

UNIVERSITE D'EVRY VAL-D'ESSONNE

CEMIF : Centre d'Etude de Mécanique d'Ile de France.



Laboratoire Systèmes Complexes du CEMIF

THÈSE

pour l'obtention du titre de
Docteur de l'Université d'Evry Val-d'Essonne
Spécialité :
SCIENCES DE L'INGENIEUR

Télétravail et Réalité Virtuelle : application à la téléopération via Internet

Samir OTMANE

Soutenance le : 13 Décembre 2000

JURY

M. P. Coiffet	: Directeur de Recherche au CNRS,	Rapporteur
M. P. Fuchs	: MCF, HDR, ENSMP Paris,	Rapporteur
M. M. Soto	: MCF, HDR, Univ. Paris V,	Rapporteur
M. N. Agoulmine	: Professeur, Université d'Evry,	Rapporteur
M. F. Chavand	: Professeur, Université d'Evry,	Examineur
M. A. Kheddar	: MCF, Université d'Evry ,	Examineur
M. P. Even	: MCF, Versailles,	Examineur
M. M. Mallem	: Professeur, Université d'Evry,	Directeur de Thèse

Table des matières

Table des matières

Introduction	1
1 Le Télétravail: Etat de l'art	5
1.1 Bref historique	6
1.2 Les tendances technologiques pour le télétravail et le travail coopératif . .	7
1.2.1 Les tendances technologiques	7
1.2.2 Le travail coopératif	8
1.2.3 Les Interfaces Homme-Machine avancées	9
1.3 Télétravail: Téléopération et Télérobotique	10
1.3.1 La Téléopération	10
1.3.1.1 Le problème de délai de transmission	11
1.3.1.2 Evolution des systèmes de téléopération	12
1.3.2 La Télérobotique	13
1.3.2.1 Quelques domaines d'application	14
1.3.3 La réalité virtuelle et la réalité augmentée: Outils indispensables pour le télétravail	17
1.3.3.1 La réalité virtuelle: Bref Etat de l'art	17
1.3.3.2 Quelques domaines d'application de la RV	19
1.3.3.3 La réalité augmentée:	20
1.3.3.4 Quelques domaines d'application de la RA	21
1.3.3.5 Quelques exemples de systèmes de téléopération utilisant la RV et/ou la RA	23
1.4 Internet et les environnements télérobotiques: Bref état de l'art	25
1.4.1 Australia's Telerobot on the Web	26
1.4.2 Mercury Project	26
1.4.3 KhepOnTheWeb	27
1.4.4 Xavier	27
1.4.5 RHINO	27
1.4.6 PumaPaint project	28

1.5	Bilan	28
2	L'assistance graphique pour le Télétravail	31
2.1	Les métaphores graphiques : Bref état de l'art	31
2.1.1	La métaphore des mécanismes virtuels	32
2.1.2	La métaphore des guides virtuels	33
2.2	Les guides virtuels pour l'assistance au télétravail via Internet	39
2.2.1	Les méthodes de construction des guides virtuels	39
2.2.1.1	Les Coniques et les superconiques	40
2.2.1.2	Les superquadriques	41
2.2.1.3	Les B-Splines	44
2.2.2	Représentation et manipulation des guides virtuels sur écran	46
2.2.2.1	Le modèle de la caméra graphique	46
2.2.2.2	Sélection d'un objet de l'espace sur l'écran	48
2.2.2.3	Détection des collisions	50
2.3	Les propriétés des guides virtuels	54
2.3.1	La notion de guide virtuel simple et composé	55
2.3.1.1	Guide virtuel simple	55
2.3.1.2	Guide virtuel composé	56
2.3.2	Notion de guide virtuel passif ou actif	56
2.3.2.1	Guide virtuel passif	56
2.3.2.2	Guide virtuel actif	57
2.3.3	Formalisme proposé pour les guides virtuels	58
2.3.3.1	Structures de données proposées pour les guides virtuels	58
2.3.3.2	Les cas d'utilisation	60
2.4	Bilan	61
3	Le système expérimental de télétravail proposé	63
3.1	Caractéristiques d'un système de télétravail idéal	64
3.2	Modélisation géométrique d'environnement pour le contrôle en Réalité Augmentée	64
3.2.1	Modélisation géométrique de la scène	64
3.2.1.1	Représentation filaire	65
3.2.1.2	Représentations surfaciques	65
3.2.1.3	Représentations volumiques	65
3.2.1.4	La représentation choisie pour le télétravail via Internet	67
3.2.2	Modélisation et calibration de la caméra	68
3.2.2.1	Modèle interne de la caméra	69
3.2.2.2	Modèle externe de la caméra	70
3.2.2.3	Modèle global de la caméra	70
3.2.2.4	Modèle géométrique direct de la caméra (MGD)	71

3.2.2.5	Modèle géométrique inverse de la caméra (MGI)	71
3.2.2.6	Calibration de la caméra	72
3.2.3	Modélisation et calibration du robot	72
3.2.3.1	Modèle Géométrique Direct du robot (MGD)	74
3.2.3.2	Modèle Géométrique Inverse du robot (MGI)	74
3.2.3.3	Calibration de la tige du robot	75
3.3	Description du système ARITI	76
3.3.1	Description du matériel	76
3.3.2	Description du logiciel	77
3.3.2.1	Site distant (esclave)	77
3.3.2.2	Site opérateur (maître)	77
3.3.3	Module de Réalité Mixée (RM)	78
3.3.3.1	Les Données	78
3.3.3.2	Les fonctions d'assistance	80
3.3.4	Module de Téléopération	81
3.3.5	Module de Téléprogrammation	82
3.3.6	Module de Télésupervision et Télécoopération	84
3.3.6.1	Mission avec un seul opérateur	85
3.3.6.2	Mission avec quatre opérateurs en télécoopération	85
3.4	Architecture réseau du système ARITI	86
3.4.1	Protocole de communication client/serveur image	87
3.4.1.1	Demande d'une image vidéo	88
3.4.1.2	Schéma fonctionnel du serveur image	88
3.4.2	Protocole de communication client/serveur commande	91
3.4.2.1	Protocole de connexion pour la commande du robot réel	91
3.4.2.2	Protocole de télésupervision	91
3.4.2.3	Schéma fonctionnel du serveur commande	92
3.4.3	Limites et Sécurité du protocole	93
3.5	Bilan	93
4	Applications	97
4.1	Télétravail avec un robot à 4 ddl	97
4.1.1	Description du site esclave	97
4.1.2	Description du site client	99
4.1.2.1	IHM administrateur	99
4.1.2.2	IHM utilisateur	102
4.1.2.3	Description de l'outil de création et de manipulation des guides virtuels	102
4.1.3	Mode Téléopération	104
4.1.3.1	Tâche d'atteinte de cible	105
4.1.3.2	Tâche de suivi de cible mobile	107

4.1.3.3	Tâche de suivi de contour d'un objet	108
4.1.3.4	Tâche de saisie et dépôt d'un objet	110
4.1.4	Mode Téléprogrammation de tâches	115
4.1.5	Mode Télésupervision et Télécoopération	115
4.2	Télétravail avec un robot mobile	116
4.2.1	Description des robots mobiles	117
4.2.2	Description de l'IHM	118
4.2.3	Guides virtuels pour traverser une porte	119
4.2.3.1	tâche avec deux plans répulsifs	119
4.2.3.2	tâche avec une courbe B-Spline attractive	119
4.3	Autres applications potentielles	122
4.3.1	Nucléaire	122
4.3.2	Médical	122
4.3.3	Enseignement	123
4.4	Bilan	123
5	Evaluation et Expérimentation	125
5.1	Evaluation des performances des tâches avec l'interface de ARITI	126
5.1.1	Conditions expérimentales	126
5.1.2	Atteinte d'une cible	128
5.1.2.1	Tests sans guide virtuel	128
5.1.2.2	Tests avec guide virtuel répulsif	131
5.1.2.3	Tests avec guide virtuel attractif	132
5.1.3	Saisie et dépôt d'un objet	133
5.1.3.1	Saisie du cylindre numéro 1	133
5.1.3.2	Dépôt du cylindre numéro 1 sur le support numéro 3	134
5.2	Evaluation des algorithmes de compression d'images vidéo	136
5.2.1	Conditions expérimentales	136
5.2.1.1	Evaluation des taux de compression d'image	137
5.2.1.2	Evaluation des temps de compression d'image	138
5.3	Evaluation des performances de l'architecture client/serveur de ARITI	140
5.3.1	Connexion avec un seul opérateur	141
5.3.1.1	Connexion locale	141
5.3.1.2	Connexion moyenne distance	143
5.3.1.3	Connexion longue distance	146
5.3.2	Connexion avec plusieurs opérateurs avec retour prédictif distribué	147
5.3.2.1	Connexion locale	147
5.3.2.2	Connexion moyenne et longue distance	150
5.4	Bilan	151
	Conclusions	153

Table des figures

Liste des tableaux

Annexes	i
A Les méthodes de compression Gzip et Huffman	i
A.1 Codage de Huffman	i
A.1.1 Avantage de la compression d'Huffman	ii
A.1.2 Exemple de compression	ii
A.2 Codage de Gzip	ii
A.2.1 Avantage de la compression GZip	iii
B Schémas de fonctionnement des tâches de téléprogrammation	v
B.1 Atteindre un point cible	v
B.2 Atteindre une cible mobile	v
B.3 Suivre le contour du cylindre 1	v
B.4 Prendre/déposer un objet	v
B.5 Tour de Hanoï	vii
C Les sites clients ayant utilisé ARITI	ix
C.1 En Europe	ix
C.2 Dans le monde	x
D Les différents systèmes utilisant Internet comme moyen de contrôle . . .	xiii

Introduction

Amorcée dans les années 1970, l'informatisation professionnelle a récemment élargi son empire du poste de travail isolé à la prise en charge des tâches communicant par réseau. Les instruments de communication asynchrones tels que les messageries, le transfert de fichiers, les forums de discussions ou les serveurs de données ont aujourd'hui conquis une place de plus en plus importante dans les organisations. Cependant, la diffusion de ces outils reste pour l'heure très inégale et l'appréciation de leurs effets sociaux et économiques est sujette à controverse.

L'architecture des futures organisations "virtuelles" reposerait toute entière sur un réseau de machines à coopérer dotées du pouvoir d'abolir les distances et de faire naître entre ses membres une connectivité généralisée. Ces outils de communication, promettent aussi de créer un espace de travail commun permettant de partager des applications, d'échanger des documents animés ou sonores, d'écrire sur un même texte, d'élaborer un diagnostic médical sur un même cliché, et de faire circuler dans les réseaux des conférences, des cours ou des diagnostics, etc.

Enfin la perspective d'une mise à disposition de tels outils sur des réseaux non nécessairement professionnels (via la téléphonie large bande ou le réseau internet) ouvre des perspectives d'usages d'applications coopératives.

La téléprésence, ou la présence virtuelle, qui devient de plus en plus courante grâce à l'évolution technologique, implique une ouverture du potentiel du télétravail. En effet, on peut imaginer que certaines actions pourront un jour être effectuées à distance, comme par exemple des opérations chirurgicales accomplies par l'intermédiaire d'un robot piloté par le chirurgien se trouvant loin du patient. On voit que le potentiel du télétravail est fortement lié à la virtualité et à la téléprésence, donc aussi à la robotique et à l'informatique, autant qu'aux communications.

Cette thèse propose donc la réalisation d'un système permettant le télétravail via Internet.

Le *chapitre 1* est dédié à une analyse bibliographique sur le télétravail et une synthèse des problématiques auxquelles nous nous attacherons dans cette thèse. En effet, nous présentons au début de ce chapitre, un petit historique qui résume les origines du télétravail et comment cette nouvelle forme de travail est perçue dans le monde. Dans la seconde partie, nous présentons quelques tendances technologiques qui ont contribué à l'évolution du télétravail et du travail coopératif, en décrivant les principes et les concepts des nouvelles générations d'interfaces homme-machine avancées. Dans la troisième partie, nous présentons une autre forme de télétravail, celle liée au contrôle à distance de systèmes robotisés, nous parlerons alors de la télérobotique et de la téléopération et nous verrons comment les techniques de la réalité virtuelle et augmentée ont contribué à percer certains

verrous liés généralement à la distance qui sépare l'opérateur du robot à contrôler. Dans la quatrième partie, nous présentons brièvement quelques systèmes robotiques qui utilisent Internet comme moyen de communication et nous présentons aussi les performances et les limites de ces systèmes. Nous terminons ce chapitre, par un bilan à partir duquel nous dégagerons quelques problématiques, essentiellement pour la téléopération de robots sur Internet.

Dans le *chapitre 2* nous présentons quelques solutions apportées aux problèmes d'assistance à l'opérateur pour le télétravail, en particulier d'assistance graphique à la téléopération d'un robot. En effet, dans la première partie, nous présentons un bref état de l'art sur les métaphores graphiques. Nous verrons comment elles sont utilisées suivant les auteurs et suivant les champs d'application, bien qu'elles soient issues d'une même technologie, à savoir la réalité virtuelle. Dans la seconde partie, nous présentons les différentes méthodes, géométriques et analytiques utilisées pour la construction, la représentation et la manipulation des objets virtuels en général. D'autre part, nous verrons comment ces objets virtuels sont transformés en de véritables outils d'assistance pour la téléopération. Afin de rendre ces outils facilement configurables par un opérateur, un formalisme est proposé et implanté. La structure générale de ce formalisme est représentée à la fin de ce chapitre.

Le *chapitre 3* présente une étude à la mise en œuvre d'un système de télétravail. Nous présentons les outils logiciels et matériels nécessaires à la réalisation d'un système de télétravail, comment intégrer les nouveaux concepts et les solutions déjà apportées aux problèmes liés à la télérobotique, sur une plate-forme universelle et à moindre coût. Nous verrons, comment rendre l'utilisation des systèmes de télétravail suffisamment souple et intuitive pour l'opérateur, comment les outils d'assistance présentés dans le chapitre 2 peuvent être utilisés pour améliorer les performances de l'opérateur lors du télétravail. Dans la première partie de ce chapitre, nous présentons les caractéristiques d'un système de télétravail idéal. Ensuite, nous présentons les méthodes nécessaires pour le contrôle en réalité augmentée. Il s'agit des méthodes de modélisation géométrique de l'environnement, de calibration de la caméra et du robot. Dans la troisième partie, nous présentons notre système expérimental de télétravail, baptisé ARITI "Augmented Reality Interface for tele-robotic applications via Internet". Il s'agit d'un système de télétravail dont la conception et la mise en œuvre sont basées d'un côté sur des techniques de la réalité virtuelle et augmentée et de l'autre côté sur l'exploitation des techniques de développement réseaux et en particulier sur Internet. Dans la quatrième partie, nous présentons l'architecture réseau et les différents protocoles de communication de l'architecture client/serveur du système ARITI.

Le *chapitre 4* présente deux applications réalisées, une de télémanipulation et l'autre de téléopération d'un robot mobile. Nous présentons aussi quelques applications potentielles pouvant bénéficier du système de télétravail décrit dans le chapitre 3. La première partie de ce chapitre, présente une application réalisée avec un banc d'essai à 4 degrés de libertés transformé en robot téléopérable via Internet grâce au système ARITI. Nous décrirons les tâches réalisées suivant les différents modes de télétravail ainsi que les guides virtuels que nous avons utilisé pour réaliser quelques tâches en téléopération. Nous présentons ensuite, l'interface homme-machine pour cette application ainsi que le modèleur 3D que nous avons réalisé pour créer et manipuler ces guides virtuels. La seconde partie, présente une application d'un projet en cours de réalisation pour l'assistance aux

personnes handicapées (un projet en collaboration avec l'Association Française contre la Myopathie (AFM)) utilisant un robot mobile. Dans la troisième partie de ce chapitre, nous présentons d'autres domaines d'applications où l'utilisation d'un tel système de télétravail peut être intéressante.

Le *chapitre 5* présente les différentes expérimentations réalisées avec le système ARITI utilisant un robot esclave à 4 ddl. Dans la première partie, nous évaluons les performances de certaines tâches avec le mode de téléopération en utilisant l'interface de ARITI; Pour cela, plusieurs opérateurs humains ont participé aux expérimentations. Il s'agit de réaliser des missions de téléopération d'une part, sans assistance de guides virtuels et d'autre part, avec assistance de guides virtuels répulsifs ou attractifs.

Dans la seconde partie, nous évaluons les différents algorithmes de compression d'images vidéo. Il s'agit d'étudier d'une part, les taux de compression fournis par les différentes méthodes de compression d'images et les temps de compression d'images pour chaque méthode utilisée. Ces évaluations vont nous permettre de déterminer un algorithme de compression implanté dans le serveur d'images permettant ainsi de choisir automatiquement la méthode à utiliser suivant la dynamique des images. Dans la troisième partie, Nous étudions le comportement du système et les performances de l'architecture client/serveur du système ARITI à la fois, en fonction de la distance séparant le site maître (client) du site esclave et en fonction du nombre d'opérateurs utilisant simultanément le système pour un travail coopératif avec un retour prédictif distribué.

Chapitre 1

Le Télétravail : Etat de l'art

Souvent considéré comme une conséquence du progrès technique, le télétravail n'est pas si nouveau qu'on voudrait nous le faire croire. Au 18ème siècle, dans la période pré-industrielle, les travaux manufacturés se faisaient à domicile. A l'époque, les échanges avec l'entrepreneur obligeaient ce dernier à parcourir les foyers pour distribuer les matériaux et récupérer l'ouvrage terminé. C'est l'industrialisation qui, au 19e siècle a nécessité des moyens de productions capitalistique. Il a alors fallu regrouper les travailleurs dans un lieu de production centralisé et unique, appelé alors "usine".

Les technologies de l'information ont simplement permis d'un côté, pour les activités du tertiaire de ne plus concentrer la production du service dans un lieu unique. D'un autre côté, ces technologies ont permis de dispenser la présence de l'homme dans des milieux dangereux (milieu nucléaire, haute température, etc.) ou tout simplement pour des missions d'exploration (milieu spatial etc.).

Nous commençons ce chapitre par un petit historique qui rappelle les origines du télétravail et comment cette nouvelle forme de travail est perçue dans le monde, comme porteuse d'espoir pour certains et spectre effrayant pour d'autres. Dans la deuxième section, nous présentons quelques tendances technologiques qui ont contribué considérablement à l'évolution du télétravail et du travail coopératif. Nous décrivons ensuite, les principes et les concepts des nouvelles générations d'interfaces homme-machine avancées. Dans la troisième section, nous présentons une autre forme de télétravail, celle liée au contrôle à distance de systèmes robotisés, nous parlerons alors de la télérobotique et de la téléopération et nous verrons comment les techniques de la réalité virtuelle et augmentée ont contribué à percer certains verrous liés généralement à la distance qui sépare l'opérateur du robot à contrôler. Dans la quatrième section, nous présentons brièvement quelques systèmes robotiques qui utilisent Internet comme moyen de communication et nous présentons aussi les performances et les limites de ces systèmes. Nous concluons ce chapitre, par un bilan dans lequel nous dégagerons quelques problématiques, essentiellement pour la téléopération de robots sur Internet.

1.1 Bref historique

Durant les crises pétrolières des années 1970, le concept de déplacer le travail plutôt que le travailleur qui consomme de l'essence avait commencé à séduire. Mais il a fallu du temps pour que l'idée prenne forme, car certains facteurs devaient être en place pour qu'elle puisse réussir (existence de technologies bon marché et fiables, réduction des coûts des télécommunications, etc.). De l'autre côté, une autre forme de travail à distance est apparue en 1971, lorsque l'Union Soviétique a réussi à commander à distance un robot mobile sur la Lune (cette forme de télétravail est appelée "téléopération" voir la section 1.3).

La période de crise, entre la fin des années 80 et le début des années 90, n'a guère aidé l'évolution du télétravail. Aux Etats-Unis comme au Royaume-Uni, des prédictions ambitieuses avaient été avancées très tôt sur le potentiel du télétravail, prédictions qui se sont avérées erronées. Par exemple, en 1985, le "Institute for the future" de Palo Alto avait prédit que 40% (Béréziat *et al.*, 2000) des employés américains seraient des télétravailleurs en l'an 2000.

Le rapport *Telefutures*¹ de 1996 résume l'histoire: "Malgré un démarrage assez lent, l'Union Européenne (UE) a placé, en 1994, l'adoption du télétravail au sommet de la liste d'actions du rapport Bangemann pour rendre l'Europe plus compétitive." L'UE s'est même donnée comme objectif d'atteindre les 10 millions de télétravailleurs en Europe pour l'an 2000. Le rapport *Telework 99* de la Commission Européenne présente deux chiffres de sources différentes qui estiment le nombre de télétravailleurs en Europe entre 6 et 9 millions à ce jour.

Des études récentes suggèrent que vue les nombreux changements induits par la "société de l'information", l'Amérique du Nord a pris une longueur d'avance sur l'Europe et connaît aujourd'hui une évolution rapide du nombre de ses télétravailleurs. Jack Nilles² (Nilles, 1998) pense qu'il y avait peut être 2000 télétravailleurs en 1970, qui seraient passés à une centaine de milliers en 1980, bien qu'il reconnaisse qu'aucune enquête adéquate n'était faite à l'époque. Ses estimations pour 1990 étaient de 2.4 millions. En 1994 elles s'élevaient à 7.8 millions et 12.7 millions pour 1996. Nilles prédit que ce chiffre doublera presque pour atteindre 24.7 million en l'an 2000.

Cependant, le télétravail continue à se développer très rapidement en Europe, bien qu'il y a dix ans, le télétravail était une façon particulière de travailler, concernant seulement une élite technologique. Mais il s'est depuis transformé en une forme d'organisation du travail de plus en plus banale dans certains Etats-membres et/ou secteurs économiques importants. En 1999, il y a suivant les études (Béréziat *et al.*, 2000), de 6.5 à 9 millions d'Européens engagés dans ces nouvelles pratiques de travail impliquant directement l'utilisation des Nouvelles technologies d'Information et de Communication (NTIC). La répartition est inégale entre les Etats-membres comme le montre le tableau 1.1.

Comme il existe encore une variété de termes utilisés pour décrire les nouvelles formes de travail, le travail à distance étant une appellation générique, les chercheurs ne sont pas systématiquement d'accord sur ce que constitue le télétravail. Ceci explique la disparité entre les différentes estimations du nombre de télétravailleurs. Ces données permettent

1. *Telefutures*: A study on teleworking in Ireland (voir <http://www.forbairt.ie/telefutures>)

2. Jack Nilles est généralement considéré comme "le père du télétravail" avec des publications régulières qui font référence sur le sujet depuis 1973.

	Nbr. télétrav. (*1000)	% télétrav.	% Progres.97 – 98
Allemagne	1 800	5.1	+53
Autriche	67	2.0	+33
Belgique et Luxembourg	250	6.2	+25
Danemark	300	11.6	+20
Espagne	120	0.9	+50
Finlande	220	10.0	+59
France	420	1.8	+67
Grèce	50	1.3	+160
Irlande	58	7.1	+16
Italie	350	1.7	+40
Pays-Bas	1 200	18.2	+100
Portugal	100	2.2	+67
R.U	1 455	5.5	+13
Suède	300	9.0	+67
Total UE	6 690	4.5	+45
USA ^a	14 700	12.9	+42

TAB. 1.1 – Développement du télétravail en Europe, 1998-1999. d'après (Béréziat et al., 2000)

^aSource USA: Cyber Dialogue, New-York, mid-year 1998 telecommuters : <http://www.cyberdialogue.com/press/release/telecommuting.html>

seulement de tracer les grandes tendances, et de donner une idée sur la manière dont des conditions sociales et/ou légales différentes favorisent ou non le développement du télétravail.

1.2 Les tendances technologiques pour le télétravail et le travail coopératif

1.2.1 Les tendances technologiques

Les développements considérables et extrêmement rapides de la technologie et la large diffusion des micro-ordinateurs performants à des prix abordables ainsi que la baisse des coûts des services de télécommunications, ont rendu le télétravail possible.

Les lignes RNIS³ (ou Numéris, nom commercial du service proposé par France Télécom) offrent aux télétravailleurs la possibilité de transférer des fichiers très rapidement et d'avoir accès à des outils de travail coopératifs synchrones comme les visioconférences. La diffusion du RNIS permet aux télétravailleurs d'échanger voix, images et données avec un débit élevé. Il est probable que la technologie RNIS sera fortement concurrencée par la nouvelle technologie ADSL (Asymmetric Digital Subscriber Line), fournissant des débits encore plus élevés que le RNIS et ceci sur une simple ligne téléphonique. D'autres technologies liées aux réseaux câblés et à la diffusion par satellite, peuvent chacune procurer un accès rapide à Internet.

Les progrès techniques considérables dans le domaine de la compression et décompression de la voix ou d'images pour la visionconférence, ont permis à la téléphonie sur Internet d'apparaître en codant la voix en signaux numériques pouvant être transmis par Internet. Bien que la qualité ne soit pas aussi bonne que celle des lignes téléphoniques classiques ou Numéris, et qu'il subsiste des problèmes de compatibilité et de normalisation entre systèmes, ainsi que les phénomènes d'encombrement du réseau mondial, cet usage est acceptable essentiellement pour des utilisateurs non professionnels à l'heure actuelle.

1.2.2 Le travail coopératif

On appelle "travail coopératif", le travail en équipe ("Teamwork") (Bannon et Schmidt, 1989), (Schal et Zeller, 1990), (Greenberg, 1991), tout ce qui entraîne des besoins de communication et de coordination entre personnes afin d'assurer conjointement un travail.

Le "*Computer Supported Cooperative Work*" (CSCW), traduit en français sous le terme "Activités multi-participantes informatisées", est le domaine de recherche répondant aux questions suivantes: quelles sont les caractéristiques du travail coopératif comme opposé au travail effectué par des individus isolés? Comment l'informatique peut-elle être appliquée pour résoudre les problèmes logistiques du travail coopératif? Comment les concepteurs abordent-ils les délicats et complexes problèmes de conception des systèmes qui façonnent les relations sociales?

3. RNIS: Réseaux Numérique à Intégration de service

Le CSCW est un domaine de recherche multi-disciplinaire intéressant, les sociologues, les psychologues, les ergonomes et les informaticiens. Bien que le CSCW comporte le terme “Computer”, les outils mis en jeu dans cette discipline dépassent de loin l’ordinateur en incluant la téléphonie, les messageries, la vidéo et les systèmes d’imagerie. En tant que domaine scientifique, le “Travail Coopératif” est encore immature et les problèmes théoriques sont nombreux.

Les verrous existant sont d’une manière générale à la fois d’ordre :

- **Informatique ou communicationnel** : Transmettre avec une bande passante suffisante, développer des systèmes d’installation et de maintenance commodes pour des utilisateurs qui ne sont souvent pas les prescripteurs et,
- **Ergonomiques et d’usage** : Développer des interfaces coopératives multimédia conviviales et d’un apprentissage aisé.

De manière plus précise et non nécessairement exhaustive, on peut noter les problèmes ergonomiques liés à l’existence même d’une interface pour les systèmes de vidéo-conférence : lorsque deux individus entrent en communication en face à face, il n’y a pas d’interface, pas de boutons à cliquer ou de numéro à composer. La complexité des interfaces des systèmes de communication actuels est telle qu’elle est un obstacle à l’utilisation de ces systèmes (sans parler des manipulations à réaliser pendant la session elle-même, par exemple lorsque que l’on veut ajouter un participant, partager un document ou segmenter les participants en sous-sessions de travail disjointes). Il est donc nécessaire d’inventer de nouvelles interfaces pour ces systèmes, plus “transparentes” sinon invisibles.

Un autre verrou d’utilisabilité des outils CSCW réside dans leur manque d’intégration. Alors que les activités de groupes sont “à géométrie variable”, les outils existants ne couvrent en général qu’un mode de coopération : synchrone/asynchrone (ou bien temps réel/temps différé). La réalisation de tels environnements CSCW remet en question une bonne partie des infrastructures logicielles existantes (systèmes d’exploitation, protocoles réseau, etc.).

Une autre difficulté du même ordre réside dans la disponibilité d’outils qui s’exécutent de façon transparente dans des environnements hétérogènes en adoptant le “look and feel” de chaque plate-forme. La question est donc de simplifier les tâches cognitives de l’utilisateur en déléguant le plus possible de celles-ci aux applications. Il s’agira donc ici de rendre cette hétérogénéité transparente aux utilisateurs tout en exploitant au mieux les capacités de chaque plate-forme (i.e. sans s’en remettre au plus grand dénominateur commun, comme le fait par exemple JAVA⁴).

1.2.3 Les Interfaces Homme-Machine avancées

Il convient de faciliter l’accès de l’utilisateur au “monde”, incluant l’information et la connaissance mais allant au delà. L’enjeu des Interfaces Homme-Machine (IHM) avancées est de proposer de nouveaux moyens d’accès à l’information afin que, quels que soient les changements d’environnement, ces informations et ces connaissances soient accessibles. Ces contraintes d’accès sont mises en évidence dans des conditions liées aux transports,

4. Un langage de programmation objet de chez “Sun Microsystems”, il génère un code exécutable universel

à la mobilité, à l'internationalisation et au handicap.

Les IHM avancées vont permettre la création de services fédérateurs, comme l'accès à l'ensemble de la messagerie asynchrone (vocal, mail, fax). La reconnaissance et la synthèse de la parole viennent compléter ou remplacer clavier et écran. Le poste informatique personnel et/ou la base de données commune deviennent télé-opérables. Internet et téléphonie mobile, le mariage est séduisant. Il devient détonnant lorsque l'on y ajoute les possibilités des IHM avancées.

Il convient aussi de réaliser des systèmes de communication plus naturels, ce qui entraîne bien sûr une plus grande complexité dans leur fonctionnement. Combien de personnes n'utilisent qu'une faible partie des fonctions de leur traitement de texte ou de leur téléphone d'entreprise? Combien de personnes sont également hermétiques aux concepts du monde informatique, principalement ceux du PC? Il faut donc arriver à des interfaces d'apparence simples, permettant au plus grand nombre de communiquer en utilisant les nouvelles technologies disponibles, mais l'effort pour y arriver ne saurait être sous-estimé. La communication homme-machine qui faisait apparaître ces deux acteurs face à face évolue à présent vers un partenariat homme-machine qui les place côte à côte pour résoudre ensemble, et en coopération, un problème commun. Les domaines couverts par ce thème concernent les différentes modalités nouvelles de communication : langages (le vocal ou l'écrit), visuelles et graphiques, gestuelles (dans les aspects d'analyse et de synthèse/retour d'efforts), ainsi que leur combinaison dans des systèmes de communication multimodales, ou dans des systèmes de réalité virtuelle ou augmentée. Dans tous ces cas, les aspects télématiques et communication à distance doivent être présents, voir essentiels.

