 2005.
- [Courreges 2008] Courreges, F., Vieyres, P., Poisson, G., Robotized Tele-echography, *Teleradiology*, pp. 139-154.
- [Cowey 1998] Cowey, A., Small, M., Ellis, S., No abrupt change in visual hemineglect from near to far space, *Neuropsychologia*, vol. 37, n° 1, pp. 1-6, 1998.
- [Cybernetix 2011] www.cybernetix.fr, dernier accès: 1er septembre 2011.
- [Delarue 2007] Delarue, S., Contributions en vue de rendre plus naturelle l'interaction entre une personne handicapée et son assistance robotisée, thèse doctorat, 2007.
- [Di Nocera 2005] Di Nocera, F., Lorenz, B., Parasuraman, R., Consequences of shifting from one level of automation to another: main effects and their stability, *Human factors in design, safety, and management*, pp. 363-376, 2005.
- [Di Nocera 2007] Di Nocera, F., Camilli, M., Terenzi, M., Nacchia, R., Cognitive Aspects and Behavioral Effects of Transitions Between Levels of Automation, technical report, 2007.
- [Diosi 2004] Diosi, A., Kleeman, L., Advanced sonar and laser range finder fusion for simultaneous localization and mapping, *Intelligent Robots and Systems (IROS)*, vol. 2, pp. 1854-1859, 2004.
- [Dongen 2005] van Dongen, K., van Maanen, P.P., Designing for dynamic task allocation, *Seventh International Naturalistic Decision Making Conference (NDM7)*, 2005.
- [Duric 2002] Duric, Z., Gray, W.D., Heishman, R., Li, F., Rosenfeld, A., Schoelles, M.J., Schunn, C., Wechsler, H., Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction, *Proceedings of the IEEE*, vol. 90, n° 7, pp. 1272-1289, 2002.

- [Durrant-Whyte 2006] Durrant-Whyte, H., Bailey, T., Simultaneous localisation and mapping (SLAM): Part I the essential algorithms, *Robotics and Automation Magazine*, vol. 13, n° 2, pp. 99-110, 2006.
- [Edwards 2004] Edwards, J.L., A Generic, Agent-Based Framework for Design and Development of UAV/UCAV Control Systems, Technical Report, 2004.
- [Elfes 1989] Elfes, A., Using occupancy grids for mobile robot perception and navigation, *Computer*, vol. 22, n° 6, pp. 46-57, 1989.
- [Elfes 1996] Elfes, A., Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework, *Reasoning with Uncertainty in Robotics*, pp. 91-130, 1996.
- [Elmenreich 2008] Elmenreich, W., Constructing dependable certainty grids from unreliable sensor data, *Robotics and Autonomous Systems*, vol. 56, n° 12, pp. 1094-1101, 2008.
- [Elson 2002] Elson, J., Girod, L., Estrin, D., Fine-Grained Network Time Synchronization using Reference Broadcasts, *ACM SIGOPS Operating Systems Review*, vol. 36, pp. 147-163, 2002.
- [Endsley 1995] Endsley, M.R., Toward a theory of situation awareness in dynamic systems, *Human Factors*, vol. 37, n° 1, pp. 32-64, 1995.
- [Endsley 1999] Kaber, D.B., Endsley, M.R., The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task, *Theoretical Issues in Ergonomics Science*, pp. 1-40, 2004.
- [Ersson 2001] Ersson, T., Hu, X., Path planning and navigation of mobile robots in unknown environments, *International Conference on Intelligent Robots and Systems*, pp. 858-864, 2001.
- [Escribano 2004] Escribano, J., Cerrada, JA, Gomez, R., Cerrada, C., Basic Strategies to Enhance Teleoperation Platforms through the Internet, *ICEIS*, pp. 242-245, 2004.
- [Ferland 2009] Ferland, F., Pomerleau, F., Le Dinh, C.T., Michaud, F., Egocentric and exocentric teleoperation interface using real-time, 3D video projection, *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pp. 37-44, 2009.
- [Fernandez 2004] Fernandez, J.L., Sanz, R., Benayas, J.A., Dieguez, A.R., Improving collision avoidance for mobile robots in partially known environments: the beam curvature method, *Robotics and Autonomous Systems*, vol. 46, n° 4, pp. 205-219, 2004.