1.3 Télétravail : Téléopération et Télérobotique

Nous proposons ici un bref historique et une synthèse ainsi qu'un classement rapide des systèmes robotiques sur la base du *mode de contrôle* utilisé. Le terme *contrôle* désigne ici aussi bien la commande au sens de l'automatique que les interactions entre l'homme et la machine. Généralement le mode de contrôle est choisi en fonction de l'application suivant plusieurs critères (La complexité de la tâche à accomplir, la nature des moyens de transmission, la charge acceptable par l'opérateur, la puissance de calcul et les algorithmes embarqués à bord, les capteurs disponibles, etc.).

Bref historique

Vers l'année 1770, la construction de machines simples, ainsi que leur mécanisation a marqué le départ de la révolution industrielle. Au début du 20ème siècle, les machines automatiques fixes et les chaînes de montage permettent le développement de la production de masse. Le travail à la chaîne favorise l'accroissement des taux de production en réduisant les temps de fabrication. Les cycles d'usinage et de montage sont simples et répétitifs, à cadence fixe. Apparaissent alors des machines-outils munies de commandes automatiques rudimentaires, telles que les matrices de programmation par fiches qui permettent d'effectuer des séquences d'opérations prédéterminées. Les machines à copier permettent d'usiner des pièces par copie des déplacements d'un palpeur sur un gabarit modèle au moyen de moteurs asservis qui commandent les mouvements de l'outil d'usinage sur l'ébauche (Koren, 1985). En 1953, des recherches menées au MIT conduisent

au développement d'une nouvelle technologie de machines numériques adaptées à des opérations précises et répétées. Le concept du robot industriel est breveté en 1954 (brevet US n° 2988237). Ce brevet décrit la réalisation d'un bras mécanique asservi capable d'effectuer des tâches à caractère industriel. En 1958, Planet Corporation (USA) combine avec succès cette technologie avec la technologie des télémanipulateurs développée pour l'industrie nucléaire américaine et crée le premier robot manipulateur industriel. La société Unimation voit le jour peu après, et livre le premier *Unimate* à General Motors en 1961. Vers la fin des années 70, ces robots dits "de première génération" se généralisent à l'ensemble de la production industrielle. Pendant que se développe ce nouveau concept "d'usine automatique", la robotique aborde d'autres domaines. Ainsi, l'Union Soviétique réussit à téléopérer un robot mobile sur la Lune en 1971. En 1976, la sonde *Viking I* de la NASA⁵ se pose sur la planète Mars, équipée d'un bras manipulateur téléopéré depuis la terre afin de prélever des échantillons de sol et de rochers (Freedman, 1993) ou encore la mission du robot *sojourner* en 1997.

1.3.1 La Téléopération

Dans cette partie, nous présentons quelques concepts de base et quelques problèmes de fond liés à la téléopération. Pour un état de l'art sur la technologie de la téléopération nous conseillons les références suivantes, (Vertut et Coiffet, 1984) et (Sheridan, 1992).

La Téléopération désigne les principes et les techniques qui permettent à l'opérateur humain d'accomplir une tâche à distance, à l'aide d'un système robotique d'intervention (*dispositif esclave*), commandé à partir d'une station de contrôle (*dispositif maître*) (figure 1.1), par l'intermédiaire d'un canal de télécommunication. Elle a pris ses origines dans le besoin de prolonger le geste de l'homme au delà de la main, et se poursuit par l'ambition de se trouver là ou on ne se trouve pas (*Téléprésence*) (figure 1.2).

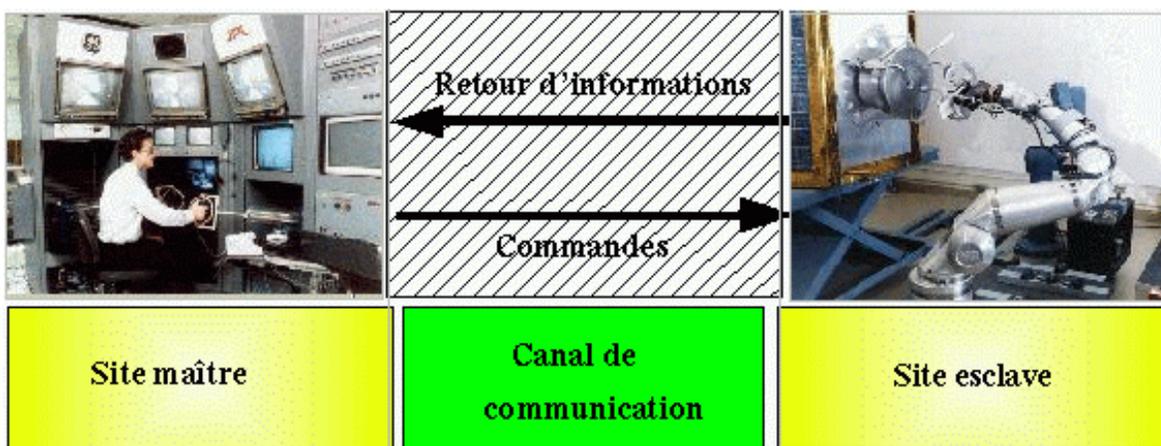


FIG. 1.1 – Illustration de l'architecture générale d'un système de Téléopération

De nos jours, la téléopération s'applique aussi bien aux bras redondants, aux préhenseurs mécaniques polyarticulés, et à diverses sortes de robot mobiles à roues (Clement *et al.*, 1988), (Causse et Crowley, 1993), et à pattes (Kheddar *et al.*, 1996) . Cependant, un certain nombre de problèmes de fond et complexes existent encore. Nous citerons par exemple, l'inévitable problème de délai dû au canal de transmission entre les deux sites,

5. National Aeronautics and Space Agency



FIG. 1.2 – Exemples de système : A gauche, le système de téléopération du CEA/CEREM; A droite, le système de téléprésence avec un robot mobile du groupe de recherche en téléprésence “California Corporation Telepresence Research”

maître et esclave qui implique un retard de transmission d’informations sensorielles utiles à l’opérateur. En effet tout retard de transmission entraîne des instabilités difficilement compensables (Ferrell, 1965), (Ferrell, 1966), (Bejczy *et al.*, 1990) et (Sheridan, 1993). d’Autres problèmes qui ne cessent de préoccuper l’esprit de nombreux chercheurs dans ce domaine concernent le choix d’un système de commande universel (Bejczy, 1992) ainsi que la compréhension de nos propres sous-systèmes sensoriels (Brooks, 1990) et (Lederman et Klatzky, 1994).

1.3.1.1 Le problème de délai de transmission

De nombreuses recherches ont été réalisées pour l’étude et l’analyse du problème de délai (retard) de transmission en téléopération. Deux approches ont été utilisées afin d’analyser ce problème (Kheddar, 1997). La première est basée sur l’automatique classique et la seconde est basée sur la théorie des lignes et de la passivité.

La première approche, est de nature compensatoire, l’idée était d’introduire des correcteurs dans la boucle d’asservissement afin de compenser les systèmes à retards (DeLarminat, 1993).

La seconde approche, considère le retard comme subi, c’est à dire comme partie intégrante du système (Niemeyer et Slotine, 1991), (Leung et Francis, 1992), (Spong, 1993) et (Lawrence, 1993) et (Andriot, 1997).

Malheureusement, ces deux approches exigent que le délai de transmission soit connu, alors que la majorité des protocoles publics actuels de transmission de données informatique ne garantissent pas la constance des délais de transmission. En effet indépendamment de la taille des données transmises, le délai peut varier d’une manière très aléatoire pour certains protocoles de transmission (par exemple Internet). Les travaux de Kosuge et son équipe ont montré (Kosuge *et al.*, 1996) que ce problème entraîne une instabilité même si les variations de délai sont relativement petites.

Dans les systèmes de téléopération originels, l’opérateur doit impérativement subir un apprentissage qui peut être long avant d’obtenir une bonne adaptation et une bonne maîtrise du système de téléopération (Zak et Das, 1995). En plus l’adaptation obtenue

peut ne pas être adéquate pour une autre architecture. La sophistication et le nombre important d'applications téléopérées engendre une augmentation de la charge de travail de l'opérateur, sa fatigue et ce qui entraîne une diminution de ses performances et augmente ainsi les risques d'erreurs de téléopération.

1.3.1.2 Evolution des systèmes de téléopération

L'évolution de l'informatique et des supports de télécommunication numérique est à l'origine de l'évolution de la téléopération. Grâce à eux, les robots esclaves ont pu être déportés à des distances considérables (Vertut et Coiffet, 1984). En effet, initialement l'opérateur commandait le robot esclave dans un niveau assez bas, les trajectoires issues des actions désirées sont envoyées directement via le dispositif maître. Le retour d'information (Images, force de contact, etc) se fait directement vers l'opérateur. Mais malheureusement, cette approche originelle n'a pas duré très longtemps, car les chercheurs se sont vite rendu compte de la nécessité d'exploiter les possibilités que peuvent offrir des ordinateurs pour apporter une assistance à l'opérateur. Depuis, plusieurs approches et concepts ont été proposés pour améliorer et faire évoluer les systèmes de téléopération. Nous résumons ici un classement des différentes approches représentant les modes d'interaction Homme-Machine que l'on retrouve généralement dans la plupart des systèmes de téléopération.

❶ La Téléopération Assistée par Ordinateur - TAO

La TAO (Vertut et Coiffet, 1984), est vue comme une voie intermédiaire entre la téléopération originelle bas niveau et la supervision. L'objectif de la TAO est de réaliser à chaque instant et pour toute étape de la tâche, une exploitation optimale des ressources d'un ordinateur. Tout se fait dans la partie maître, le robot reste esclave des actions de l'opérateur aidé par des programmes informatiques. La TAO s'oriente de plus en plus vers le développement des interfaces homme-machines pour la programmation hors ligne, l'entraînement de l'opérateur à la désignation de missions téléopérées etc. Depuis peu, la TAO bénéficie d'un atout nouveau qui est *la Réalité Virtuelle (RV)* (Burdea *et al.*, 1992). Un paragraphe concernant l'utilisation de cette technologie est présenté dans la suite de ce chapitre.

❷ La Téléopération à Désignation d'Objectif (supervisée) - TDO

Dans ce mode, l'opérateur est plutôt vu comme un superviseur (Sheridan, 1992). L'intervention de l'opérateur se limite dans ce cas à la désignation d'objectifs qui seront réalisés par le robot. Par exemple, l'opérateur peut désigner sur écran un objet à atteindre par le robot. Un autre exemple consiste à désigner dans un haut niveau une tâche que doit réaliser le robot (saisir objet, déposer objet, etc.). Ce concept vise une exploitation optimale du côté du robot.

❸ La Téléopération Semi autonome - TSA

La plupart des systèmes de téléopération actuels, s'orientent vers l'utilisation des deux concepts précédents (TAO et TDO). En effet ils tentent de moderniser la téléopération par une meilleure exploitation simultanée de l'autonomie du robot et

des capacités de l'opérateur. Certains auteurs parlent de la Téléopération à commande partagée (Sheridan, 1992), (Hayati et Venkataraman, 1989), (Bejczy *et al.*, 1990), (Kim *et al.*, 1992) et (Michelman et Allen, 1994). La commande partagée (*shared control*) est référencée dans un bon nombre de schémas de commande des systèmes de téléopération. La plupart des travaux se dirigent vers une approche plutôt unifiée (Conway *et al.*, 1990), (Graves et Volz, 1995), (Tarn *et al.*, 1995) ou encore (Spenny et Schneider, 1997). Ces derniers proposent des méthodes que l'on peut modifier dynamiquement afin de générer une variété de modes de commandes.

Nous pouvons résumer dans la figure 1.3, le classement de ces trois concepts selon le degré d'autonomie du dispositif esclave .

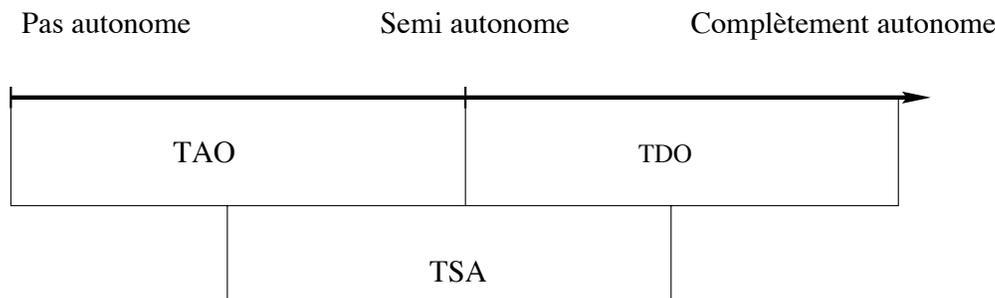


FIG. 1.3 – Classement de la TAO, TDO et TSA selon le degré d'autonomie du dispositif esclave.

1.3.2 La Télérobotique

La télérobotique est une forme de téléopération lorsque l'opérateur réalise des tâches à distance en utilisant un robot. Ce dernier peut fonctionner d'une façon autonome.

La télérobotique résulte en fait de la fusion des deux domaines originellement séparés qui sont la *téléopération* et la *robotique*. En effet, la robotique autonome n'étant pas encore tout à fait au point, le robot doit présentement être opéré à distance par un opérateur humain. On doit donc tenir compte des principes développés en téléopération. Cependant, comme le robot peut exécuter des tâches élémentaires de façon autonome, on parle de télérobotique plutôt que de téléopération.

La télérobotique trouve des applications partout où l'homme a des difficultés à travailler directement (milieu hostile, lointain, trop grand ou trop petit) et où les tâches sont suffisamment complexes ou imprévisibles pour rendre difficile une automatisation complète. La condition principale de développement de la télérobotique est sa capacité à concurrencer l'intervention directe d'un homme ou l'utilisation d'un système automatique très spécialisé. Dans le premier cas, l'atout de la télérobotique est tout d'abord le remplacement d'un travail humain pénible ou dangereux par un autre, plus sûr et confortable. Dans le second cas il faut montrer l'intérêt d'un matériel plus versatile que le système automatique dédié à l'application envisagée.

Pour répondre à la question pourquoi la télérobotique, Sheridan du MIT (Sheridan, 1992) fournit les raisons suivantes :

- ① Pour améliorer la performance et la fiabilité dans l'accomplissement des tâches,
- ② Pour améliorer la sécurité de l'opérateur,
- ③ Pour réduire le travail humain.

1.3.2.1 Quelques domaines d'application

① *Le domaine nucléaire*

Le nucléaire a été le premier domaine à stimuler le développement de systèmes de télérobotique. L'industrie nucléaire s'intéresse généralement aux applications suivantes :

- **La manipulation de produits radioactifs :** Elle se fait en cellule spécialisée, dans laquelle des télémanipulateurs mécaniques sont utilisés pour amener et retirer le produit. L'opérateur travaille en vision directe et il n'y a pas de problèmes d'éloignement (figure 1.4). L'introduction de la télérobotique n'est pas encore envisagée dans ce type d'application.
- **La maintenance des installations :** Ce type de tâche est essentiellement réalisée par des machines spécialisées. Ces dernières offrent l'avantage d'être parfaitement adaptées à la tâche. Par contre leur mise en oeuvre est longue et elles ne sont pas adaptées aux changements de situation et ne peuvent être réutilisées dans d'autres tâches. La télérobotique est en train de trouver sa place suite aux recherches menées ces dernières années.
- **Le démantèlement d'installation et l'intervention suite à un incident :** L'utilisation des systèmes télérobotiques est une voie en cours de recherche concernant ces deux domaines. L'exploitation des nouvelles technologies comme la *Réalité Virtuelle / Augmentée* est un atout favorable pour la poursuite de ces recherches.



FIG. 1.4 – Exemple du système Poste de Travail Mobile (PTM) du CEA piloté par un contrôleur TAO 2000

② *Le domaine spatial*

Les principales applications de la télérobotique dans ce domaine sont l'exploration et la maintenance de satellites (figure 1.5). L'éloignement du site de travail et l'important temps de transmission des informations, impliquent l'utilisation de robots ayant une grande autonomie (basée sur une architecture complexe et intégrant de nombreux capteurs) sans pour autant exclure la possibilité de les téléopérer (la mission du *sojourner* en 1997 sur la planète mars en est un exemple).

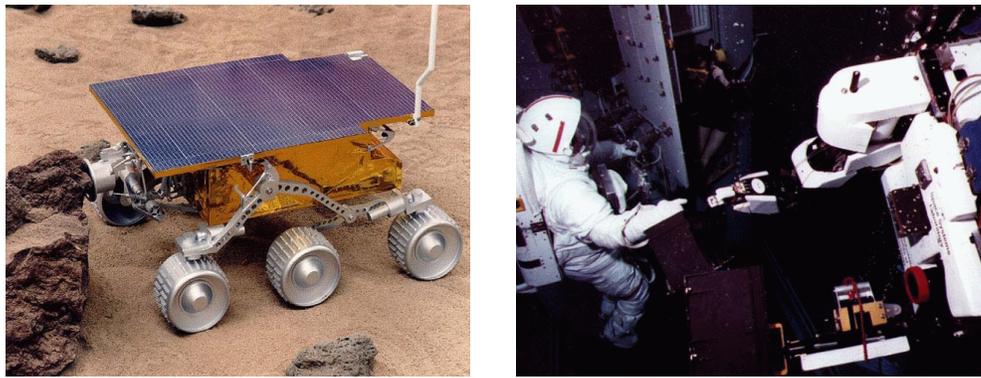


FIG. 1.5 – Exemples de robots : A gauche, l'exploration de Mars par "sojourner",    droite , la maintenance de satellites

3 Le domaine sous-marin

A l'exception des applications militaires, les applications civiles sont principalement li  es    l'industrie de l'offshore, l'inspection, la construction et la maintenance de conduites, structures, câbles sous-marins voire aussi des investigations scientifiques (  paves, esp  ces marines, etc.). Des bras manipulateurs embarqu  s    bord de v  hicules peuvent   tre autonomes ou m  me t  l  op  r  s (figure 1.6). Ce domaine continue      tre le principal domaine d'activit   en nombre de syst  mes de t  l  robotique.



FIG. 1.6 – Exemple d'utilisation d'un robot embarqu   sur un v  hicule sous marin JASON ((Sayers et al., 1995))

4 Le domaine militaire

La t  l  robotique intervient pour la d  sactivation de mines, l'observation de territoires ennemis ainsi que la t  l  commande d'engins tels que les chars, les avions et les h  licopt  res.

5 L'aide aux handicap  s

Ce domaine est extr  mement riche de possibilit  s. De nombreux projets ont   t   tudi  s et mis en   uvre afin d'aider des handicap  s    mieux vivre. On peut citer par exemple le projets SPARTACUS (Galerie, 1989) et le projet MASTER du CEA (Cammoun et al., 1994) ont permis l'automatisation de quelques t  ches quo-

tidiennes à partir d'une commande utilisant les mobilités disponibles de l'handicapé.

⑥ *Le domaine médical*

Les médecins et les chirurgiens, utilisent de plus en plus de robots afin de les assister ou même de les remplacer dans certaines tâches. La téléchirurgie (figure 1.7) semble avoir un avenir prometteur, en particulier dans la chirurgie oculaire, qui réclame une grande précision et une sécurité extrême (Grace *et al.*, 1993) (Hunter *et al.*, 1994). D'autres applications peuvent être trouvées en chirurgie mini invasive (Laparoscopie et coelioscopie).

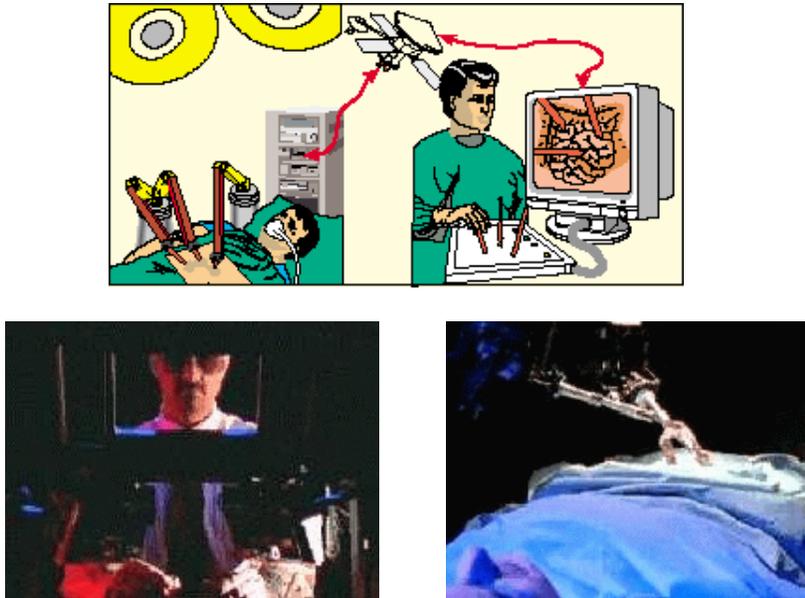


FIG. 1.7 – *Illustration de la téléchirurgie*

Nous noterons que l'évolution des systèmes de téléopération est étroitement liée à l'évolution des technologies d'interactions homme-machine. Nous nous intéressons tout particulièrement aux technologies de la *Réalité Virtuelle (RV)* et de la *Réalité Augmentée (RA)* qui ont été rapidement adoptées au profit de la téléopération. Bien que le terme de RA est considéré comme une méthode de la RV, nous avons préféré les séparer afin de montrer les similitudes et les différences qui existent entre ces deux concepts. Ce sont des techniques qui ont permis d'une part, de faire face au problème de délai de transmission, et d'autre part de fournir une meilleure interaction homme-machine.

1.3.3 La réalité virtuelle et la réalité augmentée : Outils indispensables pour le télétravail

1.3.3.1 La réalité virtuelle : Bref Etat de l'art

Bien que ce mot ne soit à la mode que depuis quelques années, la réalité virtuelle est née de recherches qui ont débuté dans les années cinquante dans des milieux aussi divers que les laboratoires de la NASA ou les studios d'Hollywood. Les pionniers de ce monde artificiel furent des visionnaires qui empruntèrent des techniques aussi variées que

le cinéma, l'informatique, l'automatique et l'électronique, les briques de ce qui fût appelé la Réalité Virtuelle (RV).

Il existe autant de définitions qu'il y a de champs d'application à cette technologie, en plus, cette technologie est plutôt vue comme quelque chose de magique, aux possibilités quasi illimitées, la réalité étant tout autre chose. Les contraintes auxquelles doit faire face cette technologie sont nombreuses. Ce qui a provoqué de multiples définitions de la RV. Nous avons retenu deux définitions qui à nos yeux semblent plus explicites. La première est donnée par Burdea et Coiffet (Burdea, 1993) concernant le terme RV et la deuxième est plus générale, elle introduit la notion d'Environnement Virtuel (EV) retenu notamment par Thibout (Thibout, 1996) et dont la RV constitue une partie. Pour ce qui est des interfaces de la réalité virtuelle, nous recommandons la référence (Fuchs, 1996).

Définition de la RV :

Un système de réalité virtuelle est une interface qui implique la simulation en temps réel et des interactions via de multiples canaux sensoriels. Ces canaux sensoriels sont ceux de l'homme, vision, audition, toucher odorat et le goût.

Définition de l'EV :

L'environnement virtuel est constitué d'un ensemble de techniques permettant de reproduire le plus fidèlement, par calcul, le comportement d'entités 3D. Ces techniques se décomposent en deux classes :

❶ L'interface homme-machine (interaction avec l'EV)

- ☛ Perception visuelle, sonore, tactile.
- ☛ Acquisition de données (gant, souris 6D, etc.)
- ☛ Communication accrue entre l'homme et la machine (multimodalité)

❷ Le comportement des entités de l'EV

- ☛ Modèles cinématique, dynamique
- ☛ Gestion des collisions
- ☛ Modèles d'éclairage
- ☛ etc...

Si l'on regarde les types d'applications utilisant la réalité virtuelle, on se rend compte que le divertissement, les arts et l'éducation dominent ce marché. L'autre secteur important concerne la haute technologie comme l'aérospatial, la robotique, la médecine, les sciences ou le domaine militaire. On peut considérer que tous les domaines de la vie courante peuvent et pourront un jour être touchés par la RV.

Heslser et Doherty⁶, ont classé les domaines en fonction du nombre possible d'applications.

6. <http://brahma.imag.fr/Multimedia/DESSGI/Realite-Virtuelle/html/applications.html>

Domaine	Nombre d'application
Simulation	73
Visualisation	67
Education	66
Apprentissage	65
Divertissement	65
Art graphique	65
Armement	52
Aérospatial	50
Téléprésence	50
Médecine	49
Architecture	46
Audio-visuel	41
Affaires	40
Télérobotique	39
Communications	38

TAB. 1.2 – *Etude marketing des applications de RV*

1.3.3.2 Quelques domaines d'application de la RV

- ❶ **L'Architecture :** On peut créer des plans, imaginer des architectures de construction de cités, de villes, enfin grâce au virtuel on peut faire visiter une maison avant qu'elle ne soit construite. Ce procédé est surtout utilisé pour les handicapés. A l'aide d'une chaise roulante virtuelle, on peut vérifier s'il y aura un problème lors des déplacements à l'intérieur du bâtiment en question.
- ❷ **L'armée :** Les militaires utilisent la RV pour plusieurs raisons comme : la simulation. Grâce à la RV, l'armée peut entraîner ses régiments pour des missions spécifiques ou bien pour les entraîner de façon générale. Nous avons eu la preuve de l'efficacité de la RV lors de la Guerre du Golf. Son efficacité a été clairement mise en évidence, puisque les pilotes ont pu s'entraîner durant des mois sur des simulateurs virtuels avec des données qui provenaient directement des satellites espions américains. A leur arrivée sur le champs de bataille, ils en connaissaient déjà tous les recoins.
- ❸ **Le divertissement :** C'est sans doute dans ce secteur que la RV est la plus exploitée à son état pur. Des jeux très réels dans un monde imaginaire où tout est permis. Les rêves les plus fous y sont autorisés. La majorité des jeux virtuels qui existent sont basés sur le principe de la simulation. Être dans un véhicule et le diriger au travers de différents environnements, en ressentir toutes les secousses. Les jeux virtuels font de plus en plus fureur.
- ❹ **L'enseignement :** L'apprentissage à l'aide des hypermédia entraîne une navigation non linéaire et interactive par l'usage d'un matériel éducatif qui atteint les sens de l'étudiant : vision, audition, toucher, odorat. Les ordinateurs pour les hypermédia, connectés aux réseaux internationaux à large bande, réalisent la plupart de l'enseignement interactif, plus spécialement l'enseignement des détails techniques dans les domaines de l'art, des affaires, de l'histoire, des langues, de la médecine, de la musique, des sciences. Les instructeurs donneront moins d'explications, parce que

les meilleurs matériaux des meilleurs enseignants du monde sur presque n'importe quel sujet seront ajoutés aux présentations hypermédia de son propre professeur et sont disponibles sur simple appel à n'importe quel moment du jour et de la nuit quand les étudiants veulent apprendre.

⑤ **La télérobotique :** Utilisée pour des interventions dans des milieux hostiles tels que : l'intérieur d'une centrale nucléaire, les fonds sous-marin, l'intérieur d'un volcan... c'est d'ailleurs ce principe qui a été utilisé lors des récentes missions spatiales sur la planète Mars. Grâce à la RV les techniciens peuvent s'entraîner à manipuler le robot, améliorer leurs réflexes en causant des fausses situations d'urgence ou de faux problèmes techniques. Ils peuvent aussi enregistrer une série de mouvements ou de tâches, que le robot devra faire, avant de les faire faire par le robot dans une situation réelle. Dans une opération de télérobotique la plupart des actions du robot sont préprogrammées. Lorsque le robot effectue ces phases préprogrammées, l'opérateur ne croise pas les doigts. Il doit superviser le robot durant l'accomplissement de sa tâche ou, dans certains cas, il doit assister le robot (téléassistance). Les techniques de RV en robotique sont susceptibles de remplir deux fonctions. La première est la simulation, qui permet à l'opérateur de s'entraîner (à manipuler le robot) et de programmer des tâches qui seront ensuite exécutées par le robot. La deuxième fonction est de servir d'interface entre l'opérateur et le robot au cours d'une mission, par laquelle l'opérateur peut manipuler le robot et l'assister.

⑥ **La chirurgie :** En chirurgie, la tendance est de réduire la taille de l'accès à la zone à opérer. L'objectif de cette tendance est de réduire le risque d'infection, la morbidité postopératoire et bien sûr le temps d'hospitalisation. Ces techniques se nomment vidéoscopie et endoscopie. Ces deux techniques fonctionnent très bien, mais la visibilité du chirurgien et l'accessibilité à la zone à opérer sont fortement réduites. Grâce à la RV, ce problème peut être enfin résolu. Ces techniques consistent à introduire une micro-caméra, qui transmet les images à un écran installé près de la table d'opération, par une petite incision et par une autre, introduire les instruments de chirurgie. Tout ceci pour effectuer l'acte chirurgical. La visibilité étant restreinte et le chirurgien étant privé du contact avec les organes, l'opération devient très difficile. À l'aide d'échographie, de scanographie ou d'imagerie par résonance magnétique nucléaire, il est possible de reconstituer la zone à opérer en une image tridimensionnelle visible à l'aide d'un système de RV. Le chirurgien a donc une meilleure visibilité et puisqu'il peut grossir l'image qu'il reçoit, il est même plus précis. Il faut comprendre que ce procédé est à un stade peu développé. Par ce fait, il ne peut pas être utilisé en pratique, il faut attendre encore quelques années avant que ce procédé n'atteigne sa maturité. Cette technique ne servira pas seulement à réduire le risque d'infection, la morbidité postopératoire, le temps d'hospitalisation et à augmenter la précision de l'acte chirurgical et la visibilité. Mais elle permettra à de jeunes chirurgiens de s'entraîner sur des modèles virtuels avant de pratiquer des opérations sur de vrais sujets.

La NASA s'intéresse à la téléchirurgie, pour permettre l'opération de quelqu'un dans l'espace. Le chirurgien opérerait sur un modèle virtuel, et un robot assistant opérerait réellement le patient. Les capteurs d'efforts devraient, dans ce cas, fournir et faire ressentir, les plus légers efforts appliqués par les outils chirurgicaux du robot. Nous ne sommes pas encore rendu à faire de la téléchirurgie à l'aide de la télérobotique, mais ce procédé est en cours de conception et de développement.

Il faut noter qu'au début de l'apparition de la réalité virtuelle, on pensait que tous les domaines allaient être touchés et que beaucoup d'applications basées sur ce principe verraient le jour. Aujourd'hui, la tendance est de ne pas réaliser des applications purement virtuelles, mais, avec les méthodes de ces dernières, d'augmenter la perception des utilisateurs. Ce nouveau concept se nomme la réalité augmentée. Dans ce cas, l'utilisateur n'est plus complètement immergé dans un monde non réel, mais des informations virtuelles, viennent renforcer sa perception de la réalité. La définition de la réalité virtuelle n'étant pas encore unanime, il est difficile de dire si la réalité augmentée est ou non de la réalité virtuelle.