- [Fitts 1951] Fitts, P.M., *Human Engineering for an Effective Air Navigation and Traffic Control System*, 1951.
- [Fong 2001] Fong, T., Thorpe, C., Vehicle teleoperation interfaces, *Autonomous robots*, vol. 11, n° 1, pp. 9-18, 2001.
- [Fox 1997] Fox, D., Burgard, W., Thrun, S., The dynamic window approach to collision avoidance, *IEEE Robotics and Automation Magazine*, vol. 4, n° 1, pp. 23-33, 1997.
- [Friz 1998] Friz, H., Design of an Augmented Reality User Interface for an Internet based Telerobot using Multiple Monoscopic Views, thèse doctorat, 1998.
- [Gaillard 1997] Gaillard, J.P., *Psychologie de l'Homme au Travail: Les Relations Homme-Machine*, 1997.

- [Garcia 2003] Garcia, C.E., Carelli, R., Postigo, J.F., Soria, C., Supervisory control for a telerobotic system: a hybrid control approach, *Control engineering practice*, vol. 11, n° 7, pp. 805-817, 2003.
- [Gibson 1966] Gibson, J.J., *The senses considered as perceptual systems*, 1966 (réédité en 1983)
- [Gibson 1977] Gibson, J. J., *The theory of affordances*. In R. E. Shaw & J. Bransford (Eds.), *Perceiving, acting, and knowing: Toward an ecological psychology* (pp.67-82), 1977.
- [Gibson 1979] Gibson, J.J., *The ecological approach to visual perception*, 1986.
- [Grasso 1998] Grasso, R., Prévost, P., Ivanenko, Y.P., Berthoz, A., Eye-head coordination for the steering of locomotion in humans: an anticipatory synergy, *Neuroscience Letters*, vol. 253, n° 2, pp. 115-118, 1998.
- [Hamner 2008] Hamner, B., Singh, S., Roth, S., Takahashi, T., An efficient system for combined route traversal and collision avoidance, *Autonomous Robots*, vol. 24, n° 4, pp. 365-385, 2008.
- [Han 2001] Han, K.H., Kim, S., Kim, Y.J., Kim, J.H., Internet control architecture for internet-based personal robot, *Autonomous robots*, vol. 10, n° 2, pp. 135-147, 2001.
- [Hanford 2011] www.hanford.gov, dernier accès: 1er septembre 2011.
- [Hardin 2008] Hardin, B.C., *On Autonomous Multi-agent Control in Wilderness Search and Rescue: A Mixed Initiative Approach*, Master Thesis, 2008.
- [Henderson 2001] Henderson, T., Latency and user behaviour on a multiplayer game server, *3rd International Workshop on Networked Group Communications (NGC)*, 2001.
- [Hirche 2003] Hirche, S., Stanczyk, B., Buss, M., Transparent exploration of remote environments by Internet telepresence, *Proceedings of Int. Workshop on High-Fidelity Telepresence and Teleaction jointly with the conference HUMANOIDS*, 2003.
- [Hoc 2000] Hoc, J.M., From human-machine interaction to human-machine cooperation, *Ergonomics*, vol. 43, n° 7, pp. 833-843, 2000.
- [Inagaki 1998] Inagaki T, Moray N, Itoh M, Trust self-confidence and authority in human-machine systems. In *Proceedings of the IFAC man-machine systems*, pp. 431-436, 1998.
- [Internetworldstats 2011] www.internetworldstats.com, dernier accès: 1er septembre 2011.
- [Iriki 1996] Iriki, A., Tanaka, M., Iwamura, Y., Coding of modified body schema during tool use by macaque postcentral neurones, *Neuroreport*, vol. 7, n° 14, pp. 2325-2330, 1996.
- [Iwama 2004] Iwama, T., Kaneko, A., Machizawa, A., Toriyama, H., Real-time measurement of one-way delay in the internet environment, *Proceedings of ATF2004*, pp.317-320, 2004.
- [John 2003] St John, M., Kobus, D.A., Morrison, J.G., DARPA Augmented Cognition Technical Integration Experiment (TIE). DARPA Technical Report 1905. Décembre 2003.
- [Johnson 2007] Johnson, R., Testing adaptive levels of automation (ALOA) for UAV supervisory control, Technical Report, 2007.
- [Karsenty 1995] Karsenty, L., Brézillon, P., *Coopération homme-machine et explication*, *Le Travail Humain*, vol. 58, n° 4, pp. 289-310, 1995.

- [Keskinpala 2004] Keskinpala, H.K., Adams, J.A., Objective data analysis for a PDA-based human robotic interface, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2809-2814, 2004.
- [Keyes 2010] Keyes, B., Micire, M., Drury, J., Yanco, H., Improving human-robot interaction through interface evolution, Human-Robot Interaction, pp. 183-202, 2010.
- [Krishnamurthy 2007] Krishnamurthy, P., Khorrami, F., GODZILA: A Low-resource Algorithm for Path Planning in Unknown Environments, Journal of Intelligent and Robotic Systems, vol. 48, n° 3, pp. 357-373, 2007.
- [Krotkov 1996] Krotkov, E., Simmons, R., Cozman, F., Koenig, S., Safeguarded teleoperation for lunar rovers: From human factors to field trials, IEEE Planetary Rover Technology and Systems Workshop, 1996.
- [Lacquaniti 1983] Lacquaniti, F., Terzuolo, C., Viviani, P., The law relating the kinematic and figural aspects of drawing movements, Acta Psychologica, vol. 54, n° 1-3, pp. 115-130, 1983.
- [Lacquaniti 1987] Lacquaniti, F., Maioli, C., Anticipatory and reflex coactivation of antagonist muscles in catching, Brain Research, vol. 406, n° 1-2, pp. 373-378, 1987.
- [Lane 2001] Lane, J.C., Carignan, C.R., Akin, D.L., Advanced operator interface design for complex space telerobots, Autonomous Robots, vol. 11, n° 1, pp. 49-58, 2001.
- [Larson 2006] Larson, J., Bruch, M., Ebken, J., Autonomous navigation and obstacle avoidance for unmanned surface vehicles, SPIE Proc. 6230: Unmanned Systems Technology VIII, pp. 17-20, 2006.
- [Lee 2005] Lee, K.H., Kim, S.H., Kwak, Y.K., Mobility improvement of an Internet-based robot system using the position prediction simulator, International Journal of Precision Engineering and Manufacturing, vol. 6, n° 3, pp. 29-36, 2005.
- [Leishman 2010] Leishman, F., Horn, O., Bourhis, G., Smart wheelchair control through a deictic approach, Robotics and Autonomous Systems, vol. 58, n° 10, pp. 1149-1158, 2010.
- [Mantel 2009] Mantel, B. Information entre les sens et mouvement exploratoire pour percevoir si un objet est atteignable. Thèse de doctorat, 24 septembre 2009.
- [Massimo 1989] Massimo, M., Sheridan, T., Variable force and visual feedback effects and teleoperator man/machine performance, Nasa Conference on Space Telerobotics, 1989.
- [Mestre 1995] Mestre, D., Péruch, P., Rapport final sur l'expérimentation "vision mobile" réalisée dans le cadre du contrat "Vision mobile et téléopération", Cognition et Mouvement, 1995.
- [Meyer 2003] Meyer, J., Feinshreiber, L., Parnet, Y., Levels of automation in a simulated failure detection task, IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2101-2106, 2003.
- [Miller 2005] Miller, D.P., Machulis, K., Visual aids for lunar rover tele-operation, 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space, 2005.
- [Millot 2008] Millot, P. and Vanderhaegen, F., Toward cooperative and human error tolerant systems, 17th IFAC World Congress, 2008.

- [Mills 1989] Mills, D.L., On the Accuracy and Stability of Clocks Synchronized by the Network Time Protocol in the Internet System, ACM SIGCOMM Computer Communication Review, vol. 20, n° 1, pp. 65-75, 1989.
- [Mills 1997] Mills, D.L., Thyagarjan, A., Huffman, B.C., Internet Timekeeping Around the Globe, 1997.
- [Minguez 2004] Minguez, J., Montano, L., Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios, IEEE Transactions on Robotics and Automation, vol. 20, n° 1, pp. 45-59, 2004.
- [Monferrer 2002] Monferrer, A., Bonyuet, D., Cooperative robot teleoperation through virtual reality interfaces, Sixth International Conference on Information Visualisation, pp. 243-248, 2002.
- [Moon 2000] Moon, S.B., Measurement and Analysis of End-to-end Delay and Loss in the Internet, Thèse doctorat, 2000.