1.3.3.3 La réalité augmentée :

La Réalité Augmentée (RA) peut être définie comme une combinaison de la vraie scène visualisée par l'utilisateur et d'une scène virtuelle produite par l'ordinateur. D'une manière générale cette technique consiste à augmenter la scène réelle avec des informations virtuelles supplémentaires. Cette augmentation peut prendre différentes formes selon les domaines d'application comme nous allons le voir par la suite. Par exemple dans le domaine de la télérobotique, généralement cette opération consiste à superposer un modèle virtuel sur une image réelle grâce à un mécanisme appelé la *calibration* (cf. § 3.2.2). On parlera alors de la *visualisation prédictive* ou encore des *prédicteurs graphiques* utilisés généralement en téléopération soit pour pallier le problème de délai (Noyes et Sheridan, 1984), (Bejczy *et al.*, 1990) et (Bejczy et Kim, 1990), soit pour rehausser l'image vidéo pour compenser ses défauts (Mallam *et al.*, 1992), figure 1.8) et (Oyama *et al.*, 1992). Des exemples de systèmes à base de cette technologie sont présentés à la fin de cette partie.

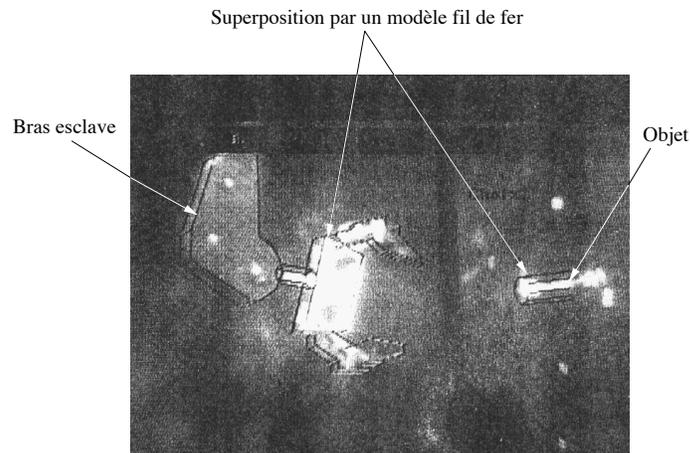


FIG. 1.8 – Rehaussement d'images vidéo dégradées par superposition d'une image synthétique "fil de fer" à une image caméra

1.3.3.4 Quelques domaines d'application de la RA

L'évolution de la technologie d'affichage et des systèmes informatiques, ainsi que le traitement temps réel d'images vidéo a rendu possible l'affichage d'images virtuelles cor-

rectement appariées aux images vidéo. La réalité augmentée a trouvé sa place et a apporté des solutions à des problèmes rencontrés dans de nombreux domaines.

- ① **Médical :** L'utilisation des systèmes de la RA dans ce domaine est une voie très encouragée par de nombreux chercheurs. Généralement ce sont les opérations chirurgicales qui en profitent le plus. Par exemple, un robot est utilisé pour assister le neurochirurgien lors d'une opération chirurgicale sur un patient atteint d'une tumeur au cerveau. Des images sont transmises à l'ordinateur qui les traduit en 3D et durant l'opération, le cerveau virtuel est en permanence superposé à l'image du cerveau réel du patient. Cette superposition permet au système de guider le chirurgien jusqu'à la tumeur moyennant un rayon laser. Quatre systèmes de ce genre existent aux Etats Unis et deux en France dont un à l'hôpital Necker (Paris).
- ② **Divertissement :** Une forme simple de la RA est utilisée pendant le journal de télévision. La plupart du temps le journaliste se trouve devant un fond bleu ou vert et cette image réelle est augmentée avec des images générées par ordinateur.
- ③ **Formation militaire :** Les militaires utilisent aussi des systèmes de RA où des informations s'affichent sur leur écran de pilotage, ou bien un viseur indiquant la cible à atteindre. On trouve un système distribué de simulation de guerre SIMNET qui utilise aussi des techniques de la RA (Urban, 1995).
- ④ **Conception, fabrication, entretien et réparation :** Quand les techniciens de maintenance se trouvent en face d'une nouvelle pièce qui ne leur est pas familière, au lieu de consulter de nombreux manuels, il peuvent utiliser un système de RA où l'affichage de l'équipement (pièce) peut être augmenté avec des informations pertinentes pour la réparation. Par exemple les composantes qui sont défectueuses vont clignoter sur l'écran facilitant ainsi la localisation donc la réparation (Feiner *et al.*, 1993), (Uenohara et Kanade, 1995). Les chercheurs de chez Boeing développent un système de RA permettant de remplacer le gros travail destiné à la réalisation des installations électriques pour leur avion (Caudell, 1994). En utilisant ce système expérimental les techniciens sont guidés par l'affichage augmenté qui leur montre l'acheminement des câbles.
- ⑤ **Robotique et Télérobotique :** Comme nous l'avons mentionné au début de cette section, la RA est utilisée pour améliorer la perception visuelle ou encore anticiper une situation réelle et prévenir des cas dangereux. En effet, la superposition d'un modèle virtuel sur une image réelle est très utilisée en Téléopération. Le fait d'augmenter l'image vidéo avec un modèle en fil de fer par exemple, facilite la visualisation du monde 3D distant (généralement lorsque le site distant est visualisé par une seule caméra). Si l'opérateur désire effectuer un déplacement sur le robot réel, il peut le réaliser sur le robot virtuel (fil de fer) qui n'est rien d'autre qu'un rehaussement graphique du vrai robot. L'opérateur peut alors décider de l'exécution de la tâche après avoir vu le résultat de la simulation, ainsi les problèmes d'une opération de télérobotique liés à la distance séparant les deux sites sont éliminés (problème de délai, image bruitée, etc ...). La téléopération via un système de RA offre plusieurs avantages :

Avantage des systèmes de RA :

- ✓ **Délai invariant :** *L'opérateur reçoit un contrôle visuel depuis l'image graphique et non pas du site distant. Ainsi, les performances de l'opérateur ne sont pas affectées par le temps de réponse incertain (délai).*
- ✓ **Affichage rehaussé :** *l'image vidéo est augmentée par la présence d'informations abstraites comme les distances entre objets du site distant, l'environnement du robot et l'affichage des trajectoires générées.*
- ✓ **Erreurs de manipulation réduites :** *La tâche de simulation réduit les erreurs de collision, les erreurs d'inattention et prévient aussi de certaines configurations indésirables.*
- ✓ **Niveau de contrôle très élevé :** *L'opérateur se trouve à l'extérieur de la boucle spatio-temporelle. Il contrôle le robot virtuel sur la machine locale, qui envoie ensuite des instructions de haut niveau à la machine distante.*
- ✓ **Modélisation partielle de l'environnement de la tâche :** *Des modèles partiels d'objets qui ont un rapport avec la tâche à exécuter peuvent être construits interactivement, on peut donc utiliser certaines procédures comme la détection de collisions par exemple.*

1.3.3.5 Quelques exemples de systèmes de téléopération utilisant la RV et/ou la RA

❶ Jet Propulsion Laboratory (JPL USA)

Cette équipe travaille sur un système permettant à l'opérateur de ne plus être tributaire du temps de propagation entre le site esclave et le site maître (Fiorini *et al.*, 1992), (Kim, 1993). L'opérateur dispose dans le site maître d'une interface lui permettant de configurer le système, de gérer l'acquisition des données pertinentes et de visualiser en temps réel les forces et couples s'exerçant sur le bras. Une interface vidéo permet de gérer le retour d'information des caméras ainsi que la superposition à ces dernières des images synthétiques (prédicteurs graphiques FIGURE 1.9), dont certaines issues d'une base de données géométriques mise à jour en temps réel. Pour s'affranchir des temps de transfert des informations, l'image prédite du déroulement de la tâche est présentée à l'opérateur et recalée sur l'image réelle. La principale aide apportée à l'opérateur est visuelle et lui permet d'avoir une estimation de la distance et de l'orientation de l'effecteur par rapport à la cible.

❷ Electro Technical Laboratory (ETL Japon)

Cette équipe réalise un système téléopéré pour l'assemblage et le désassemblage de pièces mécaniques avec supervision en vision synthétique (Hasegawa, 1991). Un module permet la modélisation interactive d'objets en utilisant la coopération d'une caméra et d'un télémètre laser. Un module de programmation permet, à l'aide d'un système de visualisation, de superposer l'image synthétique des objets modélisés à l'image vidéo. Enfin, un module d'exécution gère le manipulateur, le contrôleur de

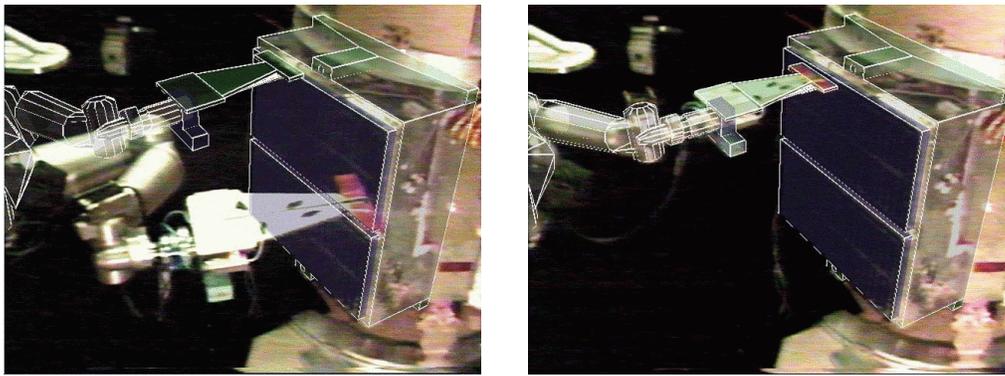


FIG. 1.9 – Utilisation de la prédiction graphique pour la téléopération spatiale. À gauche prédiction et confirmation de la tâche, à droite exécution

tâche ainsi que le transformateur de données.

③ Ergonomics in Teleoperation and Control Laboratory (ETCL Université de Toronto, Canada)

Dans ce laboratoire, a été réalisé le système ARTEMIS (Augmented Reality Tele-Manipulation Interface System) (Rastogi *et al.*, 1996) (FIGURE 1.10). Ce système permet de générer les modèles de l'environnement distant en utilisant la stéréovision. Il permet de superposer le modèle stéréographique du robot sur les images stéréovidéo du site distant, le modèle stéréographique du robot est utilisé pour simuler les tâches de manipulation et des instructions sont ensuite envoyées au robot réel pour l'exécution.

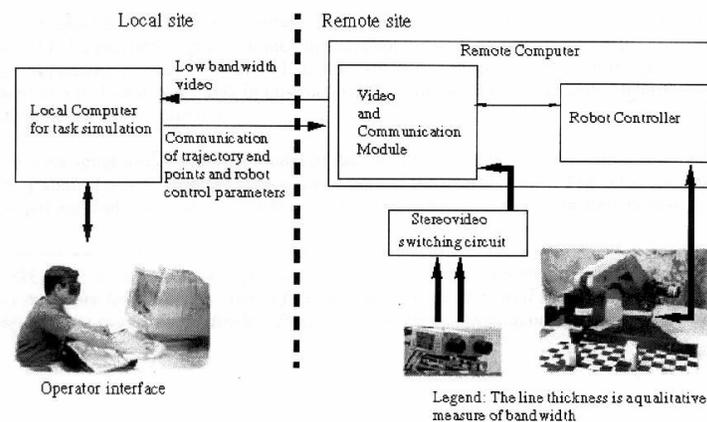


FIG. 1.10 – Architecture du système ARTEMIS

④ Commissariat à l'Énergie Atomique de Fontenay aux Roses (CEA, France)

Un système de modélisation 3D **PYRAMIDE** (Bonneau et Even, 1993) a été développé au CEA. Trois sources d'informations coopèrent pour réaliser le système:

- ◆ La base de données géométrique 3D de l'environnement constitué d'une librairie d'objets pouvant être rencontrés.

- ◆ Les images que fournissent les caméras embarquées sur un robot mobile.
- ◆ Les informations que peut fournir l'opérateur humain.

Ces informations sont ensuite retournées à l'opérateur sous forme d'images synthétiques superposées à l'image vidéo pour l'assister dans la programmation, l'exécution et la supervision des tâches.

⑤ Virtual Tools and Robotics Laboratory (Pennsylvania State University, USA)

Les travaux réalisés par cette équipe (Wang *et al.*, 1996) proposent une interface homme-machine utilisant des concepts de la réalité virtuelle et un système d'évitement d'obstacles afin de planifier les trajectoires d'un télémanipulateur en vue de la saisie d'objets. Le modèle virtuel d'une pince est superposé au départ à son image réelle. L'opérateur peut alors tester les configurations de saisie possibles à l'aide du modèle virtuel. Ceci permet de choisir un chemin de saisie optimum (sans collisions) et l'appliquer au bras réel.

⑥ Laboratoire de Robotique de Paris (LRP France)

Le concept du robot caché est utilisé pour la téléopération d'un ou plusieurs robots (Kheddar, 1997) (FIGURE 1.11). Dans ce cas l'environnement réel est modélisé et le site maître est découplé du site esclave. l'architecture utilisée est basée sur une Représentation Fonctionnelle Intermédiaire (RFI) au moyen de laquelle l'opérateur agit directement sur la tâche. L'état désiré du robot réel est déduit à partir de cette tâche. Cet état est adressé comme consigne au site esclave.

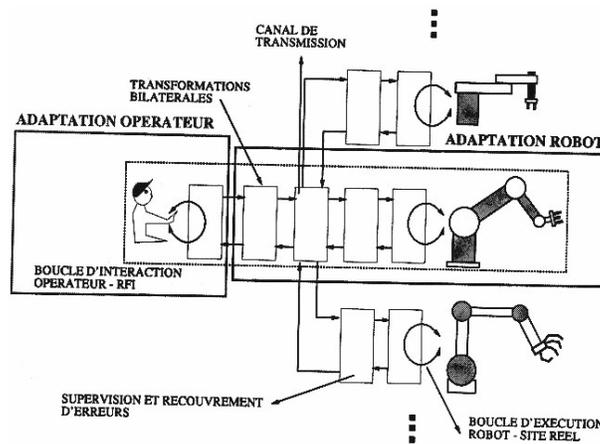


FIG. 1.11 – schéma général du système de téléopération proposé par Kheddar

⑦ Laboratoire Systèmes Complexes (LSC, France)

Dans notre laboratoire LSC du CEMIF (Centre d'Etudes de Mécanique d'Ile de France) à été développé le système MCIT (Multimedia Control Interface in Teleoperation) (Mallem *et al.*, 1993), (Loukil, 1993), (Chekhar, 1994). Ce système procure à l'opérateur une assistance à la perception et à la commande. Un module de reconnaissance de polyèdres à partir d'une images de luminance a été développé et intégré à MCIT (Shaheen, 1999). Il permet la mise à jour de la base de données des objets pré-modélisés de l'environnement. La superposition des objets reconnus sur

l'image de la scène apporte une aide visuelle à la perception. Ce système automatise des tâches fastidieuses imposées à l'opérateur comme par exemple l'appariement des primitives des objets vus par la caméra avec celles des modèles de la base de données 3D pour les identifier et les localiser.

1.4 Internet et les environnements télérobotiques : Bref état de l'art

Les systèmes de télérobotique sont traditionnellement implantés en utilisant des canaux de communication dédiés. Le déplacement du site esclave et/ou du site maître nécessite le déplacement du matériel ainsi que la reconfiguration réseau du canal de communication reliant ces deux sites. Les personnes sont obligées de se déplacer sur le site client (centre de contrôle afin de travailler et de préparer les missions). De plus si la machine cliente tombe en panne il devient difficile ou très coûteux de poursuivre la mission puisqu'il s'agit d'une machine dédiée où se trouve l'interface homme machine.

Cependant la télérobotique basée sur Internet, permet une délocalisation facile des opérateurs à faible coût, et ne dépend pas de l'endroit où se trouve le matériel qui permet de contrôler le robot qui, lui, se trouve sur le site esclave généralement distant. Cette idée a été exploitée par la NASA lors de la préparation de la mission *sojourner* sur Mars. Initialement les scientifiques étaient obligés de se déplacer jusqu'au centre de contrôle en *Californie* afin de travailler et de contrôler le robot, sachant que la communication Terre-Mars se fait avec une bande passante très limitée. Le développement d'une interface Internet (Backes, WWW), a donné la possibilité aux scientifiques de travailler et de collaborer ensemble pour contrôler le robot de n'importe où dans le monde (Backes *et al.*, 1998). Cet exemple montre une des possibilités que peut offrir le télétravail via Internet.

Les premiers sites Internet de la télérobotique sur le Web ont été créés en 1994 par l'américain Goldberg (Goldberg *et al.*, 1995) et l'Australien *Telerobot on the Web* (Taylor et Trevelyan, 1995). Depuis de nombreux sites Internet pour la robotique sur le web ont vu le jour, et actuellement le site de la NASA (NASA Space Telerobotics Program)⁷ regroupe 23 sites web y compris le nôtre (système ARITI)⁸. Dans ce qui suit, nous allons décrire brièvement quelques systèmes utilisant Internet comme moyen de contrôler des robots.

1.4.1 Australia's Telerobot on the Web

Lieu et auteurs : Australie, (Taylor et Trevelyan, 1995).

Type de robot : Un bras manipulateur ASEA IRB à 6 degrés de liberté (ddl).

Applications : manipulation d'objets.

7. http://ranier.oact.hq.nasa.gov/telerobotics_page/realrobots.html

8. <http://lsc.cemif.univ-evry.fr:8080/Projets/ARITI/>

Performances :

- l'utilisateur envoie la position et l'orientation de la pince axe par axe pour la placer à côté de l'objet. Il peut contrôler l'ouverture et la fermeture de la pince.
- Deux caméras vidéo sont utilisées pour permettre à l'utilisateur de bien placer la pince et de manipuler l'objet.

Limites :

- Il est difficile de positionner la pince et de manipuler l'objet. Ceci nécessite une grande concentration et un bon entraînement de la part de l'utilisateur.
- Le retour vidéo se fait à chaque exécution d'une tâche (pas de capture et de transfert d'images avant l'exécution d'une tâche). Le temps minimum de réponse du système est entre 15 et 20 secondes.
- Le système ne supporte pas plusieurs superviseurs (d'autres utilisateurs ne peuvent pas voir ce qui se passe avec le robot tant qu'ils ne le contrôlent pas).
- La pince peut rentrer en collision avec n'importe quel objet (aucune sécurité à ce niveau), ce qui oblige à fixer la vitesse de déplacement du robot 100mm/sec.

1.4.2 Mercury Project

Lieu et auteurs : USA - Nevada, (Goldberg *et al.*, 1995)

Type de robot : Un robot mobile.

Applications : Téléjardinage.

Performances :

- Le robot mobile est autonome et peut être téléopéré. Sa mission est d'arroser des plantes.
- Une caméra vidéo embarquée sur le robot permet de visualiser ce que fait le robot

Limites :

- Pas de retour vidéo pendant le déplacement du robot.
- Pas d'autres superviseurs possibles lorsque le système est en cours d'utilisation.

1.4.3 KheperOnTheWeb

Lieu et auteurs : EPFL - Suisse, (Saucy et Mondala, 1998)

Type de robot : Un robot mobile miniature, type Khepera .

Applications : Navigation avec évitement d'obstacles.

Performances :

- l'utilisateur spécifie la position et la vitesse du robot et les envoie ensuite pour l'exécution.
- Le système utilise deux caméras vidéo (une externe et l'autre embarquée sur le robot).
- L'utilisateur peut contrôler la caméra externe (orientation et zoom).
- Les images vidéo envoyées sont de type JPEG.

Limites :

- Le serveur d'images utilise une technique de compression qui est supporté uniquement par le navigateur " Netscape ".

- Un seul retour vidéo est possible à un instant donné (caméra externe ou embarquée).
- Le système tourne sous un système d'exploitation Windows 95 qui n'est pas stable (n'est pas fait pour ce genre d'application), ce qui oblige à redémarrer le serveur à chaque fois.
- Pas d'autres superviseurs possibles lorsque le système est en cours d'utilisation.

1.4.4 Xavier

Lieu et auteurs : USA - Pittsburgh, (Simmons, 1998)

Type de robot : Un robot mobile.

Applications : Navigation avec évitement d'obstacles.

Performances :

- Le robot est autonome et capable de recevoir des commandes vocales.
- Une caméra embarquée sur le robot permet de visualiser ce que voit le robot.
- L'utilisateur choisi des tâches de haut niveau prédéfinies par le système.

Limites :

- Le système retourne des informations de position et d'orientation toutes les 5 à 10 secondes ainsi qu'une image vidéo (de type GIF) toutes les 20 secondes.
- Le système supporte un seul client (utilisateur) à chaque fois.

1.4.5 RHINO

Lieu et auteurs : Allemagne, (Schulz *et al.*, 1998)

Type de robot : Un robot mobile.

Applications : Navigation avec évitement d'obstacles.

Performances :

- Le robot est autonome.
- L'interface du système utilise la réalité virtuelle (le robot est modélisé et l'environnement est semi-modélisé).
- Deux caméras réelles sont utilisées, une embarquée sur le robot permettant de visualiser ce que voit le robot et une autre caméra fixe qui visualise le robot dans son environnement.
- Deux caméras virtuelles sont utilisées de la même façon que les réelles. Elles permettent à l'utilisateur d'observer le déplacement du robot virtuel comme résultat de l'exécution de la tâche.

Limites :

- Pas de retour vidéo pendant le déplacement du robot.
- L'interface utilisateur (simulation virtuelle) est basé sur des bibliothèques "OpenGL" qui ne sont pas supportées par défaut par les navigateurs Internet.
- Le système supporte un seul utilisateur à chaque fois.

1.4.6 PumaPaint project

Lieu et auteurs : USA - Pennsylvanie, (Stein, 1998)

Type de robot : Un bras manipulateur PUMA 760.

Application : Peinture.

Performances :

- Le bras manipulateur est capable de reproduire un dessin réalisé par un utilisateur depuis une interface de dessin dédiée (qui ressemble à celle de “paintbrush sous windows”).
- Il existe 4 couleurs (rouge, vert, bleu et jaune) et un pinceau pour chaque couleur. Suivant la couleur et la trajectoire dessinée, le bras manipulateur va prendre le bon pinceau et reproduire la trajectoire réalisée par l'utilisateur.
- Après chaque tâche l'utilisateur peut visualiser le résultat en choisissant une des deux caméra dédiée pour ce retour vidéo.

Limites :

- Pas de retour vidéo pendant la réalisation de la tâche.
- Le système supporte un seul utilisateur à chaque fois.

1.5 Bilan

Ce premier chapitre nous a permis d'une part, de présenter les différents aspects du télétravail ainsi que les méthodes et les tendances technologiques qui contribuent énormément à son évolution. D'autre part, nous avons identifié les grandes lignes de notre problématique, à savoir :

- ❶ Quels sont les outils logiciels et matériels nécessaires pour la mise œuvre d'un système de télétravail?
- ❷ Comment intégrer les nouveaux concepts et les solutions déjà apportées aux problèmes liés à la télérobotique, sur une plate-forme universelle et à moindre coût?
- ❸ Comment rendre l'utilisation des systèmes de télétravail suffisamment souple et intuitive pour l'opérateur, et quels types d'assistances peut on lui apporter afin d'améliorer ses performances?
- ❹ Peut on construire une architecture client/serveur supportant un travail coopératif et sous quelles contraintes?
- ❺ Quels sont les avantages que peut fournir les systèmes de télétravail en particulier en utilisant le réseau Internet comme moyen de communication?

D'une manière précise, ce premier chapitre nous a permis, d'analyser les systèmes de téléopération et de téléprogrammation de robots existants, les méthodes et les techniques utilisées pour pallier les problèmes liés à la télérobotique en général et enfin d'analyser les récents travaux concernant le contrôle de systèmes robotiques via Internet.

Après ces nombreux précédents efforts, certains peuvent se demander, quelle est l'originalité du travail que nous proposons? Quelles sont les contributions scientifiques et techniques apportées à l'issue de ce travail?

Dans les chapitres qui vont suivre, nous tentons de répondre à ces deux inévitables questions ainsi qu'aux différentes problématiques posées.

Dans le chapitre suivant, nous proposons quelques outils d'assistance pour le t  l  travail. Ces outils sont bas  s sur une approche d'assistance graphique et b  n  ficient de l'avantage d'  tre param  trables. Ils sont g  n  ralement destin  s pour l'assistance    la t  l  op  ration de robots.

Chapitre 2

L'assistance graphique pour le Télétravail

Le premier chapitre nous a présenté la notion de télétravail en général et les outils technologiques dont dépend l'évolution de cette forme de travail à distance et en particulier l'outil Internet. Nous avons aussi présenté des cas particuliers de télétravail qui sont la télérobotique et plus particulièrement la téléopération, nous avons à cette occasion montré l'intérêt de la réalité virtuelle pour contourner certains problèmes comme les inévitables retards de communication par exemple. Nous avons cependant dégagé certaines problématiques auxquelles nous nous attachons dans cette thèse.

Dans ce chapitre, nous proposons des solutions aux problèmes d'assistance à l'opérateur pour le télétravail, en particulier l'assistance à la téléopération d'un robot.

Dans la première section de ce chapitre, nous présentons un bref état de l'art sur les métaphores graphiques. Nous verrons comment elles sont utilisées suivant les auteurs et suivant les champs d'application.

Dans la deuxième section, nous présentons les différentes méthodes, géométriques et analytiques utilisées pour la construction, la représentation et la manipulation des objets virtuels en général. D'autre part, nous verrons comment ces objets virtuels sont transformés en de véritables outils d'assistance pour la téléopération via Internet.

Afin de rendre ces outils facilement configurables par un opérateur un formalisme est proposé et implanté. La structure générale de ce formalisme est présentée à la fin de ce chapitre.

2.1 Les métaphores graphiques : Bref état de l'art

Parmi les apports de la réalité virtuelle à la téléopération, nous pouvons citer les métaphores graphiques connues sous le nom des guides virtuels de l'anglais *virtual fixtures* (Rosenberg, 1992) (Rosenberg, 1993), ou encore les mécanismes virtuels (Brooks et Ince,

1992) voir aussi (Fraisie, 1994) (Joly et Andriot, 1995) (Kosuge *et al.*, 1995) pour la commande en force d'un robot. Ces accessoires virtuels facilitent les d placements ou les saisies d'objets dans les phases de mod lisation ou de g n ration de trajectoires qui s'appliqueront ensuite au robot.

2.1.1 La m taphore des m canismes virtuels

Les m canismes virtuels peuvent  tre consid r s comme un ensemble d' l ments simples tels qu'une compliance, une rotation, une composante de glissement, etc. de telle sorte que les contraintes impos es au mouvement satisfassent la t che   r aliser. Les avantages d'un contr leur bas  sur les m canismes virtuels sont principalement les suivants:

- Le m canisme exprime une relation directe entre la trajectoire d sir e du manipulateur et les forces externes qui lui sont appliqu es par l'environnement virtuel.
- Par un m canisme virtuel on peut d finir un syst me de coordonn es qui convient spontan ment   la t che   laquelle il sera attribu .
- En utilisant un m canisme virtuel passif, on peut  tablir une commande stable si l'op rateur et l'environnement virtuel sont passifs.

Dans (Kosuge *et al.*, 1995), un exemple d'utilisation de m canismes virtuels pour la t l manipulation est donn  par la FIGURE 2.1

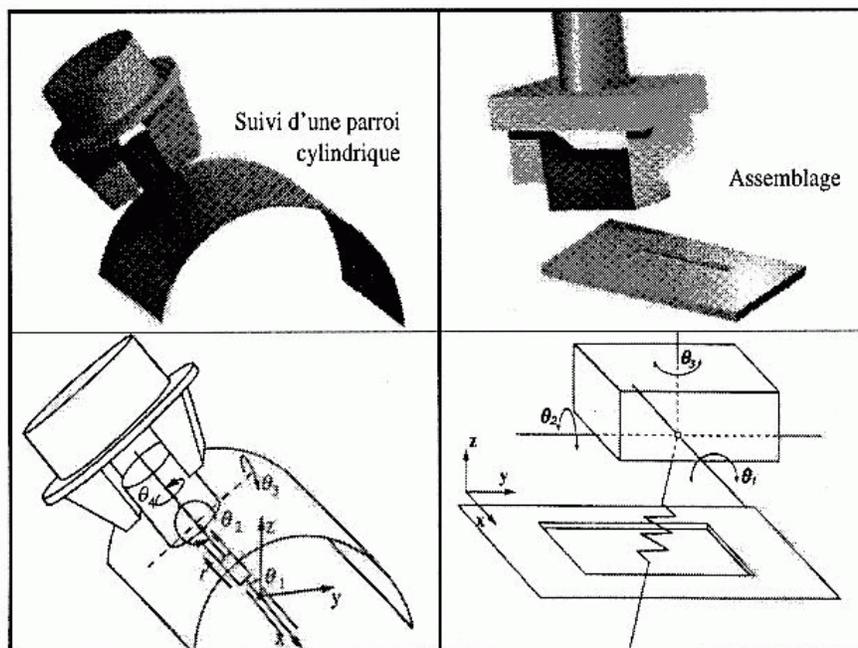


FIG. 2.1 – les m canismes virtuels pour la t l manipulation: A gauche, la structure avec les coordonn es cylindriques (x, r, θ_1) et l'imp dance m canique attach e au bout de l'outil $(\theta_2, \theta_3, \theta_4)$ qui permet le suivi de surface. A droite, un m canisme virtuel associ    la t che d'assemblage de la bo te saisie par le robot.

Dans (Joly et Andriot, 1995), le principe des m canismes virtuels est appliqu    la fois au dispositif ma tre et au robot. Ainsi, le retour d'effort vers le dispositif ma tre et la commande du robot se font   partir du m canisme virtuel (voir figure FIGURE 2.2).

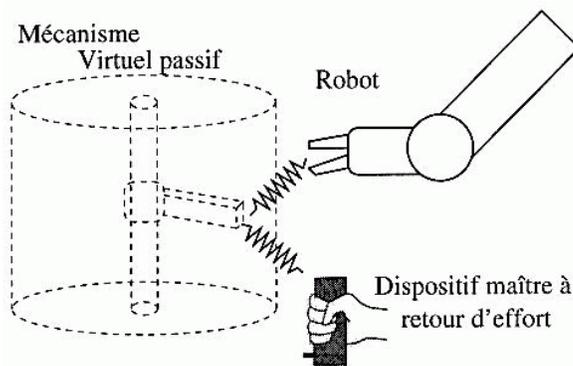


FIG. 2.2 – Téléopération basée sur le concept de mécanismes virtuels (d'après Joly et Andriot 95)

Dans une philosophie voisine des mécanismes virtuels, un nouveau paradigme est proposé dans [Backes et al. 96]. L'opérateur téléopère une trajectoire au lieu de téléopérer un robot. Le robot esclave sera alors contraint de suivre cette trajectoire durant ou après son établissement. Ces trajectoires sont nommées "chemins guides" ("motion guides"). Elles sont modifiables en ligne (durant la téléopération) ou hors ligne (durant la programmation de la trajectoire). Pendant que le robot suit une trajectoire, les commandes de l'opérateur sont du type: avant, arrière ou arrêt. Sur la trajectoire définie, des flèches visualisent le sens de déplacement du robot. Sur la trajectoire, on peut placer des tâches ("task lines") qui sont un traitement particulier à effectuer à un endroit précis du chemin guide. Le tout est implanté dans un environnement complètement virtuel. Les prédictors graphiques et la téléprogrammation intègrent aussi cet aspect dans l'interface homme - machine.