- [Moore 2009] Moore, K.S., Gomer, J.A., Pagano, C.C., Moore, D.D.W., Perception of Robot Passability With Direct Line of Sight and Teleoperation, Human Factors, vol. 51, n° 4, pp. 557-570, 2009.
- [Moray 2000] Moray, N., Inagaki, T., Itoh, M., Adaptive automation, trust, and self-confidence in fault management of time-critical tasks, Journal of Experimental Psychology: Applied, vol. 6, n° 1, pp. 44-58, 2000.
- [Morita 2002] Morita, M., Studies on Modeling Packet Delay Dynamics of the Internet using System Identification and its Application for Designing a Rate-Based Congestion Control Mechanism, Thèse master, 2002.
- [Munasinghe 2006] Munasinghe, S.R., Usui, T., Egashira, N., Lee, J.J., Nakamura, M., An Strategic Design of Internet-Based Telerobotic Systems, 2006.
- [NATO 2007] Uninhabited Military Vehicles (UMVs): Human Factors Issues in Augmenting the Force, Final Report of the RTO Human Factors and Medicine Panel (HFM) Task Group HFM-078/TG-017, 2007.
- [Nielsen 2004] Nielsen, C.W., Ricks, B., Goodrich, M.A., Bruemmer, D., Few, D., Few, M., Snapshots for semantic maps, IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2853-2858, 2004.
- [Nielsen 2007a] Nielsen, C.W., Bruemmer, D.J., Hiding the system from the user: Moving from complex mental models to elegant metaphors, The 16th IEEE International Symposium on Robot and Human interactive Communication (RO-MAN), pp. 756-761, 2007.
- [Nielsen 2007b] Nielsen, C.W., Goodrich, M.A., Ricks, R.W., Ecological interfaces for improving mobile robot teleoperation, IEEE Transactions on Robotics, vol. 23, n° 5, pp. 927-941, 2007.
- [Niemeyer 1996] Niemeyer, G., Using Wave Variables in Time Delayed Force Reflecting Teleoperation, thèse doctorat, 1996.
- [Paxson 1997] Paxson, V., End-to-end internet packet dynamics, ACM SIGCOMM Computer Communication Review, vol. 27, n° 4, pp. 139-152, 1997.
- [Philippsen 2004] - Philippsen, R., Motion planning and obstacle avoidance for mobile robots in highly cluttered dynamic environments, thèse de doctorat, 2004.

- [Piaget 1936] Piaget, J., *La Naissance de l'Intelligence chez l'Enfant*, 1936.
- [Pradalier 2005] Pradalier, C., Hermosillo, J., Koike, C., Braillon, C., Bessière, P., Laugier, C., *The cycab: a car-like robot navigating autonomously and safely among pedestrians*, *Robotics and Autonomous Systems*, vol. 50, n° 1, pp. 51-67, 2005.
- [Prendinger 2007] Prendinger, H., Ma, C., Ishizuka, M., *Eye movements as indices for the utility of life-like interface agents: A pilot study*, *Interacting with Computers*, vol. 19, n° 2, pp. 281-292, 2007.
- [Prinzel 2000] Prinzel, L.J., Freeman, F.G., Scerbo, M.W., Mikulka, P.J., Pope, A.T., *A closed-loop system for examining psychophysiological measures for adaptive task allocation*, *The International journal of aviation psychology*, vol. 10, n° 4, pp. 393-410, 2000.
- [Prinzel 2003] Prinzel III, L.J., Parasuraman, R., Freeman, F.G., Scerbo, M., Mikulka, P.J., Pope, A., *Three Experiments Examining the Use of Electroencephalogram, Event-Related Potentials, and Heart-Rate Variability for Real-Time Human-Centered Adaptive Automation Design*, Technical Report (No. NASA/TP-2003-212442), 2003.
- [Rabardel 1993] Rabardel, P., *Représentation dans des situations d'Activités Instrumentées, Représentation pour l'Action*, pp. 97-111, 1993.
- [Ricks 2004] Ricks, B., Nielsen, C.W., Goodrich, M.A., *Ecological displays for robot interaction: A new perspective*, *Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2855-2860, 2004.
- [Rover 2011] http://marsrovers.nasa.gov/mission/spacecraft_surface_rover.html, dernier accès: 1er septembre 2011.