Dans (Thibout, 1996), la notion de mécanisme virtuel est utilisée pour définir des outils virtuels (FIGURE 2.3) utilisés dans un atelier virtuel. On parlera alors de la métaphore de la boîte à outils virtuels (FIGURE 2.4).

Le concept est très intéressant pour l'assistance en téléopération. Toutefois, il n'existe pas encore une méthodologie uniforme pour l'intégration de ce concept dans une interface de téléopération. La mise en œuvre paraît assez complexe, car les mécanismes virtuels sont conçus pour des tâches spécifiques.

2.1.2 La métaphore des guides virtuels

les recherches effectuées par Louis Rosenberg (Rosenberg, 1992) et (Rosenberg, 1993) à l'Université de Stanford, ont consisté à trouver un moyen permettant d'améliorer les performances d'un opérateur humain lors des tâches de télémanipulation avec délai. Il a introduit pour la première fois le concept de guides virtuels ("Virtual Fixtures") qu'il a ensuite expérimenté sur un système de téléprésence (figure 2.5). L'opérateur contrôle le robot réel via un exosquelette pour réaliser des tâches d'insertion (il s'agit de déplacer une tige depuis un trou vers un autre trou).

Les guides virtuels utilisés par le système pour assister l'opérateur dans sa tâche sont donnés dans la figure 2.6. L'opérateur peut sentir la présence de ces guides grâce à un retour haptique et/ou auditif.

Il a montré qu'en utilisant ces guides virtuels on peut améliorer les performances de ce type de tâche jusqu'à 70 %.

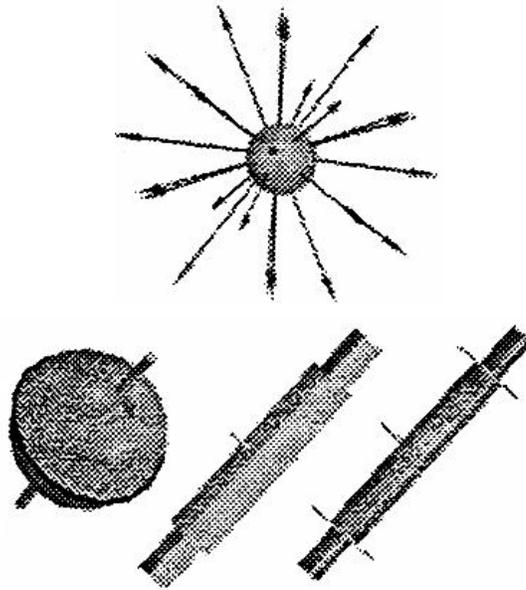


FIG. 2.3 – Exemple d'outils virtuels: en haut, des champs de force répulsifs pour l'évitement de zones dangereuses; en bas, des mécanismes virtuels à placer sur l'effecteur du robot pour guider ses mouvements (ici la rotule pour le blocage des translations, la glissière pour le suivi de droite et de pivot)

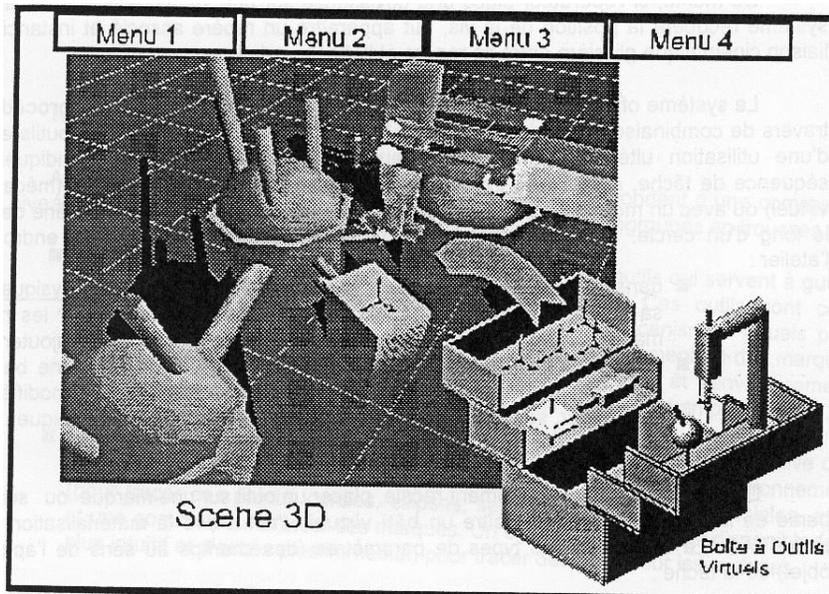


FIG. 2.4 – La métaphore de la boîte à outils virtuels

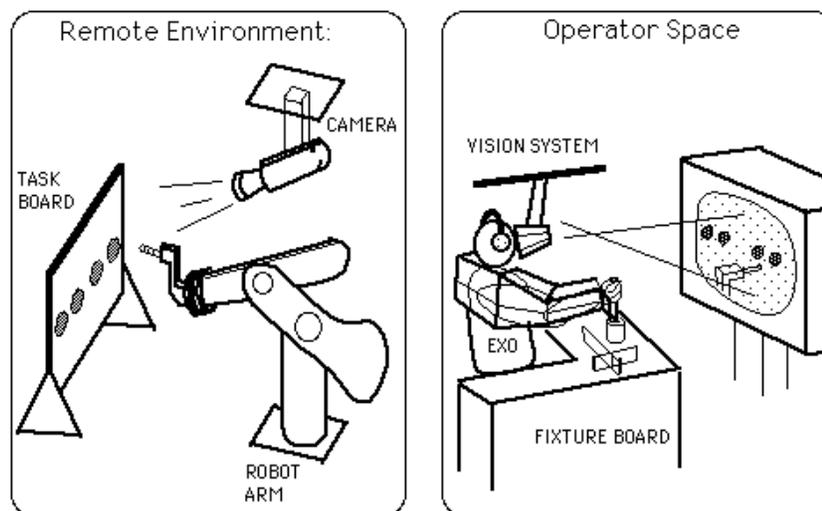


FIG. 2.5 – le système de téléprésence utilisé pour tester les performances de l'opérateur lors des tâches d'insertion avec ou sans l'aide des guides virtuels

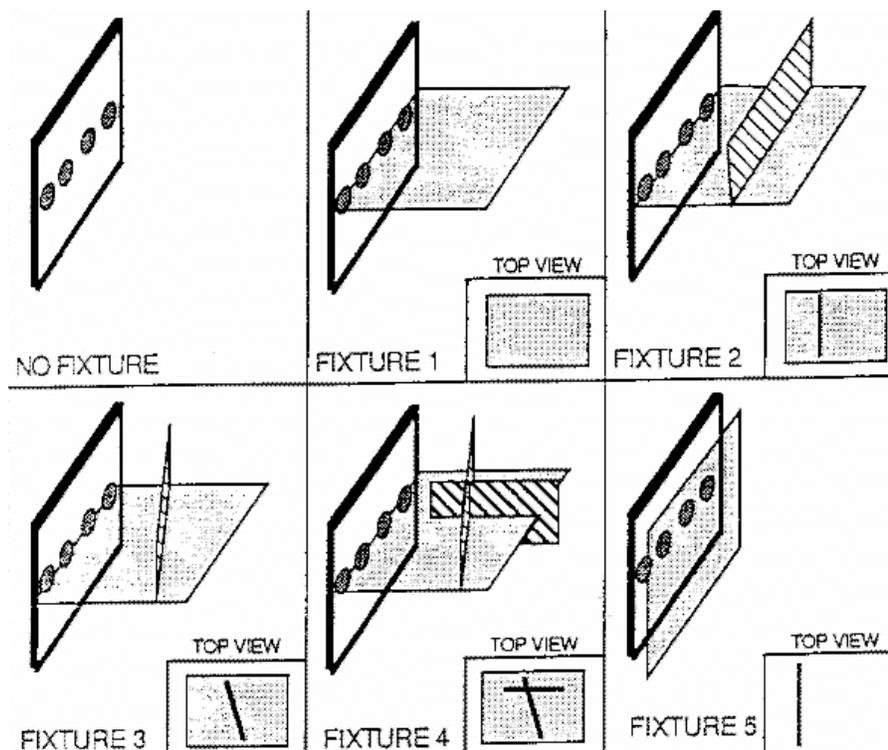


FIG. 2.6 – Exemples de guides virtuels projetés dans la zone de manipulation pour assister l'opérateur dans des tâches d'insertion d'après Rosenberg

D'autres recherches ont été effectuées dans le laboratoire GRASP (General Robotics and Active Sensory Perception) de l'université de Pennsylvanie (Sayers et Paul, 1994). L'opérateur travaille avec une représentation virtuelle du robot distant. Les actions réalisées dans ce monde virtuel sont observées ensuite envoyées sous forme de séquences d'instructions au robot réel. L'opérateur utilise un petit bras maître Puma 250 comme dispositif d'entrée/sortie (envoi des ordres au robot virtuel ou réception des retours d'efforts). Le retour visuel est fourni par la simulation virtuelle qui dans ce cas utilise le modèle virtuel du bras esclave Puma 560 (figure 2.7).

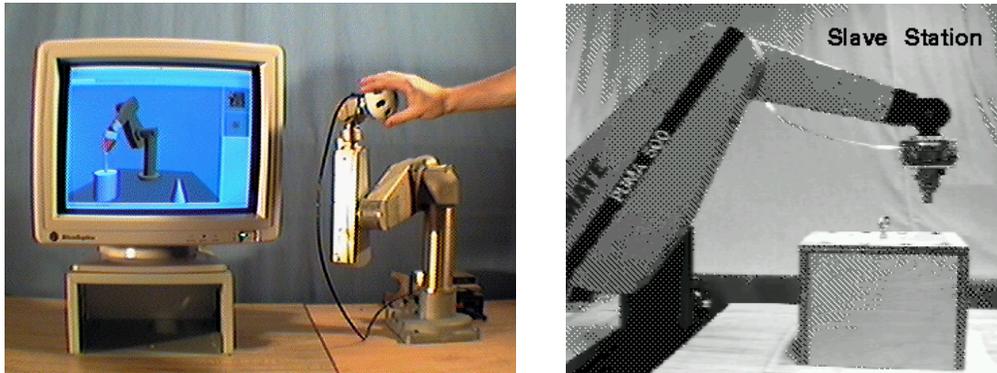


FIG. 2.7 – Le système de téléprogrammation utilisé par Sayers : A gauche, le site maître; A droite le site esclave.

La figure 2.8 montre quelques guides virtuels utilisés par ce système lors de l'expérimentation.

Une autre utilisation des guides virtuels pour une assistance à l'opérateur est donnée par Kheddar (Kheddar, 1997), voir la figure 2.9 ou encore la figure 2.10.

La notion de guides virtuels initialement introduite par Rosenberg pour assister l'opérateur, peut être décomposée en trois classes dont la classification est représentée par la figure 2.11 :

- ❶ **Les guides virtuels dédiés exclusivement à l'assistance de l'opérateur :** On y regroupera toutes les métaphores graphiques dont la fonction n'a pas de lien direct avec le robot esclave. A titre d'exemple des marqueurs de position, les divers substituts visuels ou autres dont le rôle sera orienté vers une assistance à l'accomplissement de la tâche virtuelle sans effet direct sur le robot.
- ❷ **Les guides virtuels dédiés exclusivement à la commande du robot :** On y regroupera les mécanismes virtuels ou tout autre artifice nécessaire à la stricte exécution d'une tâche.
- ❸ **Les guides virtuels partagés dédiés à la fois à l'assistance de l'opérateur et à une exploitation de l'autonomie des robots :** C'est une extension des *virtual fixtures* introduites par Rosenberg. Bien que le concept dans l'esprit de cet auteur s'applique à la téléopération pour faciliter le pilotage du robot esclave et qu'il soit déjà largement utilisé dans les systèmes de téléopération assistée par ordinateur (Coiffet et Gravez, 1991). Burdea et Coiffet (Burdea et Coiffet, 1994) proposent de l'exploiter sous une forme voisine pour le robot autonome. On peut répartir ces guides en trois sous classes:
 - ☛ **Les guides virtuels partagés à fonctions autonomes :** Cette catégorie de guides engendre une exécution autonome de la tâche virtuelle à la fois pour

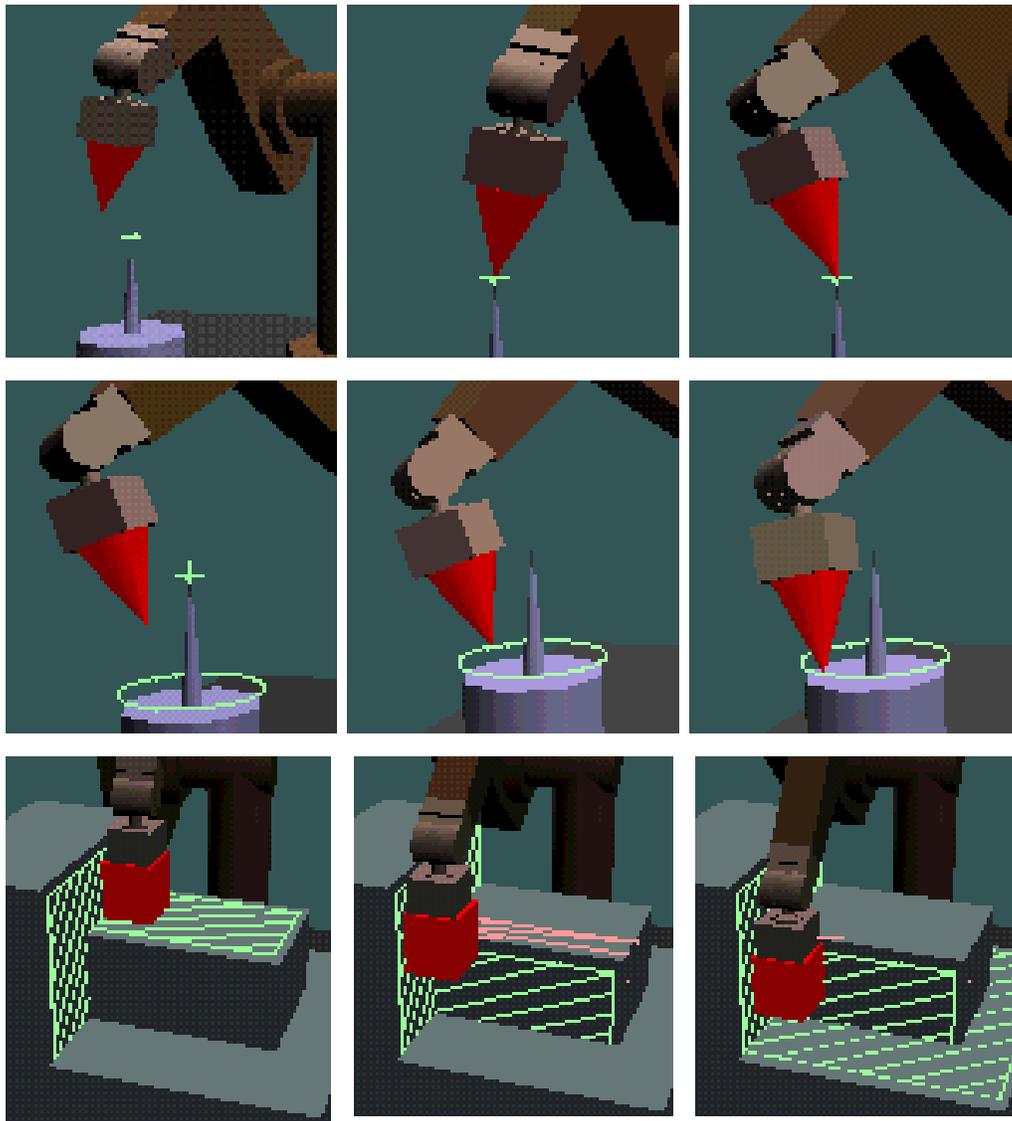


FIG. 2.8 – exemple de guides virtuels d'assistance à l'opérateur proposés par Sayers en 1994 : Première ligne, guide virtuel point-point pour déplacer la pointe de l'effecteur jusqu'à un point dans l'espace et ensuite effectuer des rotations; Deuxième ligne, pour le suivi de zones circulaires; Troisième ligne, pour effectuer un contact surface-surface.

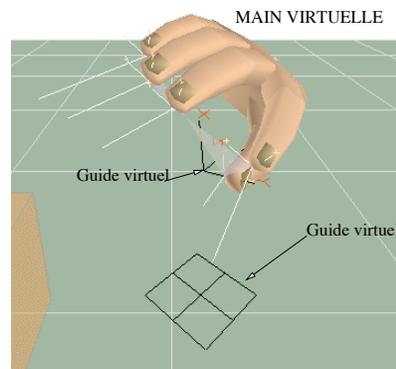


FIG. 2.9 – Guides virtuels dédiés exclusivement à l'assistance de l'opérateur. Une projection du repère du pouce sur le sol permet un guidage plus précis pour saisir ou manipuler les objets virtuels.

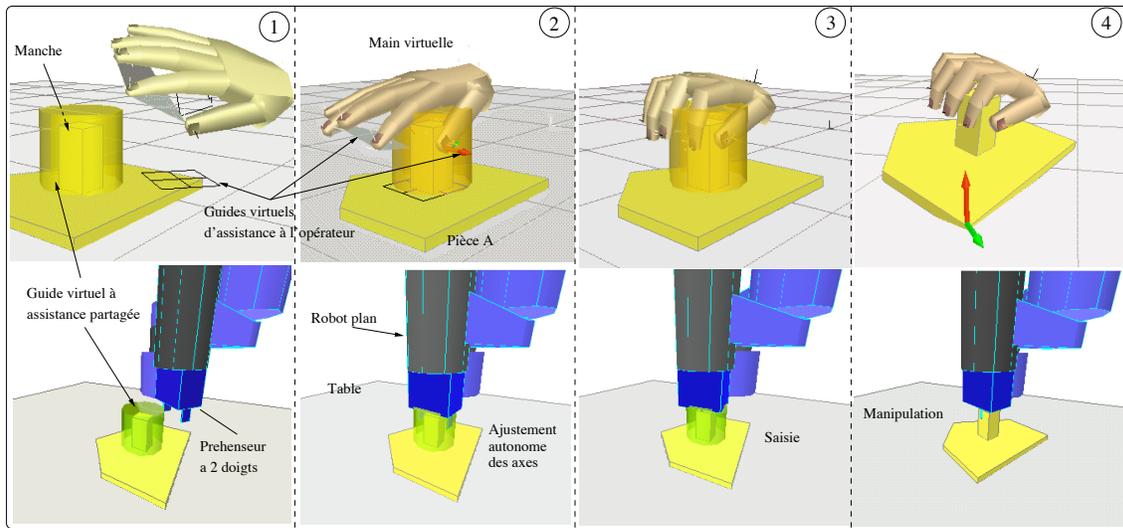


FIG. 2.10 – Exemple d'application des guides virtuels. Ajustement d'un préhenseur limité en capacité d'ouverture monté sur un robot plan de type SCARA. Cette figure illustre aussi les transformations entre l'action de l'opérateur et le robot intermédiaire de transformation.

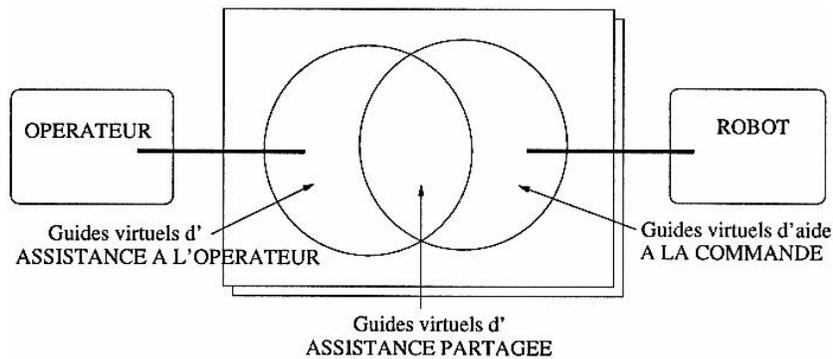


FIG. 2.11 – Classification de la métaphore des guides virtuels.

l'opérateur et pour le robot. L'exécution autonome est une aide à l'opérateur dans le sens où la tâche virtuelle se fait toute seule. Elle constitue en même temps une exploitation de l'autonomie du robot.

- **Les guides virtuels partagés à fonctions semi autonomes:** Dans ce cas l'opérateur effectue la tâche virtuelle sans assistance implicite. Le résultat de cette exécution engendre un fonctionnement autonome du côté du robot seulement.
- **Les guides virtuels partagés à fonctions de collaboration:** C'est une autre version des guides à fonctions semi autonomes. Dans ce cas l'interprétation de la métaphore donne lieu à un fonctionnement robotique issu d'une combinaison d'une fonction autonome exécutée par un robot virtuel et d'une fonction représentant l'action de l'opérateur.

2.2 Les guides virtuels pour l'assistance au télétravail via Internet

2.2.1 Les méthodes de construction des guides virtuels

Afin de permettre à l'utilisateur d'accomplir un grand nombre de tâches à l'aide des guides virtuels, nous lui proposons un large choix de guides :

- Différents types de surface, qui peuvent être utilisées pour ne pas franchir certaines limites (plan, disque, etc.)
- Des volumes ouverts, qui peuvent être traversés par le robot. (cube ouvert, cylindre, tube, etc.)
- Des volumes fermés, qui peuvent servir à définir le champs d'action du robot ou bien englober certaines parties de l'environnement dans le but de les rendre inaccessibles. (sphère, superquadriques, (Crespin, 1999), (Pira, 1998), (Blanc et Schlick, 1996) etc.)
- Des courbes qui peuvent servir de trajectoires au robot. (BSplines, (Peroche *et al.*, 1988) et (Foley *et al.*, 1990) etc.)

Ces guides sont construits par discrétisation de leur équation : on prend des points que l'on considère comme significatifs sur la surface du guide. Il suffit ensuite de relier ces points pour obtenir une représentation "fil de fer" du guide.

Si on possède une équation paramétrée du guide du type :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = f(\eta, \omega). \quad (2.1)$$

Les sommets du modèle filaire de l'objet seront les valeurs prises par la fonction f en des points équi-répartis sur son domaine de définition. Et les arêtes seront définies par l'ordre dans lequel les points ont été calculés.

Par exemple, pour une *superellipsoïde* (cf. § 2.2.1.2), on a :

$$f(\eta, \omega) = \begin{pmatrix} a_1 \cos(\eta)^{\varepsilon_1} \cos(\omega)^{\varepsilon_2} \\ a_2 \cos(\eta)^{\varepsilon_1} \sin(\omega)^{\varepsilon_2} \\ a_3 \sin(\eta)^{\varepsilon_1} \end{pmatrix} \quad (2.2)$$

où $(\eta, \omega) \in [0, \frac{\pi}{2}]$

Alors les sommets du modèle seront les points $M_{i,j}$ tels que :

$$M_{i,j} = f\left(i\frac{\pi}{2n}, j\frac{\pi}{2n}\right) \quad (2.3)$$

avec $(i, j) \in [0, n]$ et où n définit le pas de la discrétisation.

Les arêtes seront alors les couples $(M_{i,j}, M_{i+1, j})$ et les couples $(M_{i,j}, M_{i, j+1})$ (avec la convention : $n + 1 = 0$).

Dans ce qui suit nous présentons brièvement les théories inhérentes à la manipulation de la famille des superquadriques et des B-Splines avec pour objectif d'appliquer ces outils à la création de guides virtuels en 3 dimensions. Les superquadriques permettent de générer plusieurs types d'objets dans l'espace (sphère, cube, tore, etc.) à partir d'une même formule. Un autre point de vue pour construire des objets est celui des B-Splines : elles permettent de créer des objets modifiables localement.

2.2.1.1 Les Coniques et les superconiques

2.2.1.1.a Les coniques :

Les coniques sont une famille de courbes 2D dont l'équation générale peut se mettre sous la forme :

$$x^2(1 - e^2) + y^2 - 2px + p^2 = 0 \quad (2.4)$$

Elles se regroupent en trois sous-familles :

1. Si $e = 1$ L'équation devient

$$y^2 - 2px + p^2 = 0 \quad (2.5)$$

la courbe est nommée parabole FIGURE 2.12.

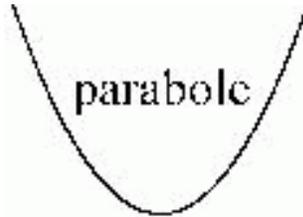


FIG. 2.12 – Exemple d'une parabole générée par la famille des coniques

2. Si $e \neq 1$ L'équation devient

$$y^2 = (e^2 - 1)\left(x - \frac{p}{1+e}\right)\left(x - \frac{p}{1-e}\right) \quad (2.6)$$

- (a) Si $e > 1$, la courbe est nommée hyperbole (FIGURE 2.13 a)
- (b) Si $e < 1$, la courbe est nommée ellipse (FIGURE 2.13 b)

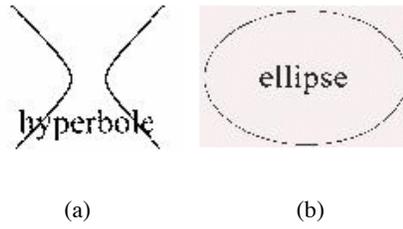


FIG. 2.13 – Exemple d'hyperboles et d'ellipse générées par la famille des coniques

2.2.1.1.b Les superconiques :

Une superconique est obtenue en appliquant une fonction de poids aux termes trigonométriques de l'équation polaire d'une conique. Notons f_p cette fonction de poids. Elle est définie par :

$$\forall t \in [-1, 1], f_p(t) = \text{sign}(t)|t|^p \quad (2.7)$$

où $p \in \mathbb{R}_+$

L'équation paramétrique polaire d'une ellipse est :

$$\forall \theta \in [-\pi, \pi], \begin{cases} x = a \cos(\theta) \\ y = b \sin(\theta) \end{cases} \quad (2.8)$$

On en déduit donc l'équation paramétrique polaire d'une superellipse :

$$\forall \theta \in [-\pi, \pi], \begin{cases} x = a f_p(\cos(\theta)) \\ y = b f_p(\sin(\theta)) \end{cases} \quad (2.9)$$

La FIGURE 2.14 montre quelques exemples obtenues par une superconique.

2.2.1.2 Les superquadrriques

Une superquadrrique est définie comme le produit¹de deux superconiques. On peut les regrouper en trois familles: les superellipsoïdes, les supertoroïdes et les superhyperboloïdes.

2.2.1.2.a Les superellipsoïdes :

Une superellipsoïde est le produit sphérique de deux superellipses de même centre. Voici donc l'équation en coordonnées polaires (η, ω) d'une superellipsoïde E :

$$E(\eta, \omega) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(\eta)^{\varepsilon_1} \\ a_3 \sin(\eta)^{\varepsilon_1} \end{pmatrix} \otimes \begin{pmatrix} a_1 \cos(\omega)^{\varepsilon_2} \\ a_2 \sin(\omega)^{\varepsilon_2} \end{pmatrix} \quad (2.10)$$

Soit : $E(\eta, \omega) = f(\eta, \omega)$ décrite dans l'équation 2.2.

Le domaine de définition D de l'équation en coordonnées polaires d'une super-ellipsoïde E peut être mis sous la forme :

1. Produit sphérique: si $p(u) = (x_p(u), y_p(u))$ et $q(v) = (x_q(v), y_q(v))$ sont deux courbes du plan, alors on définit le produit sphérique de p par q par la courbe de l'espace s telle que $s(u, v) = (x_p(u)x_q(v), x_p(u)y_q(v), y_p(u))$

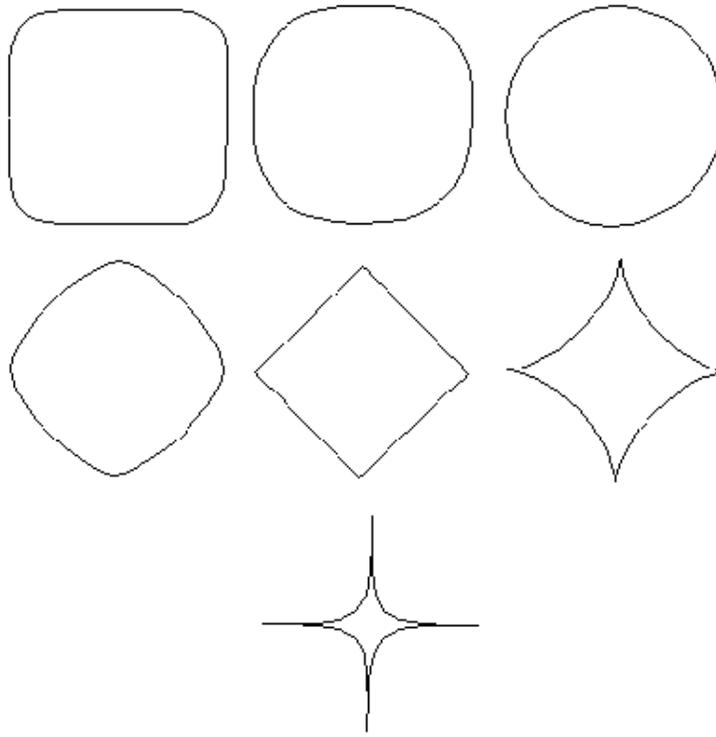


FIG. 2.14 – Exemple de superellipses, de haut en bas et de gauche à droite avec $p=0.38, 0.78, 1, 1.78, 2, 3$ et 6

$$D = D(\varepsilon_1) \times D(\varepsilon_2) \tag{2.11}$$

où :

$$D(\varepsilon_i) = \begin{cases} [0, 2\pi] & \text{si } \varepsilon \in \mathbb{N} \text{ avec } i = 1, 2 \\]0, \frac{\pi}{2}[& \text{sinon} \end{cases} \tag{2.12}$$

Les figures 2.15, 2.16 et 2.17 montrent quelques exemples de guides virtuels qui peuvent être obtenus avec une superellipsoïde.

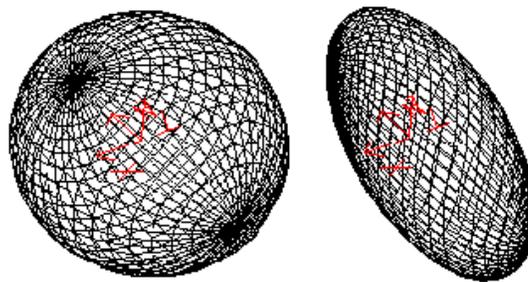


FIG. 2.15 – ellipsoïdes avec $\varepsilon_1 = \varepsilon_2 = 1, a_1 = a_2 = a_3$ et $\varepsilon_1 = \varepsilon_2 = 1, a_1 > a_2 > a_3$

2.2.1.2.b Les supertoroïdes :

Un supertoroïde est le produit sphérique de deux superellipses de centres distincts. Voici donc l'équation en coordonnées polaires (η, ω) d'un super-toroïde T :

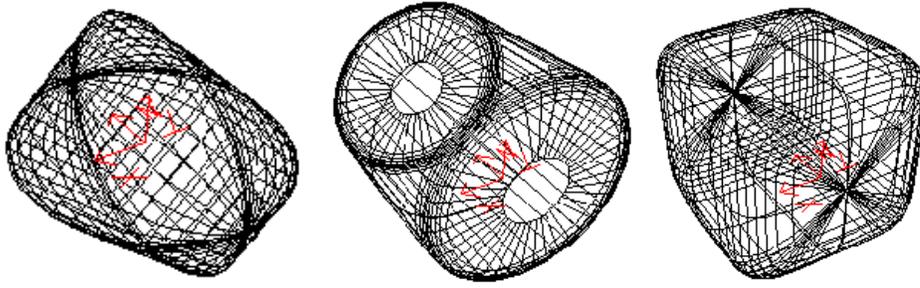


FIG. 2.16 – *ellipsoïdes avec* $\varepsilon_1 = 1, \varepsilon_2 = 2; \varepsilon_1 = 0.1, \varepsilon_2 = 1; \varepsilon_1 = \varepsilon_2 = 0.35$

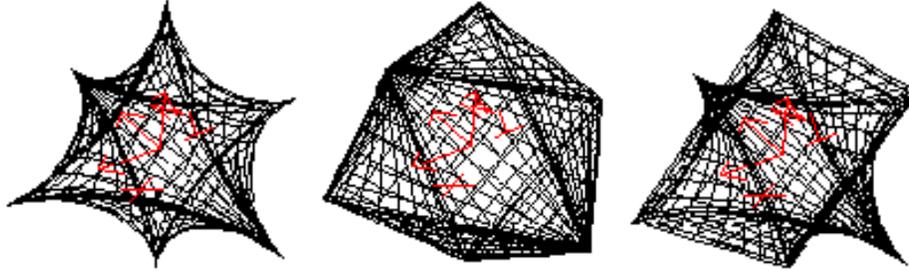


FIG. 2.17 – *ellipsoïdes avec* $\varepsilon_1 = \varepsilon_2 = 3; \varepsilon_1 = \varepsilon_2 = 2; \varepsilon_1 = 3, \varepsilon_2 = 2$

$$T(\eta, \omega) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_4 + \cos(\eta)^{\varepsilon_1} \\ a_3 \sin(\eta)^{\varepsilon_1} \end{pmatrix} \otimes \begin{pmatrix} a_1 \cos(\omega)^{\varepsilon_2} \\ a_2 \sin(\omega)^{\varepsilon_2} \end{pmatrix} \quad (2.13)$$

Soit :

$$T(\eta, \omega) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_1(a_4 + \cos(\eta)^{\varepsilon_1}) \cos(\omega)^{\varepsilon_2} \\ a_2(a_4 + \cos(\eta)^{\varepsilon_1}) \sin(\omega)^{\varepsilon_2} \\ a_3 \sin(\eta)^{\varepsilon_1} \end{pmatrix} \quad (2.14)$$

On remarque que le domaine de définition d'un supertoroïde est le même que celui d'une superellipsoïde, car en prenant $a_4 = 0$ dans l'équation paramétrique d'un supertoroïde, on obtient l'équation paramétrique d'une superellipsoïde.

Les figures 2.18 et 2.19 montrent quelques exemples de guides virtuels que l'on peut obtenir avec des supertoroïdes.

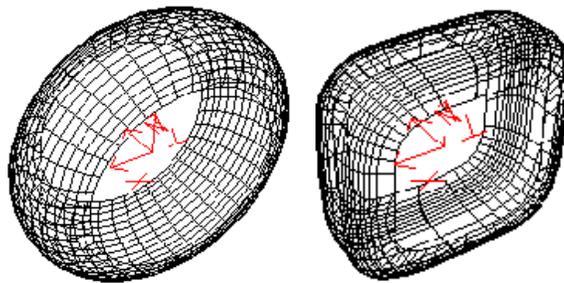


FIG. 2.18 – *supertoroïdes avec* $\varepsilon_1 = \varepsilon_2 = 1, \varepsilon_1 = \varepsilon_2 = 0.5$

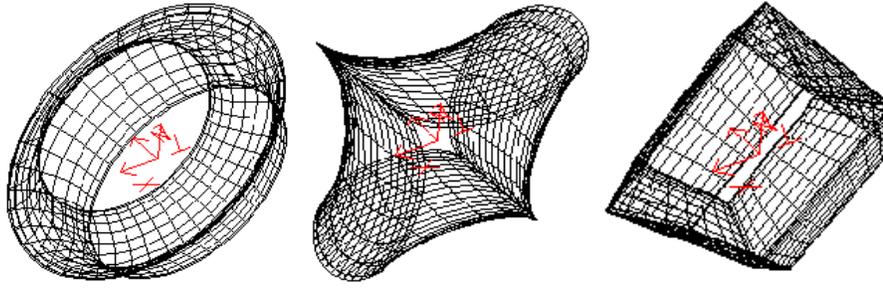


FIG. 2.19 – *supertoroïdes avec $\varepsilon_1 = 3, \varepsilon_2 = 1; \varepsilon_1 = 1, \varepsilon_2 = 3; \varepsilon_1 = \varepsilon_2 = 2$*

2.2.1.2.c Les superhyperboloïdes :

Les superhyperboloïdes sont une autre famille de superquadrique. Elles n'ont pas été implémentées dans l'application en raison de leur difficulté de représentation et du peu de type de figure qu'elles génèrent. Voici, à titre indicatif, les équations paramétriques des superhyperboloïdes à une et deux nappes.

– Les superhyperboloïdes à une nappe :

$$H_1(\eta, \omega) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_1 \frac{1}{\cos(\eta)^{\varepsilon_1}} \cos(\omega)^{\varepsilon_2} \\ a_2 \frac{1}{\cos(\eta)^{\varepsilon_1}} \sin(\omega)^{\varepsilon_2} \\ a_3 \tan(\eta)^{\varepsilon_1} \end{pmatrix} \quad (2.15)$$

– Les superhyperboloïdes à deux nappes :

$$H_2(\eta, \omega) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_1 \frac{1}{\cos(\eta)^{\varepsilon_1}} \frac{1}{\cos(\omega)^{\varepsilon_2}} \\ a_2 \frac{1}{\cos(\eta)^{\varepsilon_1}} \tan(\omega)^{\varepsilon_2} \\ a_3 \tan(\eta)^{\varepsilon_1} \end{pmatrix} \quad (2.16)$$

2.2.1.3 Les B-Splines

Nous souhaitons permettre à l'utilisateur de définir facilement une surface quelconque. Une première approche serait de lui faire entrer un échantillon de points appartenant à cette surface et de procéder par interpolation. Mais le nombre de points nécessaires à l'obtention d'un résultat convenable sera alors élevé. Une autre méthode consiste à procéder par lissage : les points fournis par l'utilisateur seront des points de contrôle qui n'appartiendront pas forcément à la courbe mais qui permettront de modifier la forme de celle-ci. Les courbes et surfaces de Bézier et des B-Splines sont basées sur un tel principe.

2.2.1.3.a Les courbes de Bézier :

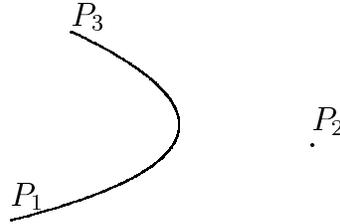
Si on note $(P_i)_{i \in [0, d]}$ les points de contrôle, la courbe de Bézier associée à ces points est définie par :

$$P(t) = \sum_{i=0}^d P_i B_{i,d}(t) \quad (2.17)$$

où :

$$B_{i,d}(t) = C_i^d t^i (1-t)^{d-i} \text{ avec } C_i^d = \frac{d!}{i!(d-i)!} \quad (2.18)$$

Voici un exemple de courbe de Bézier :



2.2.1.3.b Les surfaces de Bézier :

Les surfaces de Bézier peuvent se définir à partir des courbes de Bézier : si on note $(P_{i,j})_{i \in [0,n], j \in [0,m]}$ les points de contrôle, alors la surface de Bézier est définie par :

$$P(s, t) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} B_{i,n}(s) B_{j,m}(t) \quad (2.19)$$

2.2.1.3.c Les courbes B-splines :

Les courbes de Bézier présentent deux inconvénients :

- Le changement d'un seul point de contrôle modifie entièrement la courbe.
- Le degré du polynôme obtenu augmente avec le nombre de points de contrôle

Voici donc une autre méthode qui permet de modifier localement la courbe à l'aide des points de contrôle et d'avoir une complexité indépendante du nombre de points de contrôle.

Si on note $(S_i)_{i \in [0,d]}$ les points de contrôle, la courbe B-spline de degré k associée à ces points est définie par :

$$P(t) = \sum_{i=0}^d S_i N_{i,k}(t) \quad (2.20)$$

où :

- $N_{i,1}(u) = 1$ si $t_i \leq u \leq t_{i+1}$
- $N_{i,1}(u) = 0$ sinon
- Et pour $k \neq 0$, $N_{i,k}(u) = \frac{u-t_i}{t_{i+k-1}-t_i} N_{i,k-1}(u) + \frac{t_{i+k}-u}{t_{i+k}-t_{i+1}} N_{i+1,k-1}(u)$

Les $(t_i)_{i \in [0,d+k]}$ étant des réels quelconques. Ici, nous avons pris :

- $t_i = 0$ si $i < k$
- $t_i = i - k + 1$ si $k \leq i \leq d$
- $t_i = d - k + 1$ si $i \leq d + k$

2.2.1.3.d Les surfaces B-splines :

De manière analogue aux surfaces de Bézier, on définit une surface B-Spline ayant pour

points de contrôle les $(P_{i,j})_{i \in [0,n], j \in [0,m]}$ par :

$$P(s, t) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} N_{i,k}(s) N_{j,l}(t) \quad (2.21)$$

avec k et l les degrés des deux courbes B-Splines associées à leurs points de contrôle.

En fixant le nombre de points contrôle de la surface à 16 points, on peut expliciter matriciellement l'équation de la surface B-Spline :

$$\begin{cases} x(s, t) = S \cdot M_{Bs} \cdot G_{Bs_x} \cdot M_{Bs}^t \cdot T^t \\ y(s, t) = S \cdot M_{Bs} \cdot G_{Bs_y} \cdot M_{Bs}^t \cdot T^t \\ z(s, t) = S \cdot M_{Bs} \cdot G_{Bs_z} \cdot M_{Bs}^t \cdot T^t \end{cases} \quad (2.22)$$

Avec :

$$T = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \quad (2.23)$$

$$S = \begin{pmatrix} s^3 & s^2 & s & 1 \end{pmatrix} \quad (2.24)$$

$$M_{Bs} = \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \quad (2.25)$$

et

$$G_{Bs} = \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{pmatrix} \quad (2.26)$$

Dans le cas où l'on ne possède pas d'équation paramétrée du guide que l'on veut créer, on n'a pas de méthode générale et nous avons implanté des méthodes au cas par cas. Par exemple, pour un cylindre, nous discrétisons les deux extrémités (ouvertures) du cylindre par la méthode vue ci-dessus, puis nous relient les points qui se font face à chaque extrémité.

La figure 2.20 montre des exemples de guides virtuels obtenus par ces méthodes :

2.2.2 Représentation et manipulation des guides virtuels sur écran

Après avoir vu comment créer des guides virtuels, nous allons maintenant voir comment les représenter sur écran et permettre de les désigner et de les manipuler avec la souris. Nous présentons le modèle de projection des données 3D sur écran, nécessaire aux méthodes d'extraction de la 3D à partir de points 2D et celles de détection de collision entre le robot et son environnement, que nous décrivons par la suite.

2.2.2.1 Le modèle de la caméra graphique

Le modèle de la caméra graphique est une matrice M qui permet de transformer les coordonnées (x_o, y_o, z_o) d'un point dans le repère objet (R_o) (figure 2.21) en coordonnées u et v d'un point de l'écran. Ce modèle résulte du produit de plusieurs matrices : mise à l'échelle et translation d'origine, projection perspective et matrice de passage du repère

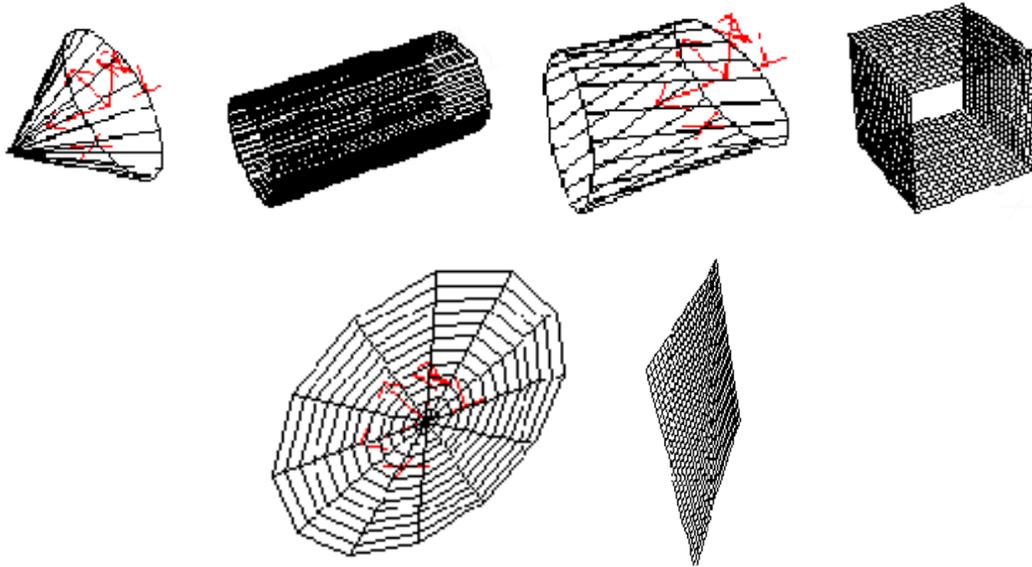


FIG. 2.20 – Cône, Cylindre, tube, cube, disque et plan

(R_o) au repère (R_c) (cf. § 3.2.2 pour une description détaillée concernant la modélisation de la caméra). Cette matrice est de la forme :

$$M = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix} \quad (2.27)$$

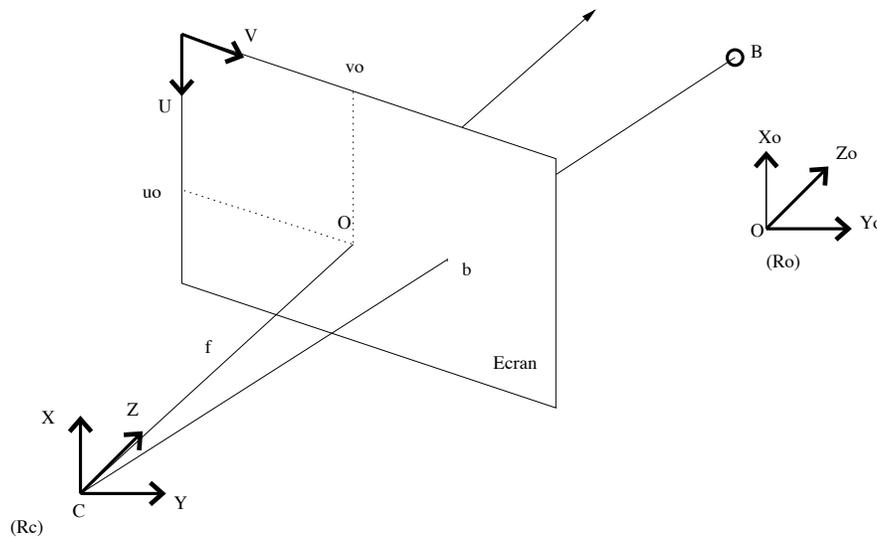


FIG. 2.21 – Représentation simplifiée de la projection perspective, f étant la focale.

Ainsi, un point (x_0, y_0, z_0) de l'espace est représenté sur l'écran par le point de coordonnées (u, v) telles que :

$$M \times \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \begin{bmatrix} s \times u \\ s \times v \\ s \end{bmatrix} \quad (2.28)$$

2.2.2.2 Sélection d'un objet de l'espace sur l'écran

L'utilisateur doit avoir la possibilité de manipuler ces objets virtuels (pour les placer convenablement, par exemple), il est nécessaire qu'il puisse les sélectionner en les désignant sur écran. Cette désignation (clic avec la souris par exemple) se traduit en coordonnées (u, v) du point de l'écran où ce clic a eu lieu. Il faut donc déterminer quel objet l'utilisateur a voulu désigner par ce clic. Les coordonnées (u, v) correspondent à une droite D , d'équation :

$$(x, y, z) \in D \iff \exists s, M \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} s \times u \\ s \times v \\ s \end{bmatrix} \quad (2.29)$$

Ensuite, il ne reste qu'à calculer quel objet possède le point qui est à la plus petite distance de cette droite. Par exemple dans la figure 2.22, c'est l'objet 3 qui a été sélectionné (le modèle 3D de cet objet étant connu).

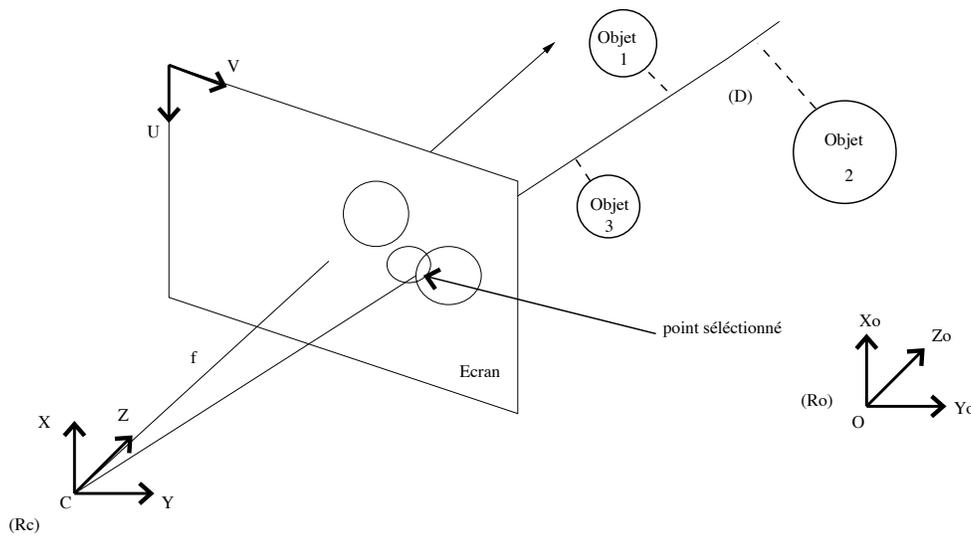


FIG. 2.22 – Exemple de sélection d'un objet virtuel sur écran

2.2.2.2.a Déformations des guides virtuels :

Les objets proposés à l'utilisateur sont des objets élémentaires (cylindre, plan, sphère, etc.) et il se peut que ces objets ne conviennent pas à l'environnement de travail, c'est pourquoi nous souhaitons permettre une déformation de ces objets afin que l'on puisse créer des guides qui s'adaptent à des environnements complexes. Le type de déformation qui convient le mieux à nos attentes est la déformation locale puisqu'elle permet de modifier le guide uniquement là où il y a problème. Le principe de cette déformation est de donner l'impression à l'utilisateur qu'il peut "étirer" un point de l'objet.

On peut représenter un objet par une liste ordonnée de ses sommets et une liste des couples de sommets qui forment une arête. Ainsi, ce cube de côté 1 mètre (figure 2.23) peut être représenté par :

$$(0, 0, 0), (0, 1, 0), (1, 1, 0), (1, 0, 0), (0, 0, 1), (0, 1, 1), (1, 1, 1), (1, 0, 1)$$

et

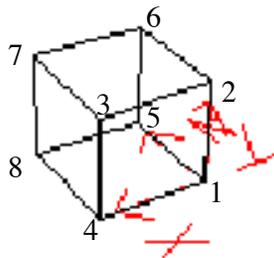


FIG. 2.23 – Exemple d'un cube avant déformation

(1, 2), (2, 3), (3, 4), (4, 1), (5, 6), (6, 7), (7, 8), (8, 5), (1, 5), (2, 6), (3, 7), (4, 8)

Ainsi, on peut associer à un objet un graphe $G = (X, U)$ où X est un ensemble dont les éléments sont appelés sommets et $U \subset X \times X$ est un ensemble dont les éléments sont appelés arcs. On peut alors définir l'application V qui à tout sommet x de X associe l'ensemble de ses voisins, c'est à dire :

$$V(x) = \{u \in X | (u, x) \in U \text{ ou } (x, u) \in U\}$$

On appellera, $d(x, y)$ la distance du sommet x au sommet y définie par :

$$d(x, y) = \min\{i \in \mathbb{N} | y \in V^i(x)\} \quad (2.30)$$

Si on appelle x_0 le point de départ de la déformation et Δ_0 la valeur de cette déformation, on a :

$$\forall x \in X | \Delta(x) = p f^{d(x, x_0) - 1} \Delta_0 \quad (2.31)$$

où p est appelé facteur de propagation initial et f facteur de dégradation de la propagation.

Exemple : Si on prend $x_0=1$, $\Delta_0 = (1, 0, 0)$, $p = 1$ et $f = 0.5$ alors :

$\Delta(2) = \Delta(4) = \Delta(5) = (1, 0, 0)$; $\Delta(3) = \Delta(6) = \Delta(8) = (0.5, 0, 0)$ et $\Delta(7) = (0.25, 0, 0)$

On obtient donc :

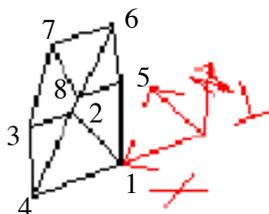


FIG. 2.24 – Résultat de la déformation sur le cube

Pour effectuer la déformation, nous avons vu que nous avons besoin d'un point initial (x_0), d'une déformation initiale (Δ_0), du facteur de propagation initial et du facteur de dégradation de la propagation. Les deux derniers facteurs peuvent être demandés à l'utilisateur à l'aide d'une interface. Mais, pour faciliter l'utilisation des déformations, nous souhaitons que l'utilisateur sélectionne avec la souris le point de l'objet à considérer comme initial, puis qu'il indique (sans relâcher le bouton de la souris) l'endroit où le point initial doit se trouver après déformation.

La sélection du point se faisant sur écran, et donc sur un plan, il faut dans un premier temps chercher quel point de l'objet a pour image le point sélectionné par l'utilisateur (cf. § 2.2.2). De plus, on ne peut pas établir une bijection entre un déplacement dans le plan de l'écran et un déplacement dans l'espace, sauf si l'on impose une condition sur les degrés de liberté de l'objet. Ainsi, si on décide de fixer la composante en z de Δ_0 à 0, tout déplacement sur l'écran pourra être traduit en un déplacement dans l'espace. En outre, lorsque l'objet a beaucoup de points, la valeur de la déformation pour les points éloignés du point initial est très petite. Pour diminuer le temps de calcul, nous avons donc permis à l'utilisateur d'imposer un seuil à partir duquel les déformations ne sont plus prises en compte.

La figure 2.25 montre quelques exemples de déformation :

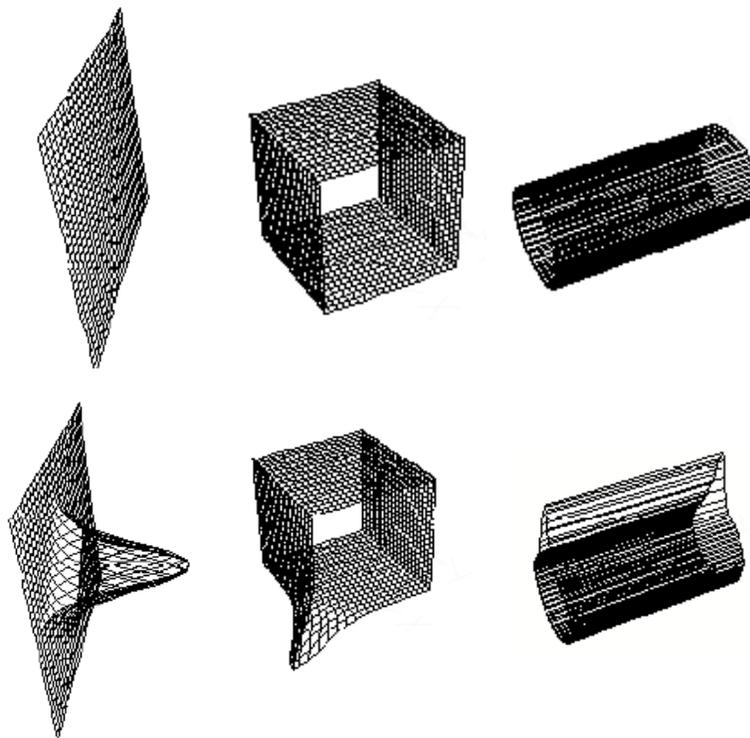


FIG. 2.25 – *Plan, Cube ouvert et Cylindre: En haut, avant déformation; En bas après déformation avec $p = 0.99$ et $f = 0.9$*

2.2.2.3 Détection des collisions

2.2.2.3.a Définition :

Pour que les guides puissent jouer un rôle autre que celui d'un simple indicateur visuel,

il faut qu'ils aient une consistance. En effet, il ne faut pas laisser l'utilisateur traverser ces guides si par exemple ils sont supposés limiter le champ d'action. Ainsi, à chaque fois que l'utilisateur rentrera en contact avec le guide, il faudra lui indiquer qu'il ne peut plus avancer dans cette direction. C'est ce contact que l'on nomme "collision" (figure 2.26).

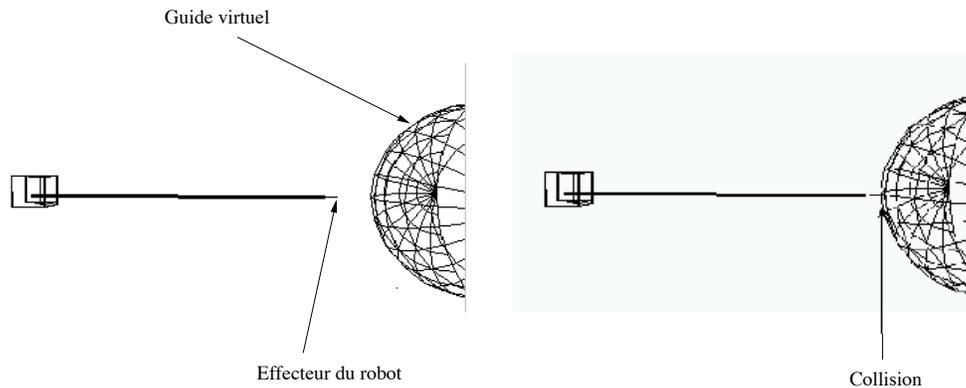


FIG. 2.26 – Collision entre l'effecteur du robot et un objet guide sous forme d'une demi-sphère

Nous avons développé deux méthodes de détection de collisions. La première est analytique, la seconde est géométrique.

2.2.2.3.b Méthode analytique :

Cette méthode est appliquée pour détecter les collisions quand le guide n'a pas subi de déformation. Lorsque tel est le cas, on connaît alors l'équation de ce guide. Nous décrivons ici brièvement la géométrie de quelques guides virtuels (figure 2.27), leurs paramètres (tableau 2.1) ainsi que leurs équations (tableau 2.2).

Type du guide	Paramètres du guide
Plan	$a, b.$
Disque	$R.$
Cube	a, b et $c.$
Cylindre, cône et tube	H, a_1 et b_1, a_2 et b_2
Cône et tube droits	H, a_1 et b_1, a_2 et b_2
Superellipsoïde	$a_1, a_2, a_3, \varepsilon_1, \varepsilon_2$
Supertoroïde	$a_1, a_2, a_3, a_4, \varepsilon_1, \varepsilon_2$

TAB. 2.1 – Les paramètres des guides virtuels pour la détection des collisions.

Où R représente le Rayon, H la hauteur, a_1 et b_1 la première ouverture (horizontale et verticale respectivement), a_2 et b_2 la deuxième ouverture (voir la géométrie de ces guides dans la figure 2.27).

Nous pouvons remarquer qu'avec ce type de méthode (disposant de l'équation du guide virtuel), la détection des collisions entre un point représentant l'effecteur du robot et un objet guide, peut se réduire à un test d'appartenance de ce point à l'objet guide. Cette écriture analytique offre aussi l'avantage de rendre la détection de collision identique quel que soit l'objet, si ce dernier est généré par une superquadrique par exemple (cf. § 2.2.1).

2.2.2.3.c La Méthode géométrique :

Lorsqu'au contraire le guide a été déformé, nous n'avons plus aucune information sur sa

Type du guide	Equation du guide
Plan	$z=0, x \in [0, a], y \in [0, b]$
Disque	$z = 0, x^2 + y^2 \leq R^2$
Cube	$z \in [0, c], x \in [0, a], y \in [0, b]$
Cylindre, cône et tube	$x \in [0, H], \left(\frac{y}{x(a_2-a_1)+Ha_1}\right)^2 + \left(\frac{z}{x(b_2-b_1)+Hb_1}\right)^2 = \frac{1}{H^2}$
Cône et tube droits	$x \in [0, H], \left(\frac{y}{x(a_2-a_1)+Ha_1}\right)^2 + \left(\frac{z}{x(b_2-b_1)+Hb_1} + 1\right)^2 = \frac{1}{H^2}$
Superellipsoïde	$\left(\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}}\right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}} = 1$
Supertoroïde	$\left(\left(\left(\frac{x}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\varepsilon_2}}\right)^{\frac{\varepsilon_2}{\varepsilon_1}} - a_4\right)^{\frac{2}{\varepsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\varepsilon_1}} = 1$

TAB. 2.2 – Les équations des guides virtuels pour la détection des collisions

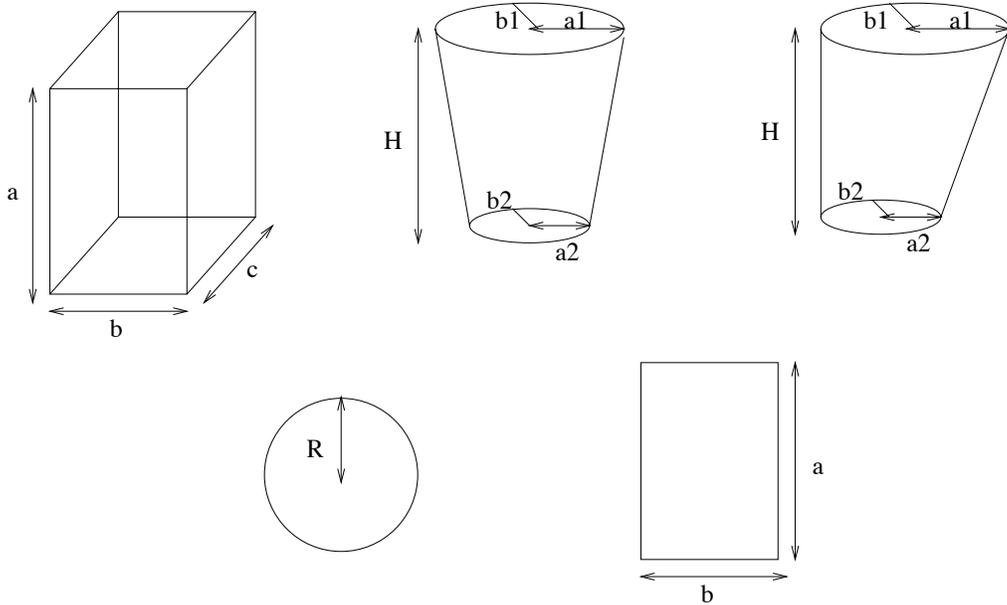


FIG. 2.27 – La géométrie des guides: de gauche à droite: Cube, (cylindre ou cône ou tube), (cône ou tube droits), plan, disque.

nature (on ne connaît pas l'équation d'un cylindre déformé par exemple). Nous pouvons donc uniquement travailler avec l'ensemble des sommets et des arêtes qui constituent l'objet.