- [Russell 2005] Russell, C.A., *Operator State Estimation for Adaptive Aiding in Uninhabited Combat Air Vehicles*, rapport n° AFIT/DS/ENG/05-01, thèse de doctorat, 2005.
- [Rybarczyk 2004] Rybarczyk, Y., *Etude de l'appropriation d'un système de téléopération dans l'optique d'une coopération homme-machine*, thèse de doctorat, 2004.
- [Schulz 1999] Schulz, D., Burgard, W., Cremers, A.B., *Robust visualization of navigation experiments with mobile robots over the internet*, *Intelligent Robots and Systems (IROS)*, vol. 2, pp. 942-947, 1999.
- [Schulz 2000] Schulz, D., Burgard, W., Cremers, A.B., *State estimation techniques for 3D visualizations of web-based tele-operated mobile robots*, *KI*, vol. 14, n° 4, pp. 16-22, 2000.
- [Shaw 1977] Shaw, R., Bransford, J., *Introduction: Psychological approaches to the problem of knowledge, Perceiving, acting and knowing*, pp. 1-39, 1977.
- [Sheridan 1992] Sheridan, T.B., *Telerobotics, automation, and human supervisory control*, 1992.
- [Simmons 2002] Simmons, R., Goodwin, R., Koenig, S., O'Sullivan, J., Armstrong, G., *Xavier: An Autonomous Mobile Robot on the Web, Beyond Webcams: an introduction to online robots*, pp. 81, 2002.
- [Simpson 2007] Simpson, T.W., Barron, K., Rothrock, L., Frecker, M., Barton, R.R., Ligetti, C., *Impact of response delay and training on user performance with text-based and graphical user interfaces for engineering design*, *Research in Engineering Design*, vol. 18, n° 2, pp. 49-65, 2007.

- [Stachniss 2002] Stachniss, C., Burgard, W., An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 508-513, 2002.
- [Stentz 1995] Stentz, A., Hebert, M., A complete navigation system for goal acquisition in unknown environments, *Autonomous Robots*, vol. 2, n° 2, pp. 127-145, 1995.
- [Steux 2010] Steux, B., El Hamzaoui, O., CoreSLAM: a SLAM Algorithm in less than 200 lines of C code, *Submission ICARCV*, 2010.
- [Strasdat 2010] Strasdat, H., Montiel, JMM, Davison, A.J., Scale drift-aware large scale monocular SLAM, *Proceedings of Robotics: Science and Systems (RSS)*, 2010.
- [Sun 2001] Sun, X., Foote, J., Kimber, D., Manjunath, B.S., Panoramic Video Capturing and Compressed Domain Virtual Camera Control, *Proceedings of the ninth ACM international conference on Multimedia*, pp. 329-347, 2001.
- [Taylor 2003] Taylor, R.M., Brown, L., Dickson, B., From safety net to augmented cognition: Using flexible autonomy levels for on-line cognitive assistance and automation, *Technical Report*, 2003.
- [Thompson 2000] Thompson, R.L., Daniel, R.W., Murray, D.W., Experiments on Operator Matching in Visual Teleoperation, *Systems, Man, and Cybernetics*, vol. 2, pp. 931-936, 2000.
- [Thrun 1998] Thrun, S., Learning metric-topological maps for indoor mobile robot navigation, *Artificial Intelligence*, vol. 99, n° 1, pp. 21-71, 1998.
- [Thrun 1999] Thrun, S., Bennewitz, M., Burgard, W., Cremers, A., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., MINERVA: A tour-guide robot that learns, *KI-99: Advances in Artificial Intelligence*, pp. 696, 1999.
- [Thrun 2003] Thrun, S., Learning occupancy grid maps with forward sensor models, *Autonomous Robots*, vol. 15, n° 2, pp. 111-127, 2003.
- [Ulrich 1998] Ulrich, I., Borenstein, J., VFH+: Reliable obstacle avoidance for fast mobile robots, *IEEE International Conference on Robotics and Automation*, 1998.
- [Ulrich 2000] Ulrich, I., Borenstein, J., VFH*: Local Obstacle Avoidance with Look-Ahead Verification, *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 2505-2511, 2000.