Le but est de ne pas laisser le robot traverser l'objet guide, nous allons donc décider qu'il y a collision si l'utilisateur approche de trop près le guide (en définissant un seuil). Il faut donc à tout moment calculer la distance entre le robot et les arêtes du guide. On définit la distance entre un point M et un segment $[AB]$ comme étant :

$$d(M, [AB]) = \min\{\|\overrightarrow{MN}\|, N \in [AB]\} \quad (2.32)$$

Ainsi, si un point "E" représente l'extrémité de l'effecteur du robot, alors, nous dirons qu'il y a collision, si la distance du point "E" aux arêtes définissant l'objet est inférieure à un certain seuil "d" (figure 2.28).

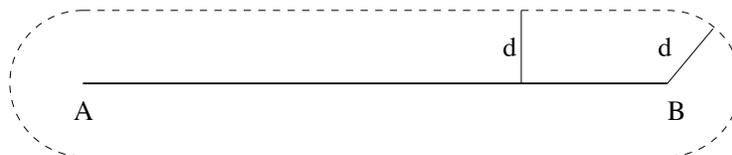


FIG. 2.28 – Zone de collision de seuil "d" autour d'un segment $[AB]$

Pour un seuil fixé, l'efficacité de cette méthode grandit avec le nombre d'arêtes définissant l'objet. Comme nous l'avons représenté dans la figure 2.29, si les arêtes de l'objet sont trop espacées, il apparaît des trous dans lesquels l'effecteur du robot pourra passer. Ainsi, il faudrait augmenter le nombre d'arêtes pour avoir de meilleurs résultats, mais cela réduirait la visibilité à l'écran (certaines parties de l'environnement seraient cachées par le guide).

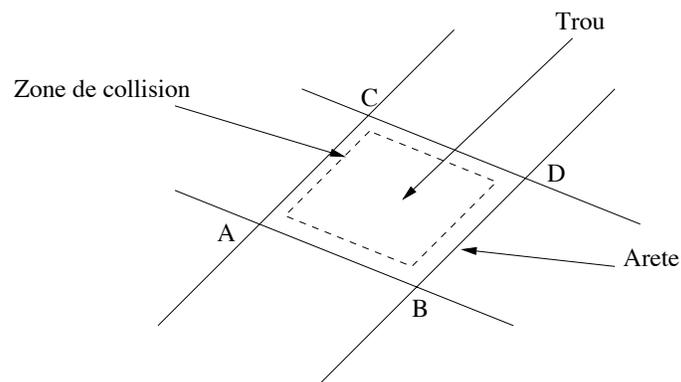


FIG. 2.29 – Existence de trous dans les guides virtuels

Le deuxième problème de cette méthode réside dans le temps de calcul. En effet, à chaque fois que l'utilisateur veut bouger le robot, il faut calculer la nouvelle distance entre le robot et le guide pour tester si ce mouvement provoque une collision.

Comme il est à la fois impératif que le système de détection de collision soit efficace et que la visibilité à l'écran soit bonne, nous avons décidé d'utiliser deux objets pour un même guide : on se servira d'un modèle avec peu d'arêtes pour représenter le guide à l'écran et d'un modèle avec beaucoup plus d'arêtes (qui ne sera pas affiché (figure 2.30)) pour faire les détections de collisions.

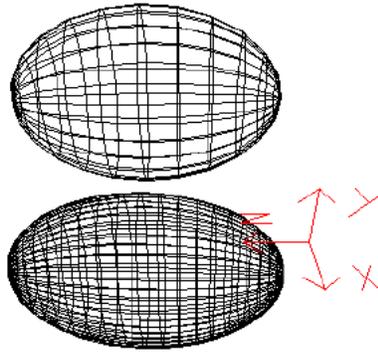


FIG. 2.30 – *Modèle utilisé à l'écran (en haut) et modèle utilisé pour les calculs (en bas)*

Pour diminuer le temps de calcul, nous avons introduit la notion de sphère de sécurité (figure 2.31). Lorsque le robot est en un point M_0 et que l'on fait le calcul de la distance, notée d_0 entre le robot et un guide statique (fixé dans l'environnement virtuel), on sait que tant que le robot reste dans la sphère de centre M_0 et de rayon d_0 il ne rentrera jamais en collision avec le guide. Ainsi, il n'est pas nécessaire d'effectuer les calculs des distances à l'intérieur de cette sphère. Une fois sorti de cette sphère, un nouveau calcul de distance est effectué pour créer une nouvelle sphère de sécurité. Cette méthode permet de réduire considérablement le temps de calcul des détections de collision.

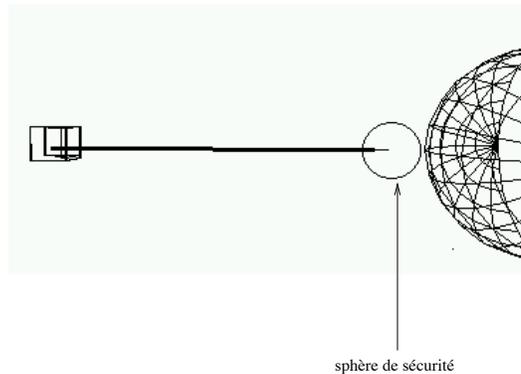


FIG. 2.31 – *Exemple de sphère de sécurité*

2.3 Les propriétés des guides virtuels

Dans ses recherches, Kheddar (Kheddar, 1997) a tenté de définir une structure globale pour les guides virtuels à l'aide d'un ensemble de champs pouvant être optionnels selon la nature et le contexte d'utilisation du guide virtuel :

– Un attachement :

Chaque guide virtuel peut être attaché à un objet virtuel ou à un endroit de l'environnement virtuel. Il peut être attaché d'une manière statique (il est alors figé à un endroit ou attaché à un objet virtuel particulier) ou dynamique (il apparaît suite à un événement, par exemple, lors de la détection de collisions). Ainsi, on définit pour chaque guide une position et une orientation dans l'espace défini par l'environnement virtuel.

– **Une zone d’influence :**

Au guide virtuel peut être associée une zone (de forme volumique, surfacique ou autre) qui jouera le rôle d’un bassin d’attraction ou tout simplement d’une zone d’action. En général, une zone d’influence est définie par une équation analytique (statique ou paramétrée) qui délimite (totalement ou partiellement) la forme du guide.

– **Une condition d’activation :**

A chaque guide est associé une condition d’activation. Elle peut se traduire par l’appartenance d’un ensemble de paramètres extérieurs à la zone d’influence ou par n’importe quelle autre condition liée à un événement.

– **Une fonction :**

La fonction du guide définit sa raison d’être. Elle peut être explicitée par des actions à établir à l’intérieur du guide virtuel.

– **Une condition de désactivation :**

La condition de désactivation met hors effet la fonction du guide virtuel. Elle peut être définie comme une négation de la condition d’activation ou l’atteinte d’un état final désiré.

L’idée d’une telle formalisation est très intéressante à l’assistance en téléopération, surtout si on veut que les guides virtuels soient utilisés comme de véritables outils d’assistance. Mais il n’existe pas encore de moyens interactifs permettant à l’opérateur de créer ses propres guides virtuels, de les configurer, de les activer, selon la tâche qu’il désire réaliser, ou même encore de les réutiliser. Un autre problème lié à l’utilisation de ces guides, se situe au niveau de leurs portabilité. En effet, ces outils virtuels sont très spécifiques et dépendent de la machine sur laquelle ils ont été créés. de même leur utilisation dépend complètement de leur concepteur.

Dans un premier temps notre approche consiste à définir une librairie de guides virtuels de base dont l’utilisation est limitée aux différentes tâches que nous avons prédéfinies (atteindre une cible, saisie d’un objet, dépôt d’un objet etc.). A chaque guide virtuel est associé des propriétés permettant à l’opérateur d’identifier ces outils et surtout de les réutiliser pour d’autres tâches. Ensuite nous avons généralisé ce concept vers la création interactive de guides virtuels et leur paramétrage en ligne (l’opérateur peut créer ses propres outils virtuels et peut les paramétrer selon la tâche qu’il désire réaliser). Ceci grâce à une interface homme-machine que nous avons développé sous l’environnement JAVA. Ce qui donne donc à l’opérateur la possibilité de créer sa boîte à outils virtuels à partir de n’importe quelle machine connectée à Internet (il s’agit donc d’une boîte à outils portable et partageable).

2.3.1 La notion de guide virtuel simple et composé

2.3.1.1 Guide virtuel simple

Un guide virtuel simple est représenté par une simple primitive géométrique. Elle peut être un segment de droite, un plan ou bien un volume défini par l’une des équations du paragraphe cf. § 2.2.2. Ces guides canoniques sont généralement utilisés pour réaliser des tâches simples telles que le suivi d’une ligne droite, évitement d’obstacles (des guides virtuels actifs avec un champ répulsif) ou bien atteindre un objet dans l’environnement virtuel (des guides virtuels actifs avec champ attractif). La figure 2.32 montre un exemple

de guide virtuel simple (primitive cône) qui peut être utilisé pour aider l'opérateur à déplacer l'effecteur du robot jusqu'à la cible.

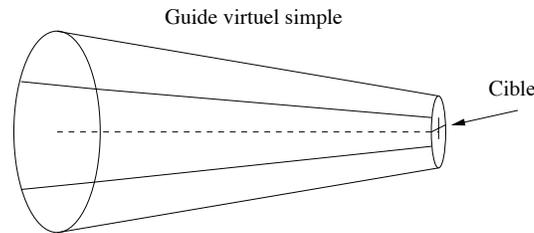


FIG. 2.32 – Exemple de “guide virtuel simple”, pour assister l'opérateur et/ou le robot à atteindre une cible

2.3.1.2 Guide virtuel composé

Un guide virtuel composé est constitué de plusieurs guides virtuels simples (un ensemble de segments définissant une trajectoire, un ensemble de plans ou bien un ensemble d'objets de forme quelconque). L'opérateur peut créer différents types de guides canoniques et peut lier ces guides simples entre eux pour former un guide composé. Ces types de guides sont généralement utilisés pour réaliser des tâches de téléopération complexes tel que suivre une trajectoire quelconque ou bien assembler et désassembler des objets. La figure 2.33 montre un guide virtuel composé, utilisé pour décrocher un objet depuis son support métallique (cf. § 4.1). Il s'agit d'une combinaison de 3 guides simples (sous forme de cylindres). Un autre exemple de guide virtuel composé est donné par la figure 2.34. Il est constitué par 2 guides simples (2 plans parallèles). Ce guide composé permet d'aider l'opérateur et/ou un robot mobile à traverser une porte.

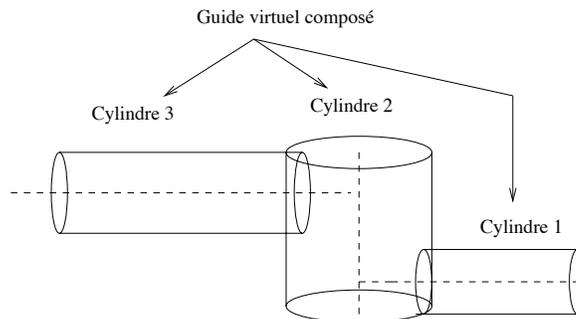


FIG. 2.33 – Exemple de “guide virtuel composé”, pour assister l'opérateur et/ou l'effecteur du robot à suivre une trajectoire et manipuler un objet

2.3.2 Notion de guide virtuel passif ou actif

2.3.2.1 Guide virtuel passif

Un guide virtuel (simple ou composé) est dit passif lorsque son utilisation est limitée à l'assistance à la perception de l'opérateur (il n'a pas d'effet direct sur le robot ni sur son environnement).

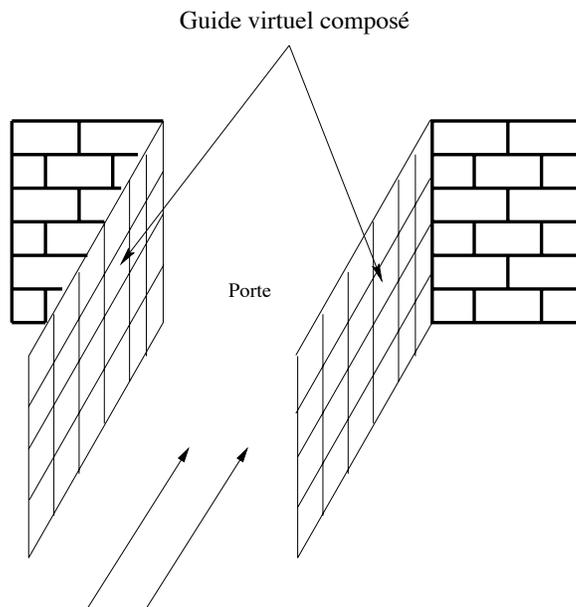


FIG. 2.34 – Exemple de “guide virtuel composé”, pour assister l’opérateur et/ou un robot mobile à traverser un porte

2.3.2.2 Guide virtuel actif

Un guide virtuel (simple ou composé) est dit actif lorsque son utilisation a un effet direct sur le robot ou sur son environnement. Les guides actifs peuvent se décomposer en deux catégories:

2.3.2.2.a Guide répulsif:

Dans ce cas, ce guide virtuel a pour rôle de générer un potentiel répulsif qui évite au robot de le traverser. Ils peut être considéré comme une barrière (pour empêcher le robot d’aller dans une zone interdite d’accès), une surface ou un volume protecteur (pour protéger des objets de l’environnement avec lesquels le robot ne doit pas avoir de collisions (figure 2.35) etc.

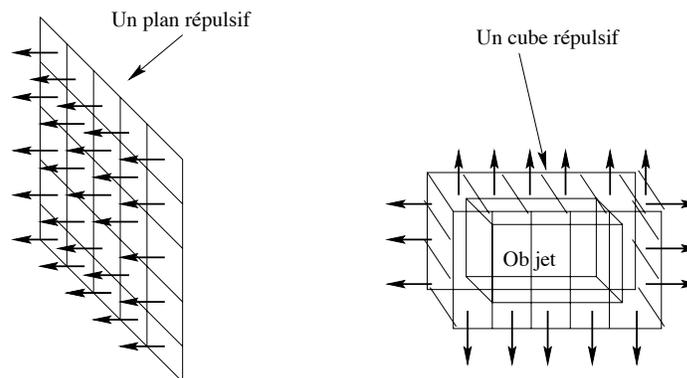


FIG. 2.35 – Exemple de “guide virtuel répulsif”, à gauche, une barrière et à droite, un volume englobant répulsif

2.3.2.2.b Guide attractif:

Son rôle est de générer un potentiel attractif permettant de contraindre l’opérateur

et/ou le robot   atteindre un objet de l'environnement (figure 2.36). Il est tr s efficace pour la g n ration de trajectoire.

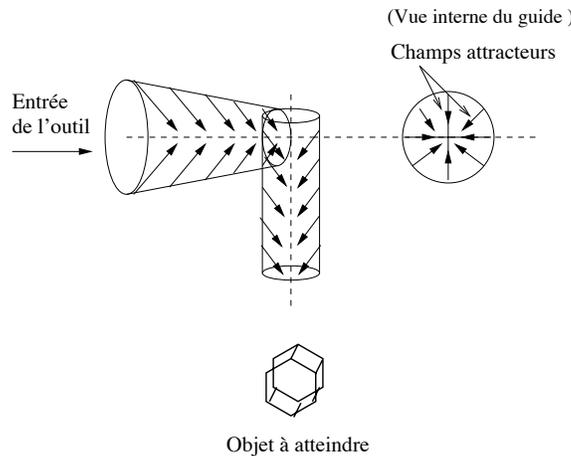


FIG. 2.36 – Exemple de “guide virtuel attractif”, pour atteindre un objet de l’environnement avec un outil port  par le robot

2.3.3 Formalisme propos  pour les guides virtuels

Nous avons vu qu’il existe de nombreuses possibilit s d’utilisation des guides virtuels. En effet, le r le d’un guide virtuel d pend enti rement de la fa on dont l’op rateur d sire l’utiliser (guide simple ou compos , statique ou dynamique, Actif ou passif, etc.). Afin de faire face   ces nombreux cas d’utilisation et de permettre la gestion des guides virtuels, une m thode de conception objet *UML* (*Unified Modeling Language*) a  t  utilis e. Cette m thode est n e en 1997 et elle propose de mod liser un syst me sous forme de diff rents diagrammes apportant chacun sa contribution.

Dans cette partie nous pr sentons essentiellement le formalisme et les diff rents types de donn es repr sentant les guides virtuels ainsi que les cas d’utilisation.

2.3.3.1 Structures de donn es propos es pour les guides virtuels

Cinq types de donn es ont  t  d finis pour repr senter un guide virtuel. Chaque type de donn e est repr sent  dans un fichier. Ces fichiers sont g n r s au moment de la cr ation d’un guide virtuel comme le montre la figure 2.37.

Soit par exemple “**nom_guide**”, le nom associ    un guide virtuel donn , le fichier contenant le mod le de ce guide sera donc “**nom_guide.obj**”, un mod le filaire faiblement discr t s  (avec peu de points et d’ar tes). Ce fichier est utilis  g n ralement pour l’affichage sur  cran et la d tection de collisions si nous poss dons l’ quation de ce guide. Dans le cas o  nous ne disposons pas de l’ quation du guide virtuel (si le guide a subi une d formation par exemple), alors le fichier “**nom_guide.obj**” est utilis  seulement pour l’affichage et un autre fichier “**nom_guide.obj2**”, un mod le fortement discr t s  (avec beaucoup de points et d’ar tes) est utilis  pour la d tection de collisions.

Les param tres de l’ quation du guide virtuel sont stock s dans un fichier “**nom_guide.obj.equ**”. Ces param tres sont charg s en m moire et utilis s pour le calcul des tests d’appartenance d’un point 3D au guide virtuel et aussi lors de la d tection de collisions entre

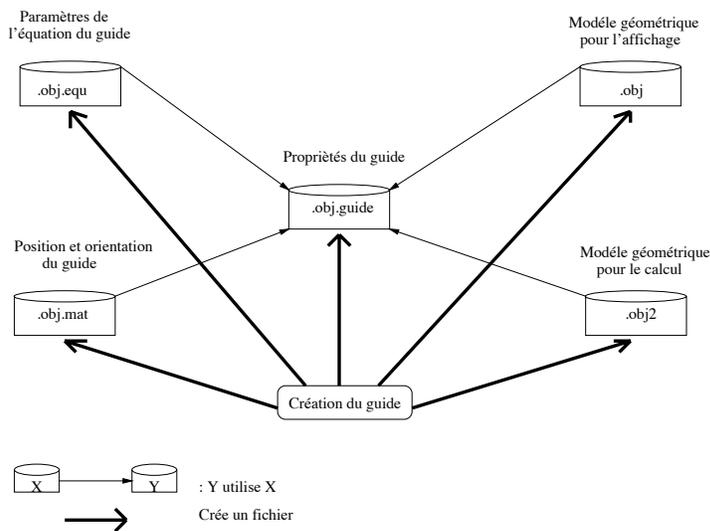


FIG. 2.37 – Représentation des différents types de données générés lors de la création d'un guide virtuel

l'effecteur du robot et le guide en question.

Une matrice homogène contenant la position et l'orientation du guide virtuel par rapport au référentiel de travail, est stockée dans le fichier “**nom_guide.obj.mat**”, elle est utilisée lors du chargement du guide afin de l'afficher au bon endroit.

Les propriétés du guide virtuel sont regroupées dans un fichier “**nom_guide.obj_guide**”. Il contient une structure de données correspondant au formalisme que nous avons utilisé et implanté. Cette structure de données (figure 2.38) contient un certain nombre de champs que nous avons jugé utiles pour définir les propriétés d'un guide virtuel.

- **Nom_guide** : Permet l'identification d'un guide virtuel
- **Type_guide** : Définit le type du guide virtuel (simple, composé, passif, actif). Si le guide est “composé” alors il contient des liens vers le guide précédent et le suivant. Dans le cas où le guide est “actif”, alors il nous informe s'il est “répulsif” ou “attractif”.
- **Référentiel** : Contient des informations concernant la position et l'orientation du guide dans son environnement virtuel.
- **Attachement** : Précise si le guide est “statique” ou “dynamique”. Dans le premier cas, il contient le nom de l'objet virtuel ou du point de passage auquel il est attaché. Dans le deuxième cas il contient la valeur “NULL” pour dire que le guide est libre.
- **Zone d'influence** : Contient le nom mathématique de l'équation ayant servi à la création de ce guide ainsi que les paramètres correspondants.

D'une manière générale, l'activation d'un guide virtuel se fait par un test d'appartenance d'un point (représentant l'effecteur du robot) au guide virtuel. Il s'agit d'une

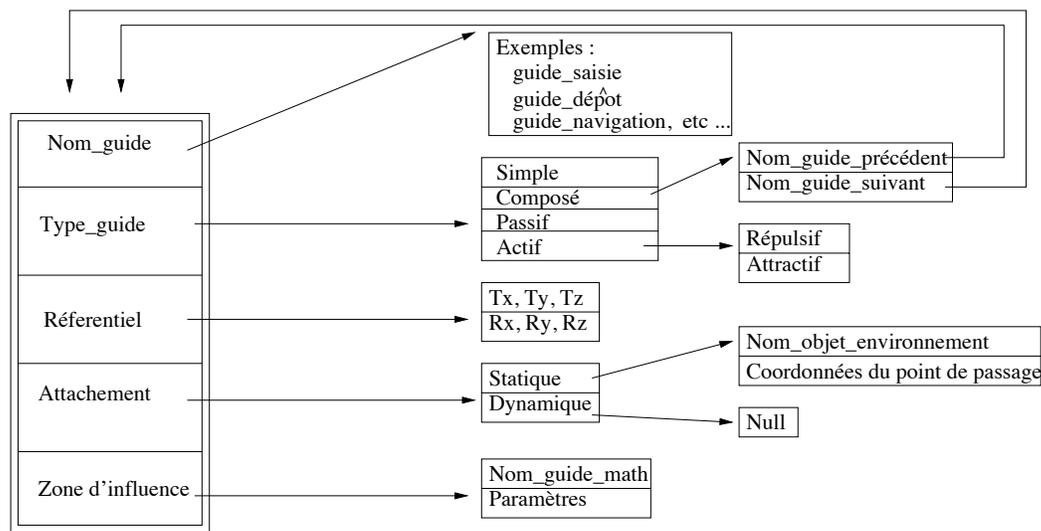


FIG. 2.38 – Schéma général de la structure des propriétés d'un guide virtuel

“**Pré-condition**” (ou encore une condition d’activation). Dans ce cas le nom et les paramètres de l’équation du guide sont récupérés depuis la structure de données définissant les propriétés de ce guide, et sont ensuite utilisés pour définir cette pré-condition. Il est aussi possible d’activer un guide par un autre évènement comme par exemple la détection de collisions ou encore définir une distance d’approche entre l’effecteur du robot et un objet de l’environnement virtuel.

Une fois le guide activé, il reste à déterminer les actions ou les opérations à réaliser en présence de ce guide virtuel, c’est ce que nous appelons la “**Fonction**” du guide. Généralement cette fonction est issue d’une combinaison des actions que réalise l’opérateur avec le robot virtuel et des fonctions définissant le type du guide. Si le guide est “passif” alors il est appelé “**guide d’assistance libre**”, s’il est “actif” (répulsif ou attractif) alors il est appelé “**guide d’assistance partagée ou semi-autonome**”. La fonction du guide peut aussi se limiter à l’exécution d’une fonction définie par le guide permettant une exécution automatique de la tâche par le robot virtuel. Dans ce cas ce guide devient “**autonome**” et est appelé “**guide d’assistance autonome**”.

La fonction du guide n’a plus d’effet sur le robot et sur l’opérateur, lorsque le robot a atteint un état final désiré (dans le cas d’un guide d’assistance à l’opérateur et d’un guide d’assistance partagée), ou encore la fin de l’exécution automatique de la tâche par le robot (dans le cas d’un guide d’assistance autonome). Il s’agit de désactiver la fonction du guide, ce qui définit la “**Post-condition**”.

2.3.3.2 Les cas d’utilisation

Dans la méthodologie *UML* l’étude des cas d’utilisation a pour objectif de déterminer ce que chaque acteur (intervenant) attend du système. En effet dans notre cas, les acteurs potentiels sont au nombre de trois :

- Utilisateur de guide.
- Superviseur.

- Administrateur de guide.

Le fait que l'utilisation des guides virtuels peut se faire par Internet, et par n'importe quel client (opérateur) connecté au réseau, il est nécessaire de déterminer le statut de cet opérateur (utilisateur, superviseur ou administrateur). Le tableau 2.3 résume ces différents cas d'utilisation. Des applications mettant en œuvre des acteurs et des guides virtuels, sont données dans le chapitre 4.

Acteur	Cas d'utilisation
Administrateur de guide	Création d'un guide
Administrateur de guide	Lecture d'un guide
Administrateur de guide	Modification d'un guide
Administrateur de guide	Sauvegarde d'un guide
Utilisateur	Utilisation des guides
Superviseur	Edition des propriétés d'un guide

TAB. 2.3 – *Les différents cas d'utilisation des guides virtuels associés à chaque acteur.*

2.4 Bilan

Dans ce chapitre, les notions de mécanisme et de guide virtuel indispensables à un système de réalité augmentée basé sur un retour prédictif sont présentées. Nous avons aussi présenté les méthodes analytiques et géométriques de création, de représentation et de manipulation des objets virtuels en général et en particulier des guides virtuels. Nous avons proposé un nouveau formalisme pour les guides virtuels afin de les rendre paramétrables, évolutifs et utilisables comme de véritables outils d'assistance à la téléopération (ce formalisme est implanté, des exemples sont donnés dans le chapitre 4).

Cependant, pour télétravailler sur Internet, il est nécessaire que ces outils d'assistance soient portables et partageables (lors d'un travail coopératif par exemple). Il est aussi nécessaire, que l'environnement virtuel (robot, objet, etc.) soit portable ainsi que tous les autres modules utiles à la réalisation d'un système de télétravail (IHM, retour vidéo, contrôle des robots réel et virtuel, etc.). Cette partie est présentée dans le chapitre suivant.

Chapitre 3

Le système expérimental de télétravail proposé

Dans ce chapitre, nous étudions l'ensemble des problématiques posées dans le premier chapitre à savoir, les outils logiciels et matériels nécessaires pour la mise œuvre d'un système de télétravail, l'intégration des nouveaux concepts et des solutions déjà apportées aux problèmes liés à la télérobotique, sur une plate-forme universelle et à moindre coût. Nous verrons, comment rendre l'utilisation des systèmes de télétravail suffisamment souple et intuitive pour l'opérateur, et comment les outils d'assistance présentés dans le chapitre précédent peuvent être utilisés pour améliorer les performances de l'opérateur lors du télétravail.

Nous commençons ce chapitre par une présentation des caractéristiques d'un système de télétravail idéal. Ensuite, nous présentons les méthodes nécessaires pour le contrôle en réalité augmentée. Il s'agit des méthodes de modélisation géométrique de l'environnement et de calibration de la caméra et du robot.

Dans la troisième section, nous présentons notre système expérimental de télétravail, baptisé ARITI "Augmented Reality Interface for telerobotic applications via Internet". Il s'agit d'un système de télétravail dont la conception et la mise en œuvre est basée d'un côté sur des techniques de la réalité virtuelle et augmentée et de l'autre côté sur l'exploitation des techniques de développement réseaux et en particulier sur Internet.

Dans la quatrième section, nous présentons l'architecture réseau et les différents protocoles de communication de l'architecture client/serveur du système ARITI.

3.1 Caractéristiques d'un système de télétravail idéal

A l'issu de ce que nous avons présenté dans la première section du premier chapitre et selon les recherches de la communauté CSCW (Bannon et Schmidt, 1989), (Schal et Zeller, 1990) et (Greenberg, 1991), nous tentons maintenant de définir les caractéristiques d'un système de télétravail idéal :

❶ Convivialité :

Le système doit posséder une IHM (Interface Homme Machine) conviviale, qui peut être utilisée facilement par des utilisateurs pas forcément spécialistes du domaine de la robotique ou de l'informatique. plusieurs types d'assistances doivent être fournis à l'utilisateur. Il s'agit généralement d'assistances à la perception de l'environnement distant, à la commande du système distant (par exemple le robot) et à la supervision des tâches réalisées.

❷ Multimodalité :

Le système doit pouvoir supporter l'échange d'informations multimodales entre le site maître et le site esclave. Il s'agit d'informations Visuelle, Sonore, Tactile et Kinesthésique.

❸ Portabilité :

l'interface homme machine doit être portable. L'utilisateur doit pouvoir télé-travailler depuis n'importe quelle machine connectée au réseau (en particulier au réseau Internet).

❹ Télé-coopération :

Le système doit supporter un télétravail coopératif, par exemple permettre de réaliser des missions complexes en faisant participer plusieurs personnes qui peuvent se trouver dans des endroits différents.

❺ Télé-maintenance :

La maintenance du système peut être réalisée à distance.

❻ Flexibilité et modularité :

Le système doit être flexible et modulaire, dans la mesure où il doit être ouvert à d'éventuelles modifications.

❼ Coût :

La réalisation d'un tel système ne doit pas engendrer un coût très important (généralement le coût du matériel utilisé).

3.2 Modélisation géométrique d'environnement pour le contrôle en Réalité Augmentée

Dans cette partie nous allons décrire les méthodes de modélisation d'environnement utilisées pour le contrôle en réalité augmentée du robot et de son environnement. Il s'agit ici de la modélisation géométrique de la scène (robot, objets, guides virtuels, etc.) et de la modélisation et calibration de la caméra et du robot.

3.2.1 Modélisation géométrique de la scène

La modélisation géométrique a pour objectif de créer un modèle de la scène, ce modèle contient deux types de données (géométriques et topologiques). Les données géométriques

sont les entités de la scène tels que le robot, les objets de l'environnement ainsi que les guides virtuels. Les données Topologiques quant à elles définissent les relations entre ces entités et les constituants d'une même entité. Il ne s'agit pas ici de faire un état de l'art complet sur les méthodes de représentations d'objets 3D. Mais nous allons décrire brièvement les plus connues et ensuite nous allons présenter celles que nous avons choisi pour télétravailler sur Internet.

Il existe trois catégories de représentation : filaire, surfacique et volumique.

3.2.1.1 Représentation filaire

Le modèle fil de fer "*wire frame*" est historiquement la première représentation graphique à avoir été mise en œuvre. Il retient de l'objet, les coordonnées des sommets définissant ainsi les arêtes les joignants. Ne connaissant que les arêtes et les sommets, plusieurs interprétations d'un même modèle peuvent être faites.

Cette méthode a pour intérêt de permettre une création et une visualisation rapide du modèle car elle permet une modification aisée des points et des arêtes. La capacité mémoire utilisée est également très faible.

3.2.1.2 Représentations surfaciques

Ce sont les représentations se référant uniquement aux surfaces de l'objet. La plus utilisée est la BREP "Boundary REPresentation". C'est une représentation du solide par ses limites (frontières) (Qiang, 1989). Les surfaces envisageables peuvent être diverses: plans, surfaces de révolution, surfaces quelconques. L'objet est construit par assemblage de primitives géométriques de base. Les primitives sont les éléments géométriques suivants : point, segment, polygone et polyèdre. Les liaisons entre ces primitives sont définies par des relations topologiques d'incidences, de contiguïté et d'inclusion. Ce type de modélisation permet implicitement de modéliser aussi les volumes dans la mesure où la position de la matière est connue.

Dans le cas le plus fréquent des polyèdres, les limites de l'objet sont des plans. Il est alors défini par ses faces, ses arêtes et ses sommets; d'où l'appellation de modèle Face-Arête-Sommet (figure3.1). Deux façons permettent de construire le modèle:

- la première repose sur la définition d'un modèle fil de fer auquel sont associées les surfaces correspondantes.
- la seconde consiste à balayer un contour de base le long d'un parcours linéaire ou non linéaire pour créer le solide correspondant.

3.2.1.3 Représentations volumiques

Ces représentations concernent le volume de l'objet. Différents modèles entrent dans cette catégorie, nous résumons quelques uns.

La représentation "*Constructive Solid Geometry* " (CSG): Il s'agit d'une méthode de construction d'objets par l'application d'opérations d'assemblage d'où son appellation. La construction de l'objet se fait par combinaison de volumes élémentaires. La démarche est semblable à celle de l'artisan, qui usine les pièces élémentaires, les assemble pour en

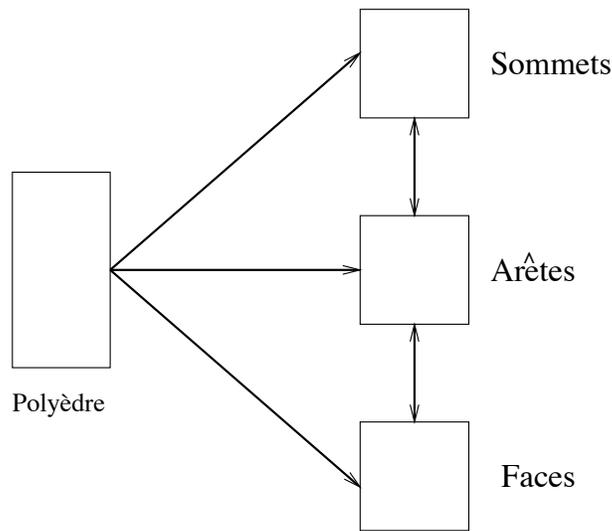


FIG. 3.1 – Représentation informatique BREP d'un polyèdre

obtenir une plus complexe figure 3.2. Elle est décrite par un arbre de construction où les feuilles sont des solides de base et les nœuds sont des opérations comme: l'union, l'intersection, la différence, les transformations géométriques composées de translations ou de rotations.

L'objet est construit à partir de primitives géométriques de base ou volumes élémentaires (cube, cylindre, etc.) auxquels sont appliqués les opérateurs logiques afin de les assembler et de créer d'autres volumes. Les primitives géométriques de base sont paramétrées. Les données canoniques simples caractérisent chacune des primitives (centre, rayon et hauteur pour le cylindre par exemple).

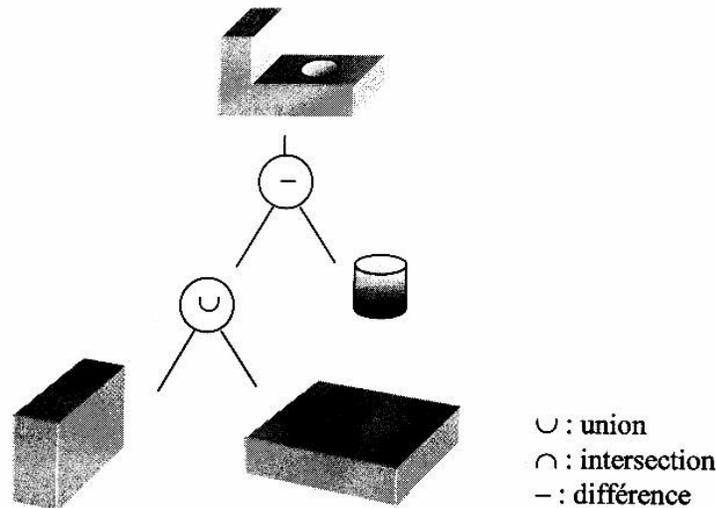


FIG. 3.2 – Représentation CSG simple

Le paramétrage des primitives et la variété d'opérateurs facilitent la création de nouveaux solides en quelques secondes. La simplicité des données canoniques des primitives rend la représentation des données concises, d'où un gain en place mémoire. Utilisée souvent en CAO, elle ne peut être cependant exploitée directement pour la visualisation et nécessite une conversion en un modèle plus approprié (la BREP par exemple) au cours

d'un processus de conversion lourd en temps de calcul.

Les volumes peuvent être représentés par Enumération Spatiale. Celle-ci contenant plusieurs sous catégories (Pampagnin, 1990), nous présentons quelque unes:

- décomposition primaire de la scène en *Voxels*. Dans ce cas les éléments de volume sont des cubes de taille fixée auxquels sont attribuées des valeurs de vérité concernant la présence ou non de la matière. La méthode est simple mais l'espace mémoire occupé est prohibitif.
- décomposition optimisée de la scène en *Octrees*. L'espace 3D est discrétisé de manière récursive. Si un cube n'est pas homogène (plein ou vide), il est décomposé en huit cubes plus petits et ainsi de suite. On obtient une description arborescente dont les feuilles sont des cubes "homogènes". Un exemple de l'utilisation de l'octree est la représentation de l'espace libre dans l'espace de configuration d'un robot manipulateur. Les deux décompositions nécessitent trop de place mémoire.

3.2.1.4 La représentation choisie pour le télétravail via Internet

Nous rappelons que notre objectif n'est pas seulement de représenter le robot et les objets, mais surtout de les représenter pour qu'ils puissent être utilisés pour le télétravail via Internet. Par conséquent, ils ne doivent pas dépendre d'une certaine plate forme ou de bibliothèques graphiques particulières qui empêcheraient leurs mobilités (utilisation à distance via le réseau Internet). De plus, ces objets ne doivent pas être lourds à l'affichage et à la manipulation et surtout ne doivent pas trop encombrer l'interface de télétravail fournie à l'utilisateur. Nous avons donc choisi le modèle *fil de fer* pour représenter le robot, les objets et les guides virtuels.

Le premier niveau de représentation du modèle se situe au niveau des fichiers. Chaque fichier est au format *wavefront*¹ dont quelques éléments principaux sont décrits comme suit :

- $v\ x\ y\ z$: Introduit un nouveau point de coordonnées (x, y, z) dans le modèle.
- $f\ p_1\ p_2\ p_3\ p_4\ \dots$: Décrit un polygone formé des points $p_1\ p_2\ p_3\ p_4\ \dots$
- $l\ p_1\ p_2$: Crée un lien (une arête) entre les points p_1 et p_2 .

Un modèle "fil de fer" d'un objet est constitué de l'ensemble des points du modèle (appelés sommets) et de l'ensemble des arêtes de l'objet.

Ainsi, nous définissons un objet par :

- Le tableau des coordonnées des sommets de l'objet.
- Le tableau des couples des sommets qui forment une arête.

La figure 3.3 illustre un exemple de représentation d'un triangle avec ce format.

1. format standard et stable utilisé pour la représentation physique des objets sous Java. Il est supporté par la plupart des navigateurs Internet

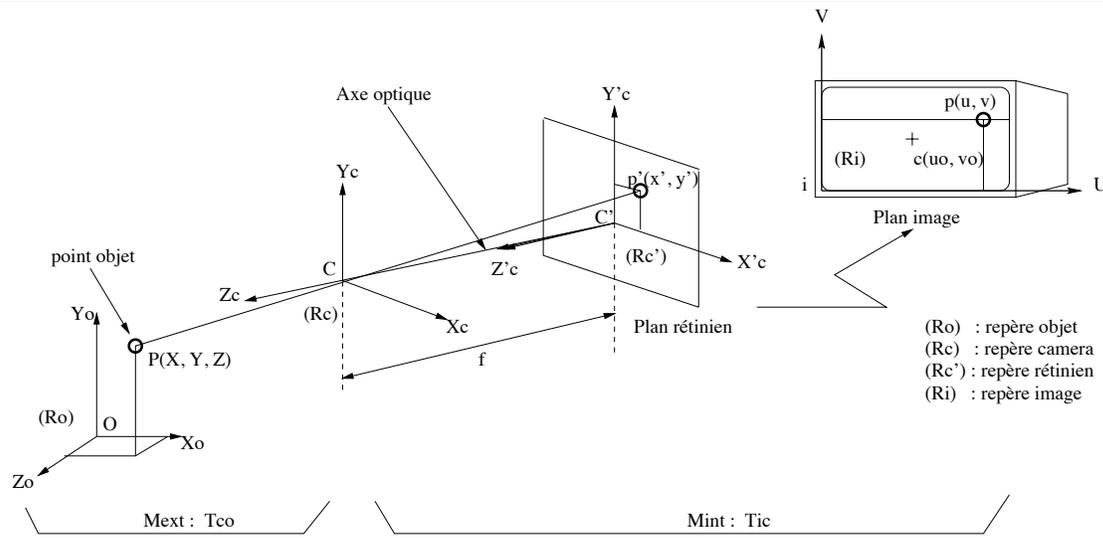


FIG. 3.4 – Modèles géométriques de caméra

3.2.2.1 Modèle interne de la caméra

Le modèle interne de la caméra est représenté par les transformations permettant de passer du repère de la caméra au repère lié au plan rétinien puis, au plan image :

❶ Projection dans le plan rétinien :

C'est la transformation qui permet de passer du repère de la caméra (R_c), au repère rétinien ($R_{c'}$). Soit $P(X_c, Y_c, Z_c)$ un point dans (R_c), sa projection dans le plan rétinien donne $p'(x', y')$ dans ($R_{c'}$). Cela donne :

$$\begin{cases} X_c/Z_c = x'/f \\ Y_c/Z_c = y'/f \end{cases} \quad (3.1)$$

❷ Projection dans le plan image :

C'est la transformation qui fait passer du repère ($R_{c'}$) au repère (R_i). Cette transformation est une homothétie suivant les deux axes $X_{c'}$ et $Y_{c'}$; puis un changement d'origine.

On obtient le modèle suivant :

$$\begin{cases} u = k_i \cdot x' + u_0 = k_u \cdot \frac{X_c}{Z_c} + u_0 & , k_u = k_i \cdot f \\ v = k_j \cdot y' + v_0 = k_v \cdot \frac{Y_c}{Z_c} + v_0 & , k_v = k_j \cdot f \end{cases} \quad (3.2)$$

ce qui donne sous forme matricielle :

$$\begin{pmatrix} Z_c \cdot u \\ Z_c \cdot v \\ Z_c \end{pmatrix} = \begin{pmatrix} k_u & 0 & u_0 & 0 \\ 0 & k_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = M_{int}(3 \times 4) \cdot \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \quad (3.3)$$

$M_{int}(3 \times 4)$ est le modèle interne, matrice de passage de (R_c) à (R_i) et les paramètres f, k_i, k_j, u_0, v_0 s'appellent les paramètres internes de la caméra (u_0, v_0 représentent l'intersection entre l'axe optique et le plan image, k_i, k_j sont les facteurs d'échelle exprimant l'homothétie entre plan caméra et plan image et f la distance focale de l'objectif)

3.2.2.2 Modèle externe de la caméra

C'est la transformation qui permet de passer du repère (R_0) au repère (R_c) . Un point de la scène ne peut être connu que par rapport à un repère lié à celle-ci, appelé repère de travail (R_0) . Trois rotations sont possibles suivant les angles $(\theta_x, \theta_y, \theta_z)$ et trois translations sont possibles suivant le vecteur $(t_a, t_b, t_c)^t$ permettant de définir la situation de (R_c) par rapport à (R_0) . Un point P , de coordonnées X, Y, Z connues dans le repère (R_0) , est représenté dans le repère (R_c) par :

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R_{3 \times 3} & T_{3 \times 1} \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = M_{ext}(4 \times 4) \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.4)$$

où

$$R_{3 \times 3} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad \text{et} \quad T_{3 \times 1} = \begin{pmatrix} t_a \\ t_b \\ t_c \end{pmatrix}$$

$M_{ext}(4 \times 4)$ est le modèle externe, exprimant la situation (position et orientation) de la caméra par rapport au repère objet (R_0) .

Les paramètres $(t_a, t_b, t_c, \theta_x, \theta_y, \theta_z)$, qui sont déterminés à partir de $M_{ext}(4 \times 4)$, sont appelés les paramètres externes de la caméra.

3.2.2.3 Modèle global de la caméra

La matrice qui permet de passer directement du repère de travail (R_0) au repère (R_i) lié au plan d'affichage, permet de déterminer le modèle global de la caméra en composant les modèles interne et externe :

$$\begin{aligned} \begin{pmatrix} Z_c \cdot u \\ Z_c \cdot v \\ Z_c \\ 1 \end{pmatrix} &= \begin{pmatrix} k_i \cdot f & 0 & u_0 & 0 \\ 0 & k_j \cdot f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} k_i \cdot f & 0 & u_0 & 0 \\ 0 & k_j \cdot f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} R_{3 \times 3} & T_{3 \times 1} \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \\ &= M_{int} \cdot M_{ext} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \end{aligned}$$

qui s'écrit :

$$\begin{pmatrix} s \cdot u \\ s \cdot v \\ s \end{pmatrix} = C_{3 \times 4} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.5)$$

avec

$$C_{3 \times 4} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{pmatrix} \quad \text{et} \quad s = Z_c$$

la coordonnée $s = Z_c$ est appelé la coordonnée homogène.

$C_{3 \times 4}$ représente le modèle global de la caméra, qui exprime la relation entre le repère objet (R_0) et le repère image (R_i). Les paramètres c_{ij} ($i = 1..3, j = 1..4$) sont appelés les paramètres globaux de la caméra.

3.2.2.4 Modèle géométrique direct de la caméra (MGD)

Ce modèle permet d'exprimer les coordonnées (u, v) du point p image du point P .

La relation (Éq. (3.5)) est décrite par les équations :

$$\begin{cases} P^t \cdot c_1 + c_{14} - u(P^t \cdot c_3 + c_{34}) = 0 \\ P^t \cdot c_2 + c_{24} - v(P^t \cdot c_3 + c_{34}) = 0 \end{cases} \quad (3.6)$$

où :

- (\cdot) : produit scalaire,
- P : vecteur de coordonnées X, Y, Z ,
- c_1 : vecteur de coordonnées (c_{11}, c_{12}, c_{13}) ,
- c_2 : vecteur de coordonnées (c_{21}, c_{22}, c_{23}) ,
- c_3 : vecteur de coordonnées (c_{31}, c_{32}, c_{33}) .

La relation (Éq. (3.6)) permet d'exprimer les coordonnées pixel $p(u, v)$ de l'image du point P . On obtient ainsi le modèle géométrique direct de la caméra suivant :

$$\begin{cases} u = \frac{P^t \cdot c_1 + c_{14}}{P^t \cdot c_3 + c_{34}} \\ v = \frac{P^t \cdot c_2 + c_{24}}{P^t \cdot c_3 + c_{34}} \end{cases} \quad (3.7)$$

3.2.2.5 Modèle géométrique inverse de la caméra (MGI)

Ce modèle exprime l'équation du rayon optique passant par le point image p des coordonnées (u, v) . La relation (Éq. (3.6)) donne deux équations indépendantes caractérisant le modèle inverse de la caméra qui peut se mettre sous la forme :

$$\begin{cases} P^t \cdot (c_1 - u \cdot c_3) + c_{14} - u \cdot c_{34} = 0 \\ P^t \cdot (c_2 - v \cdot c_3) + c_{24} - v \cdot c_{34} = 0 \end{cases} \quad (3.8)$$

et peut s'écrire :

$$\begin{cases} p^t \cdot \vec{n}\vec{u} + au = 0 \\ p^t \cdot \vec{n}\vec{v} + av = 0 \end{cases} \quad (3.9)$$

où

$$\vec{n}\vec{u} = \begin{pmatrix} c_{11} - u \cdot c_{31} \\ c_{21} - u \cdot c_{32} \\ c_{13} - u \cdot c_{33} \end{pmatrix}, \quad \vec{n}\vec{v} = \begin{pmatrix} c_{21} - v \cdot c_{31} \\ c_{22} - v \cdot c_{32} \\ c_{23} - v \cdot c_{33} \end{pmatrix}$$

et

$$au = (c_{14} - u \cdot c_{34}), \quad av = (c_{24} - v \cdot c_{34})$$

La relation (Éq. (3.9)) représente deux équations de plans, où $\vec{n}\vec{u}$ et $\vec{n}\vec{v}$ sont les normales à ceux-ci. Le rayon optique, contenant le point P et son image p , est fourni par l'intersection de ces deux plans. Le vecteur directeur du rayon optique définissant le modèle géométrique inverse de la caméra est donné par :

$$\vec{r} = \vec{n}\vec{u} \wedge \vec{n}\vec{v} \quad (3.10)$$

3.2.2.6 Calibration de la caméra

La calibration de la caméra consiste à déterminer les paramètres du modèle de celle-ci. Le pré-requis est une acquisition d'un ensemble de points dans le repère objet et de leurs images. Afin de calibrer la caméra, nous avons expérimenté le modèle linéaire décrit au début de la section § 3.2.2 (voir aussi la figure 3.4).

D'après la relation Éq. (3.6), un point donne deux équations. Afin d'identifier les paramètres globaux $c_{ij}(i = 1..3, j = 1..4)$ de la caméra, il faut donc au moins 6 points (non tous coplanaires, non optiquement alignés) et leurs images respectives. Pour avoir une meilleure précision lors de l'estimation de ces paramètres, par des méthodes d'optimisation mathématiques, plus de 6 points et leurs images sont nécessaires². Nous avons appliqué alors la méthode des moindres carrés (Loukil, 1993).

Nous avons utilisé une procédure de calibration classique, les coordonnées des points de la calibration sont fournies manuellement par un opérateur. Les coordonnées des points de l'image sont obtenues par pointage sur l'écran. Bien qu'il existe actuellement d'autres méthodes de calibration telles que l'utilisation de plusieurs vues d'un objet (Devy *et al.*, 1997) ou encore la calibration automatique (qui est basée sur l'utilisation d'un robot à quatre degrés de liberté et sur un traitement des tâches lumineuses) (Malleme *et al.*, 1999). Nous nous sommes contentés d'estimer le modèle linéaire de la caméra sans prendre en compte les distorsions et les résultats obtenus sont satisfaisants pour l'objectif fixé, celui de superposer une image virtuelle sur une image réelle.

3.2.3 Modélisation et calibration du robot

L'objectif de la modélisation du robot (l'effecteur du robot) est la détermination des relations mathématiques, entre celui-ci et l'environnement. Cette modélisation représente le formalisme mathématique permettant d'établir une relation analytique entre les coordonnées d'un point 3D de l'environnement dans le repère du monde et les consignes moteurs nécessaires pour que l'effecteur du robot atteigne ce point 3D.

Le robot utilisé est un système mécanique à 4 ddl (degrés de libertés). Il est constitué d'une grande base, d'un support, d'une tourelle et d'une tige montée sur celle-ci. A l'extrémité de la tige se trouve une pointe métallique utilisée comme moyen de saisie d'objets (figure 3.5). Les degrés de liberté du robot sont répartis comme suit : la grande base et le support effectuent deux translations parallèlement au sol, la tourelle quant à elle effectue deux rotations en site et azimuth.

2. Pour calibrer notre caméra nous avons pris 18 points

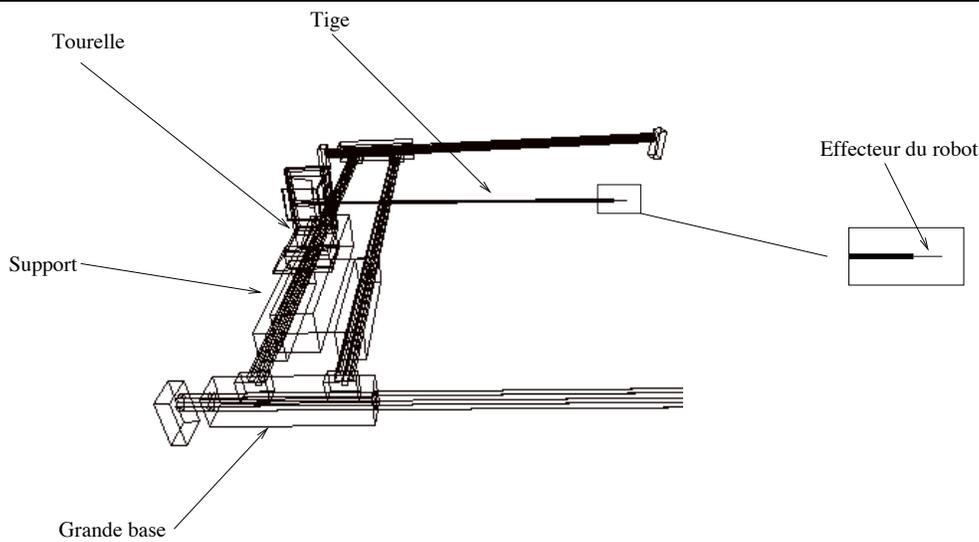


FIG. 3.5 – Les composantes du robot à 4 ddl utilisé pour la calibration

Nous représentons dans un premier temps les différents repères et paramètres du système de mesures utilisé (figure 3.6) et ensuite les deux modèles géométriques direct et inverse du robot.

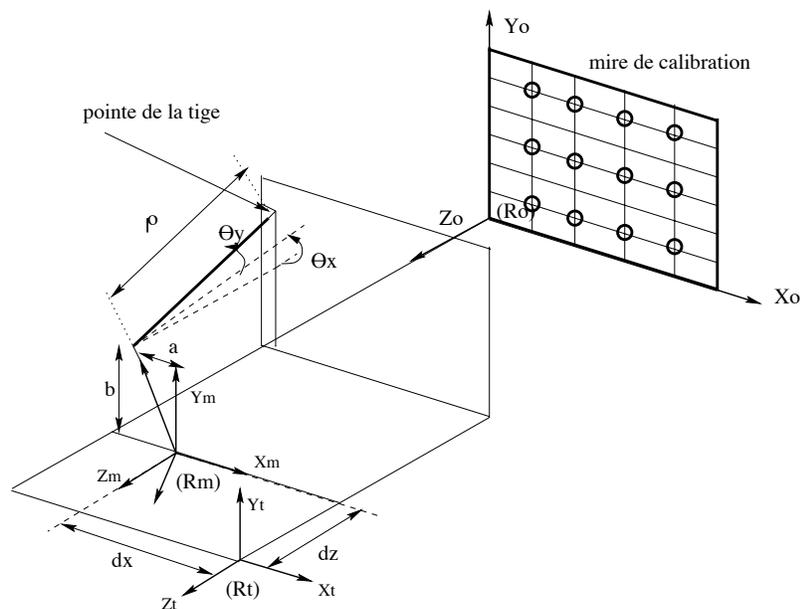


FIG. 3.6 – Les différents repères du système de mesures

- R_o : repère objet (c'est le repère de la mire utilisé pour la calibration du robot).
- R_t : repère associé à la position de départ (origine des moteurs) du robot.
- R_m : repère mécanique de la tourelle où s'effectue les rotations.
- (x_o, y_o, z_o) : coordonnées du point dans le repère objet R_o .
- $(d_x, d_z, \theta_x, \theta_y)$: consignes des moteurs du robot.
- ρ : longueur de la tige (suivant l'axe Z).
- b : bras de levier (suivant Y).
- a : bras de levier (suivant X).
- (t_{ox}, t_{oy}, t_{oz}) : paramètres de translation de R_0 vers R_t .

3.2.3.1 Modèle Géométrique Direct du robot (MGD)

Le MGD permet d'obtenir les coordonnées x_o, y_o , et z_o (en mètres) de l'extrémité de la tige du robot en fonction des paramètres $d_x, d_z, \theta_x, \theta_y$. Le passage de R_t vers R_o se fait par une translation pure, il s'exprime par :

$$\begin{cases} x_o = t_{ox} + x_t \\ y_o = t_{oy} + y_t \\ z_o = t_{oz} + z_t \end{cases} \quad (3.11)$$

les coordonnées du repère mécanique R_m s'expriment dans le repère R_t par :

$$\begin{cases} x_t = d_x + x_m \\ y_t = y_m \\ z_t = d_z + z_m \end{cases} \quad (3.12)$$

les coordonnées de l'extrémité de la tige s'expriment dans le repère mécanique par :

$$\begin{pmatrix} x_m \\ y_m \\ z_m \\ 1 \end{pmatrix} = \begin{pmatrix} C\theta_y & S\theta_x \cdot S\theta_y & C\theta_x \cdot S\theta_y & 0 \\ 0 & C\theta_x & -S\theta_x & 0 \\ -S\theta_y & S\theta_x \cdot C\theta_y & C\theta_x \cdot C\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & -\rho \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.13)$$

Soit finalement le MGD qui exprime les coordonnées dans le repère objet en fonction des autres paramètres:

$$\begin{cases} x_o = t_{ox} + d_x + a \cdot C\theta_y + b \cdot S\theta_x \cdot S\theta_y - \rho \cdot C\theta_x \cdot S\theta_y \\ y_o = t_{oy} + b \cdot C\theta_x + \rho \cdot S\theta_x \\ z_o = t_{oz} + d_z - a \cdot S\theta_y + b \cdot S\theta_x \cdot C\theta_y - \rho \cdot C\theta_x \cdot C\theta_y \end{cases} \quad (3.14)$$

avec

$$C\theta = \cos \theta \quad \text{et} \quad S\theta = \sin \theta$$

3.2.3.2 Modèle Géométrique Inverse du robot (MGI)

Ce modèle est l'inverse du précédent. Il permet d'obtenir les paramètres de translation et de rotation du robot en fonction des coordonnées x_o, y_o et z_o de l'extrémité de la tige. Le système d'équation (Éq. (3.14)), contient trois équations à quatre inconnues, donc il y a à priori une infinité de solutions. Vu la configuration du système (4 ddl), un même point peut être atteint en utilisant plusieurs combinaisons des différents moteurs. Nous avons donc imposé une valeur à l'un des moteurs pour obtenir une solution unique (ici c'est la rotation θ_y qui est fixée à zéro par défaut). Donc il nous reste plus qu'à déterminer les trois paramètres (d_x, d_z, θ_x).

On pose $C\theta_x = (1 - t^2)/(1 + t^2)$, $S\theta_x = 2t^2/(1 + t^2)$ et $t = \tan(\theta_x/2)$ dans la deuxième équation du système (Éq. (3.14)) qui devient :

$$b \cdot C\theta_x + \rho \cdot S\theta_x = y_o - t_{oy}$$

on obtient donc deux solutions :

$$\begin{cases} \theta_x = 2 \cdot \arctan\left(\frac{\rho \mp \sqrt{\rho^2 - (y_o - t_{oy})^2 + b^2}}{y_o - t_{yo} + b}\right) \\ d_x = x_o - t_{ox} - a \cdot C\theta_y - b \cdot S\theta_y \cdot S\theta_x + \rho \cdot S\theta_y \cdot C\theta_x \\ d_z = z_o - t_{oz} + a \cdot S\theta_y - b \cdot C\theta_y \cdot S\theta_x + \rho \cdot C\theta_y \cdot C\theta_x \end{cases} \quad (3.15)$$

La solution retenue est $\theta_x = 2 \cdot \arctan\left(\frac{\rho - \sqrt{\rho^2 - (y_o - t_{oy})^2 + b^2}}{y_o - t_{yo} + b}\right)$

3.2.3.3 Calibration de la tige du robot

La calibration de la tige du robot dépend des 6 paramètres suivants : $\rho, a, b, t_{ox}, t_{oy}$ et t_{oz} . Le système donné par (Éq. (3.14)) peut s'écrire sous la forme matricielle suivante :

$$\begin{bmatrix} -C\theta_x \cdot S\theta_y & C\theta_y & S\theta_x \cdot S\theta_y & 1 & 0 & 0 \\ S\theta_x & 0 & C\theta_x & 0 & 1 & 0 \\ -C\theta_x \cdot C\theta_y & -S\theta_y & S\theta_x \cdot C\theta_y & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \rho \\ a \\ b \\ t_{ox} \\ t_{oy} \\ t_{oz} \end{bmatrix} = \begin{bmatrix} x_o - d_x \\ y_o \\ z_o - d_z \end{bmatrix} \quad (3.16)$$

Donc la connaissance d'un point 3D dans le repère objet ainsi que les consignes moteurs correspondantes ($x_o, y_o, z_o, d_x, d_z, \theta_x, \theta_y$) donne 3 équations à 6 inconnues. Il faut donc 2 points au moins pour pouvoir estimer les 6 inconnues. Nous avons utilisé 4 points et la méthode des moindres carrés linéaire pour estimer ces paramètres, bien que d'autres méthodes d'optimisation peuvent être utilisées, par exemple la méthode de *Levenberg-Marquardt* (Shaheen, 1999). Mais les résultats que nous avons obtenus sont satisfaisants par rapport à l'objectif que nous nous sommes fixé (une précision inférieure au mm) Le tableau 3.1 montre les valeurs des paramètres obtenues par cette méthode.