- [Vanderhaegen 1994] Vanderhaegen, F., Crevits, I., Debernard, S., Millot, P., Human-machine cooperation : toward an activity regulation assistance for different air-traffic control levels, *International Journal of Human-Computer Interaction*, vol. 6, n° 1, pp. 65-104, 1994.
- [Vanijjirattikhan 2008] Vanijjirattikhan, R., On Network-Based Control and Sensitivity Characterization of Mobile Robot in Intelligent Space, *thèse doctorat*, 2008.
- [Vieilledent 2001] Vieilledent, S., Kerlirzin, Y., Dalbera, S., Berthoz, A., Relationship between velocity and curvature of a human locomotor trajectory, *Neuroscience Letters*, vol. 305, n° 1, pp. 65-69, 2001.
- [Vinci 2010] http://www.intuitivesurgical.com/products/davinci_surgical_system, dernier accès: 1er septembre 2011.

- [Visser 2009] Visser, A., Slamet, B.A., Pfingsthorn, M., Robust weighted scan matching with quadtrees, 5th International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster (SRMED), 2009.
- [Viviani 1991] Viviani, P., Schneider, R., A developmental study of the relationship between geometry and kinematics in drawing movements, *Journal of Experimental Psychology: Human Perception and Performance*, vol. 17, n° 1, pp. 198-218, 1991.
- [Wang 2005] Wang, M., Liu, J.N.K., Interactive control for Internet-based mobile robot teleoperation, *Robotics and Autonomous Systems*, vol. 52, n° 2-3, pp. 160-179, 2005.
- [Wang 2008] Wang, M., Liu, J.N.K., Fuzzy logic-based real-time robot navigation in unknown environment with dead ends, *Robotics and Autonomous Systems*, vol. 56, n° 7, pp. 625-643, 2008.
- [Warren 1984] Warren, W.H., Perceiving affordances: visual guidance of stair climbing, *Journal of Experimental Psychology: Human Perception and Performance*, vol. 10, n° 5, pp. 683-703, 1984.
- [Warren 1991] Warren, W.H., Mestre, D.R., Blackwell, A.W., Morris, M.W., Perception of circular heading from optical flow, *Journal of Experimental Psychology: Human Perception and Performance*, vol. 17, n° 1, pp. 28-43, 1991.
- [Wood 2004] Wood, S., Automated behavior-based interaction customization for military command and control, *Intelligent User Interface, Workshop on Behavior-based User Interface Customization*, 2004.
- [Xu 2003] Xu, F., Van Brussel, H., Nuttin, M., Moreas, R., Concepts for dynamic obstacle avoidance and their extended application in underground navigation, *Robotics and Autonomous Systems*, vol. 42, n° 1, pp. 1-15, 2003.
- [Yahja 2000] Yahja, A., Singh, S., Stentz, A., An efficient on-line path planner for outdoor mobile robots, *Robotics and Autonomous systems*, vol. 32, n° 2, pp. 129-144, 2000.
- [Yerex 2003] Yerex, K., Cobzas, D., Jagersand, M., Predictive display models for tele-manipulation from uncalibrated camera-capture of scene geometry and appearance, *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 2812-2817, 2003.
- [Yokokohji 2001] Yokokohji, Y., Iimada, T., Iida, Y., Doi, T., Oda, M., Yoshikawa, T., Bilateral teleoperation: Towards fine manipulation with large time delay, *Experimental Robotics VII*, pp. 11-20, 2001.
- [Yoon 2001] Yoon, W.K., Goshozono, T., Kawabe, H., Kinami, M., Tsumaki, Y., Uchiyama, M., Oda, M., Model-based Teleoperation of a Space Robot on ETS-VII using a Haptic Interface, *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 407-412, 2001.
- [ZEUS 2000] <http://www.sti.nasa.gov/tto/spinoff2000/hm1.htm>, dernier accès: 1er septembre 2011.
- [Zieba 2009] Zieba, S., Polet, P., Vanderhaegen, F., Debernard, S., Resilience of a human-robot system using adjustable autonomy and human-robot collaborative control, *International Journal of Adaptive and Innovative Systems*, vol. 1, n° 1, pp. 13-29, 2009.
- [Zigoris 2003] Zigoris, P., Siu, J., Wang, O., Hayes, A.T., Balancing automated behavior and human control in multi-agent systems: a case study in RoboFlag, *American Control Conference*, vol. 1, pp. 667-671, 2003.