Paramètres de calibration	Valeurs (mètre)
ρ	0.733103
a	0.019364
b	0.007175
t_{ox}	0.418756
t_{oy}	0.222311
t_{oz}	1.673854

TAB. 3.1 – Les paramètres obtenus lors de calibration de la tige du robot

3.3 Description du système ARITI

ARITI (“Augmented Reality Interface for Telerobotic applications via Internet”) (Otmane *et al.*, 1999) (Otmane *et al.*, 2000b) est un système de réalité augmentée permettant à un opérateur humain de superviser et de commander un robot via le réseau Internet. L’architecture du système a été ensuite améliorée pour supporter du Télétravail coopératif (plusieurs personnes se trouvant à des endroits différents peuvent participer et travailler ensemble pour réaliser une mission). La réalisation de ce système a fait l’objet d’une étude et d’une recherche prenant en compte des contraintes liées aux domaines de la Télérobotique (instabilité des systèmes en présence de délai) et de la Télécommunication (transmission de données informatiques).

De nombreuses questions se sont posées au niveau du choix du matériel à utiliser (Station de travail, PC, etc.), du système d’exploitation sur lequel doit s’exécuter le système de télétravail (Unix, Windows (NT, 95, 98) Linux (ou Linux temps réel), etc.), des outils de développement à utiliser (C, C++, JAVA (Java3D), VRML, OpenGL, etc.). Nous pouvons dire que le choix est assez difficile vu le nombre important de possibilités.

Pour répondre à toutes ces questions, et par rapport aux problématiques posées, nous avons jugé utile de prendre en considération les caractéristiques d’un système de télétravail idéal citées au début de ce chapitre.

Nous avons donc fait le choix suivant :

– **Matériel :**

Un simple ordinateur personnel (PC), de plus actuellement ces ordinateurs personnels sont de plus en plus puissants et à moindre coût. Il devient plus facile de changer un PC qu’une station de travail.

– **Système d’exploitation :**

Un système d’exploitation multi-tâche et multi-utilisateur “**Linux**”. Il devient plus facile de faire de la télé-maintenance sous l’environnement Linux (noyau configurable et système ouvert, protection des ressources, ...etc) que sous Windows (noyau et système fermé et n’est pas à l’abri des virus informatiques etc ...)

– **Outils de développement :**

C, C++ et JAVA ont été choisis pour le développement de notre système. C et C++ pour la partie serveurs (vidéo et commande) et JAVA pour la partie interface homme machine (client) dans le but d’avoir une interface portable (il s’agit de générer un code portable standard) pouvant s’exécuter depuis n’importe quelle machine connectée au réseau Internet.

3.3.1 Description du matériel

Le système ARITI est développé sur un PC Pentium 233 Mhz à 128 Mo de RAM. Il est relié d’un côté à une caméra noir et blanc via une carte d’acquisition vidéo Matrox Meteor. De l’autre côté, le PC est connecté à un robot via une liaison série RS232 (figure 3.7).

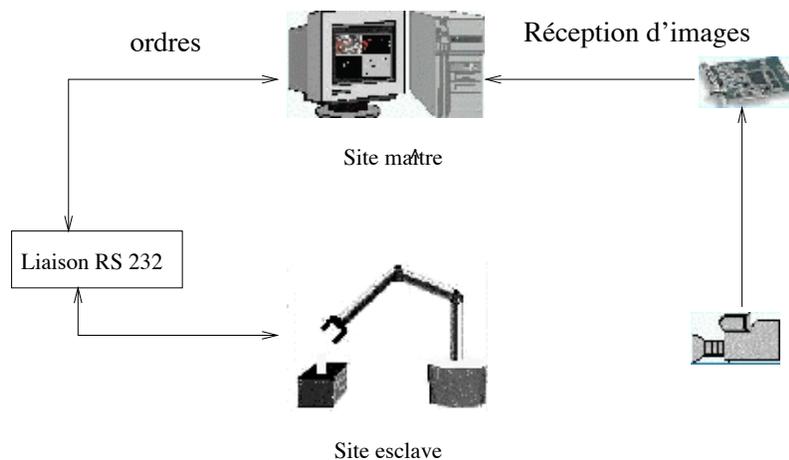


FIG. 3.7 – Illustration du matériel utilisé pour la réalisation du système ARITI

3.3.2 Description du logiciel

Le système ARITI est développé sous l'environnement Linux. Il peut être divisé en trois parties : Une partie pour le site opérateur, une autre pour le site distant et enfin le module de communication qui relie les deux sites.

3.3.2.1 Site distant (esclave)

Dans ce site se trouve un robot et une caméra vidéo ainsi que le module de communication serveur³. Ce dernier est constitué de deux serveurs, un pour les images vidéo et un autre pour les consignes robot. Le premier serveur se charge de capturer des images via la carte d'acquisition vidéo, de les compresser (voir les méthodes de compression en annexe A) et ensuite les envoie directement aux différents clients connectés. Le deuxième serveur quant à lui est chargé de récupérer les ordres envoyés par le client et les transmet directement au robot pour l'exécution (il peut se charger éventuellement de transférer des informations depuis le robot vers le client).

3.3.2.2 Site opérateur (maître)

Dans ce site⁴ se trouve un opérateur humain et une machine connectée à Internet (disposant d'un navigateur Internet ou bien ayant un interpréteur Java installé). Le premier niveau de description de cette partie se résume en deux grands modules. Un module de communication client qui permet l'échange d'informations avec le site esclave, et un module d'Interface Homme Machine (IHM) qui gère les interactions entre l'opérateur humain et le module de communication client. Ce dernier contient deux clients, un pour la réception d'images vidéo, décompression et la visualisation et l'autre se charge d'interpréter les actions de l'opérateur et ensuite les envoie sous forme d'ordres au site esclave (serveur de commandes) qui se charge de les transmettre au robot réel.

Le module d'IHM quant à lui permet à l'opérateur de spécifier le mode de travail désiré en rentrant en interaction avec un des trois modules : *Téléopération*, *Téléprogrammation*

3. L'implantation logicielle de ce serveur est faite en C++

4. L'implantation logicielle de ce site est faite entièrement en JAVA afin de satisfaire le critère de portabilité.

ou *Télé supervision et Télé coopération*. Chacun des trois modules utilise un module central appelé module *Réalité Mixée (RM)* qui combine les deux mondes virtuel et réel dans le but de fournir une IHM conviviale. Le module *RM* joue aussi le rôle d'intermédiaire entre les différents modes de travail et le module de communication client (figure 3.8).

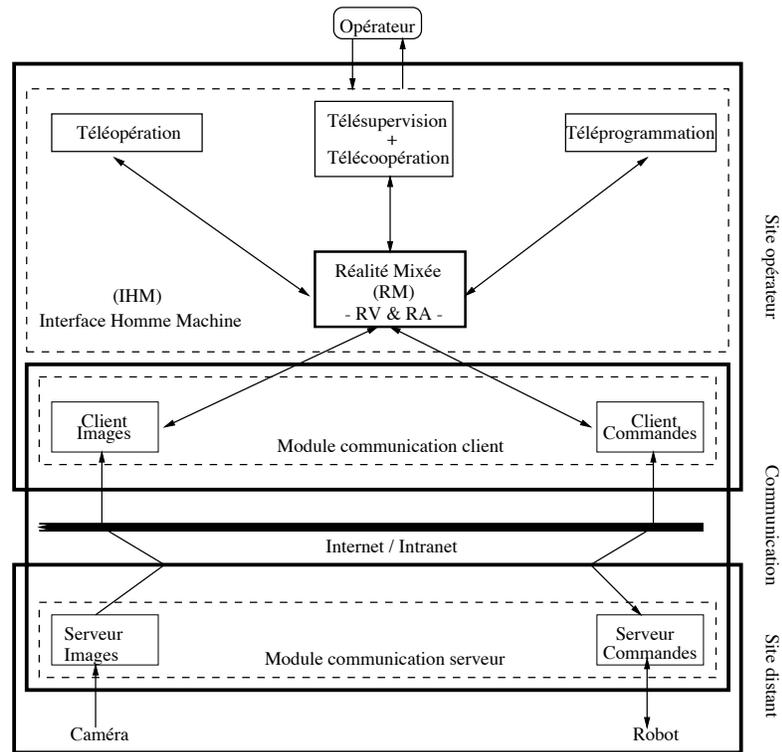


FIG. 3.8 – Architecture simplifiée du système ARITI

3.3.3 Module de Réalité Mixée (RM)

Comme son nom l'indique, ce module combine les deux techniques: La réalité virtuelle et la réalité augmentée. le mot *Réalité Mixée* de l'anglais *mixed reality* (Milgram et Kishino, 1994) (Milgram *et al.*, 1995) est utilisé généralement pour décrire tout système combinant le monde réel et le monde virtuel. Milgram et Kishino ont proposé le *continuum de la Réalité-Virtualité* montré dans la figure 3.9. Hickey et Manninen et Petri Pulli (Hickey *et al.*, 2000) ont introduit le concept de *TéléRéalité* à l'intérieur de ce continuum afin de représenter un monde réel dans lequel des données 3D sont ajoutées. A gauche de la figure 3.9 est représenté l'environnement réel (constitué des images 2D réelles) et à droite un environnement purement virtuel (constitué des données 3D). Le centre du continuum est un point dont lequel le contenu de l'environnement virtuel est identique au contenu de l'environnement réel. Bien que chaque partie de ce continuum ne peut être considérée que par abstraction, mais à nos yeux ce concept peut être utilisé pour déterminer le degré de *Téléprésence* d'un système.

Les composantes du module de *RM* peuvent être regroupées en deux grandes catégories: les données et les fonctions d'assistance.

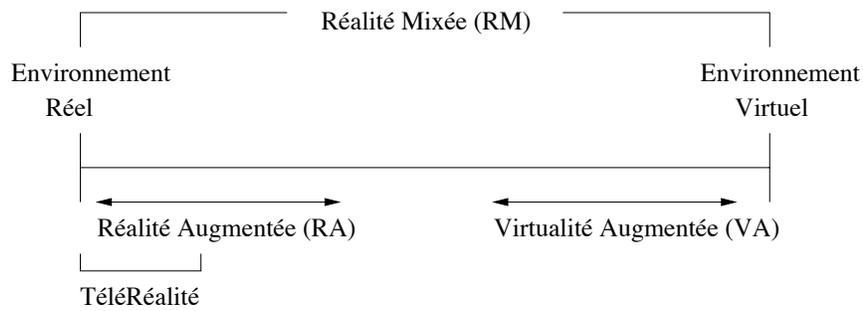


FIG. 3.9 – *Le continuum Réalité-Virtualité d'après Milgram 1994 et Hickey 2000*

3.3.3.1 Les Données

Il s'agit des données des environnements réel et virtuel. Les données de l'environnement réel sont généralement perçues par des capteurs se trouvant sur le site réel. Dans notre cas il s'agit des données 2D images vidéo (du robot et de son environnement) et des données articulaires du robot (position et orientation). L'environnement virtuel est constitué de données 3D pour la représentation graphique du robot et de son environnement (robot virtuel, objets virtuels) et contient aussi un certain nombre d'outils virtuels d'assistance tels que les guides virtuels. Ces deux types de données, réelles et virtuelles, sont combinés pour former un *Environnement Mixte (EM)* avec lequel l'opérateur travaillera (figure 3.10).

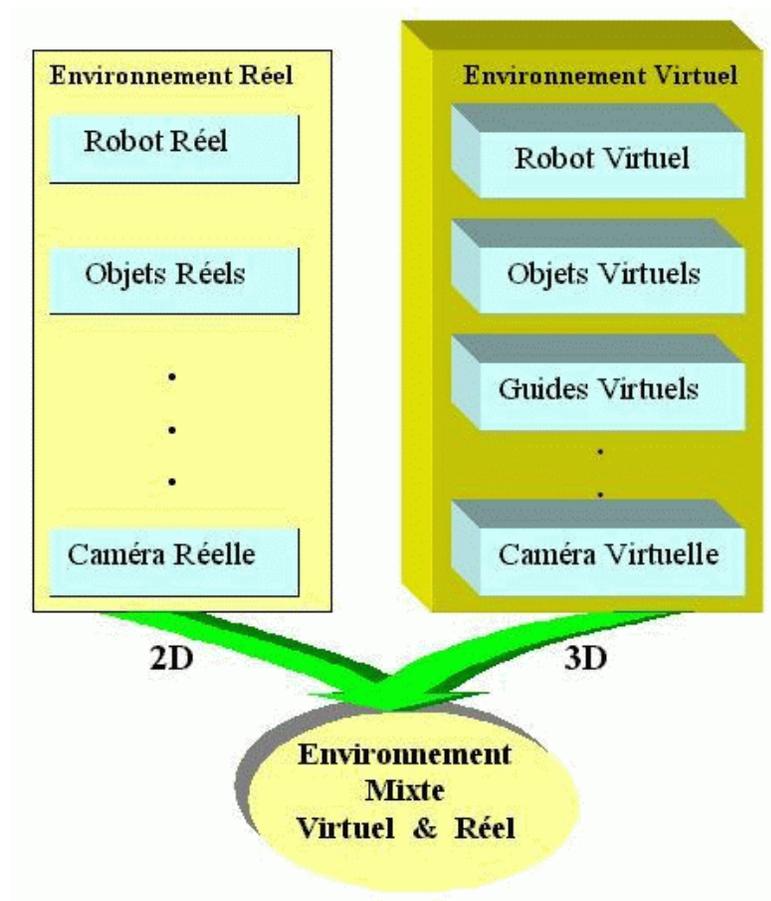


FIG. 3.10 – *Représentation des données réelles et virtuelles dans un environnement mixte*

3.3.3.2 Les fonctions d'assistance

Afin d'aider l'opérateur à télétravailler, il est utile de lui fournir trois types d'assistance, pour la perception de l'environnement, pour la commande du robot et pour la supervision de tâches. La figure 3.11 illustre les trois types d'assistance fournis à l'opérateur humain lors du télétravail.

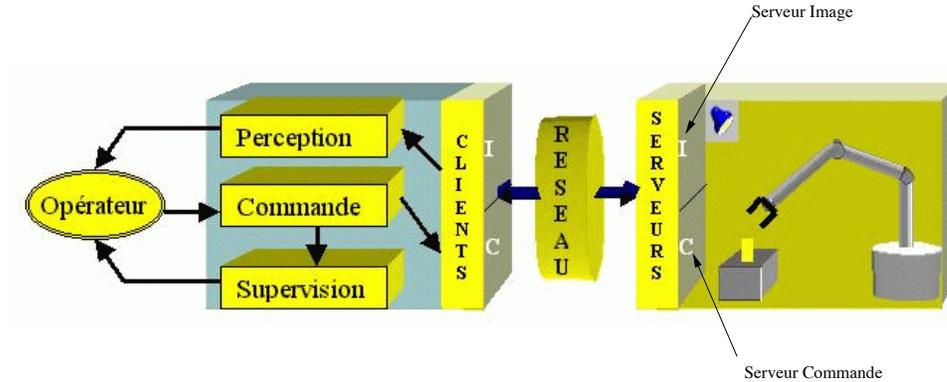


FIG. 3.11 – Les trois types d'assistance pendant le télétravail

❶ Assistance à la perception de l'environnement :

Il s'agit d'assistance à la perception d'environnement mixte. L'opérateur perçoit son environnement réel grâce au retour d'images vidéo d'une part et il perçoit son environnement virtuel moyennant plusieurs points de vue virtuels d'autre part (figure 3.12)

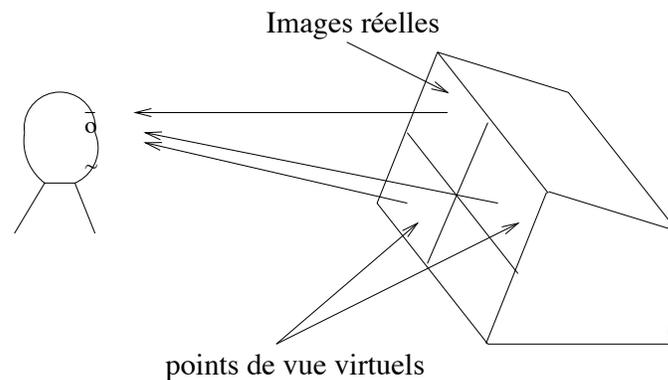


FIG. 3.12 – Illustration de l'assistance à la perception de l'environnement

❷ Assistance à la commande :

Il s'agit d'assister l'opérateur à la commande du robot réel distant. L'opérateur commande le robot réel par l'intermédiaire du robot virtuel (figure 3.13). Ceci lui permet de travailler en temps réel avec cette représentation virtuelle et ne subit pas les problèmes liés au délai de transmission dû à la distance qui sépare les deux sites (maître et esclave). Afin d'améliorer les performances de l'opérateur à la commande (en mode de téléopération), des guides virtuels sont utilisés. Ces derniers vont permettre à l'opérateur de travailler plus vite, d'être précis et surtout d'empêcher toute

erreur de téléopération involontaire.

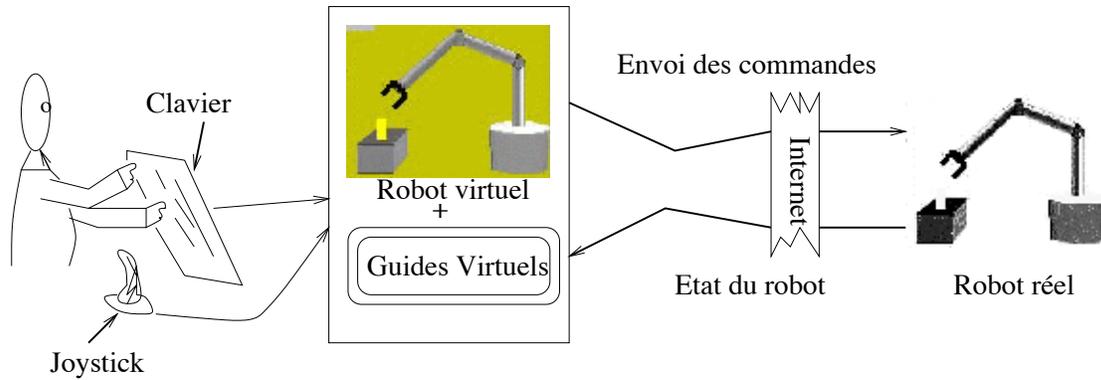


FIG. 3.13 – Illustration de l'assistance à la commande

③ Assistance à la supervision :

Il s'agit d'aider l'opérateur à vérifier le bon déroulement des tâches réalisées c'est-à-dire, vérifier si les tâches réalisées par le robot virtuel ont correctement été faites par le robot réel. En effet, la superposition permanente du monde virtuel sur le monde réel permet une confirmation de la bonne exécution d'une tâche. La présence d'un écart entre le monde virtuel et le monde réel, se traduit par des erreurs dues à l'exécution de la tâche par le robot réel. Dans ce cas l'opérateur doit désactiver la commande du robot réel et procéder au recalage du modèle virtuel sur l'image réelle (afin que les deux mondes virtuel et réel soient superposés à nouveau) et ensuite il activera la commande du robot réel (figure 3.14). Des informations textuelles telles que "Cible atteinte", "Objet saisi", "Collision détectée", ..., sont retournées à l'opérateur pour l'informer de l'état des opérations réalisées.

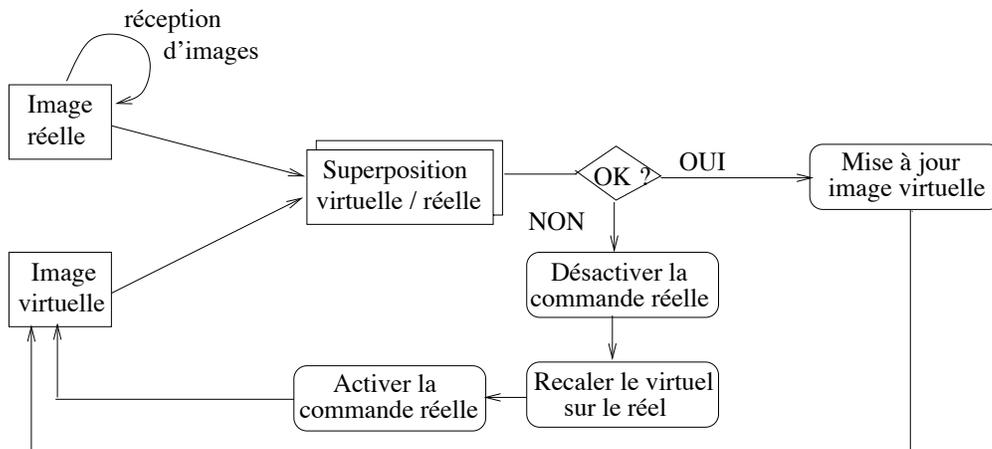


FIG. 3.14 – Organigramme d'assistance à la supervision

3.3.4 Module de Téléopération

Ce module est celui utilisé par défaut si l'opérateur n'a pas spécifié l'un des modes de télétravail : Téléprogrammation ou Télésupervision/Télécoopération. Il intercepte toutes

les actions réalisées par le robot virtuel et les transmet directement au robot réel via le module de communication. En effet l’opérateur travaille en temps réel avec une représentation intermédiaire qui est l’environnement virtuel (robot virtuel, objets virtuels, etc.).

Dans ce mode de travail, l’opérateur peut être assisté par des guides virtuels (cf. § 2.1.2) pour réaliser une tâche. Ces guides virtuels ont un rôle important en téléopération du fait qu’ils permettent d’une part d’éviter au robot réel toute mauvaise manipulation de la part de l’opérateur (erreurs involontaires dues à la fatigue ou tout simplement des erreurs d’inattention) et d’autre part de travailler vite avec moins d’imprécision.

Selon les tâches à réaliser, l’opérateur peut utiliser des guides virtuels déjà existants dans la base de données des guides, les modifier (changer leurs propriétés, les déformer, etc.), ou bien il peut créer ses propres guides virtuels grâce au module “Création guides”. Les figures 3.15 et 3.16 illustrent respectivement la téléopération sans et avec assistance à la commande.

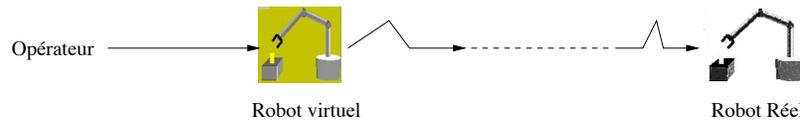


FIG. 3.15 – Schéma simplifié de la téléopération sans assistance à la commande

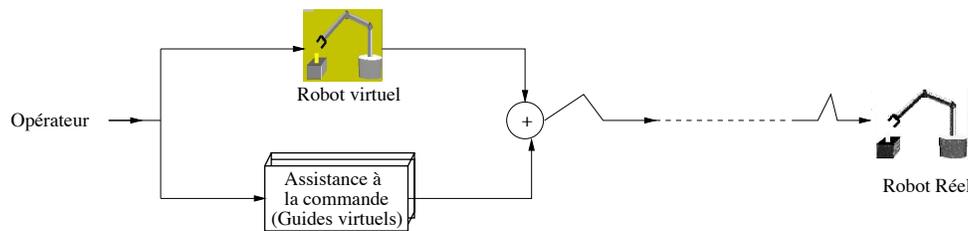


FIG. 3.16 – Schéma simplifié de la téléopération avec assistance à la commande

3.3.5 Module de Téléprogrammation

Le mode de travail “Téléprogrammation”, permet à l’opérateur de travailler dans un haut niveau d’abstraction. En effet, il s’agit de donner au robot virtuel (et au réel) des ordres de haut niveau pour réaliser une tâche. Il existe un certain nombre de tâches de téléprogrammation gérées par ce module, par exemple “Saisir objet n° 1”, “Déposer objet n° 1 sur le Support n° 2”, “Suivre le contour de l’objet n° 1”, ...etc (ces tâches sont décrites dans le chapitre Application cf. § 4.1).

Une autre tâche consistant à désigner une cible à l’écran et la faire suivre par les robots virtuel et réel est aussi gérée par ce module. Cette dernière tâche consiste à atteindre un point 3D de l’espace obtenu à partir de deux points 2D écran (figure 3.17). Afin de réaliser cette tâche, deux points de vue virtuels différents (deux fenêtres) sont utilisés. Nous décrivons dans ce qui suit le principe et la méthode utilisée pour déterminer un point 3D à partir de deux points 2D.

Soit (u_1, v_1) les coordonnées 2D du premier point obtenu par la désignation sur la première fenêtre et $M(4 \times 4)$ le modèle de la caméra graphique associé à cette fenêtre.

Soit (u_2, v_2) les coordonnées 2D du deuxième point obtenu par la désignation sur la

deuxième fenêtre et $M'(4 \times 4)$ le modèle de la caméra graphique associé à cette fenêtre. Soit (X, Y, Z) les coordonnées recherchées du point 3D dans l'espace.

La première désignation peut s'écrire sous forme :

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = M^{-1} \times \begin{bmatrix} s \cdot u_1 \\ s \cdot v_1 \\ s \\ 1 \end{bmatrix} \quad (3.17)$$

avec

$$M^{-1} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & 1 \end{bmatrix} \quad (3.18)$$

La deuxième désignation s'écrit sous la forme :

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = M'^{-1} \times \begin{bmatrix} s \cdot u_2 \\ s \cdot v_2 \\ s \\ 1 \end{bmatrix} \quad (3.19)$$

avec

$$M'^{-1} = \begin{bmatrix} x'_{11} & x'_{12} & x'_{13} & x'_{14} \\ x'_{21} & x'_{22} & x'_{23} & x'_{24} \\ x'_{31} & x'_{32} & x'_{33} & x'_{34} \\ x'_{41} & x'_{42} & x'_{43} & 1 \end{bmatrix} \quad (3.20)$$

avec l'équation Éq. (3.17), les coordonnées (X, Y, Z) du point 3D vérifient :

$$\begin{cases} X = (x_{11} \times u_1 + x_{12} \times v_1 + x_{13}) \times s + x_{14} \\ Y = (x_{21} \times u_1 + x_{22} \times v_1 + x_{23}) \times s + x_{24} \\ Z = (x_{31} \times u_1 + x_{32} \times v_1 + x_{33}) \times s + x_{34} \end{cases} \quad (3.21)$$

l'identification de Éq. (3.17) et Éq. (3.19) donne :

$$s = ((x_{14} - x'_{14}) \times B' + (x'_{24} - x_{24}) \times A') / (B \times A' - B' \times A)$$

avec

$$A = x_{11} \times u_1 + x_{12} \times v_1 + x_{13}$$

$$B = x_{21} \times u_1 + x_{22} \times v_1 + x_{23}$$

$$A' = x'_{11} \times u_2 + x'_{12} \times v_2 + x'_{13}$$

$$B' = x'_{21} \times u_1 + x'_{22} \times v_2 + x'_{23}$$

En réalité, ce point 3D représente le centre d'une sphère virtuelle mobile appelée *cible mobile* (peut être déplacée dans l'environnement virtuel). Ce concept peut être utilisé pour le suivi d'objets en mouvement. En effet le robot peut suivre cet objet si une cible mobile (guide virtuel) lui est attachée (voir le chapitre application cf. § 4.1).

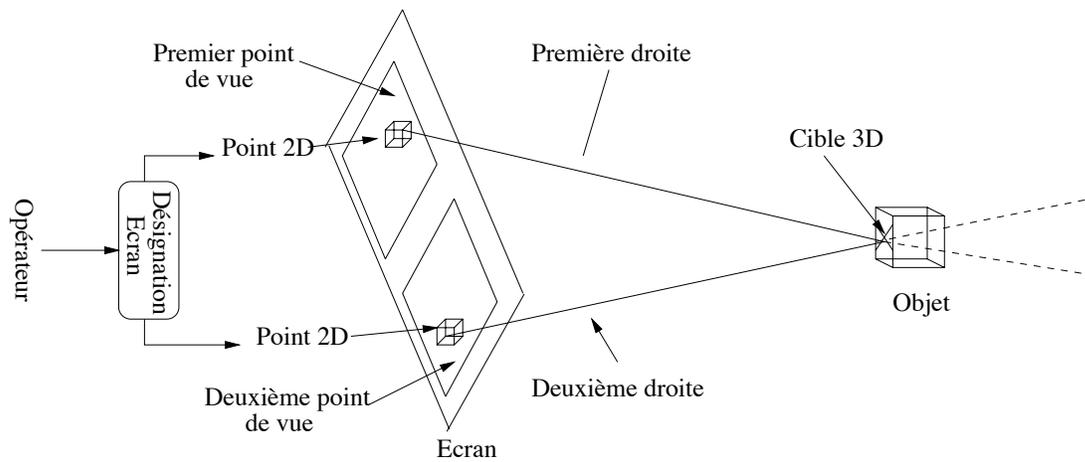


FIG. 3.17 – Illustration de la tâche “désignation d’une cible à l’écran”

En téléprogrammation de tâches, l’opérateur spécifie au système la tâche qu’il désire faire réaliser par le robot et lui demande de l’exécuter. Cette tâche (une séquence de consignes) est alors exécutée par le robot virtuel et par le robot réel simultanément (chaque consigne est générée en ligne pour le robot virtuel pour une simulation et ensuite envoyée au robot réel pour exécution). L’exécution du côté robot virtuel est synchronisée avec l’exécution du côté robot réel, ceci permet à l’opérateur s’il le désire, d’arrêter l’exécution de la tâche à n’importe quel moment (à une consigne près). La figure 3.18 illustre le principe de téléprogrammation de tâches.

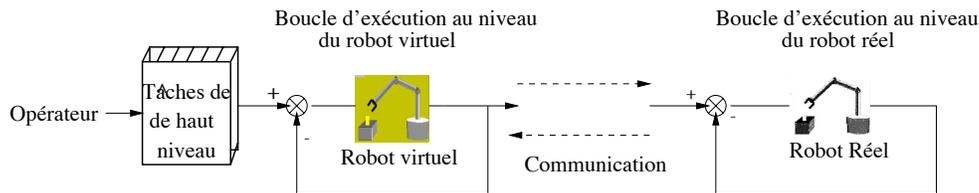


FIG. 3.18 – Schéma simplifié de l’exécution de tâches en mode Téléprogrammation.

3.3.6 Module de Télésupervision et Télécoopération

Ce module est surtout utilisé lorsque plusieurs opérateurs (clients) sont connectés au système ARITI. En effet en choisissant le mode “*Supervision*”, ces opérateurs peuvent observer l’exécution d’une tâche en réel et en virtuel simultanément. Cependant à un instant donné un seul opérateur est autorisé à commander le robot réel (il ne s’agit pas ici d’un problème technique mais d’un choix délibéré, en effet, on peut par exemple partager certains degrés de liberté du robot entre plusieurs opérateurs). Dans cette partie on définit un *client maître* comme étant le client ayant pris la commande du robot réel, et tous les autres seront appelés *superviseurs* (figure 3.19).

Le mot *Télécoopération* ou *Télétravail coopératif*, désigne la participation d’un ensemble de clients (opérateurs distants) à la préparation et à l’exécution d’une mission. Le système ARITI offre la possibilité à un ensemble d’opérateurs distants de travailler en coopération. En effet, considérons par exemple une mission M_i constituée de quatre tâches T_i (pour $i = 1..4$) qui s’exécutent d’une manière séquentielle (T_1 ensuite T_2 , T_3 et T_4). Soit PT_i la fonction “Préparation de la Tâche “ T_i ” et ET_i la fonction “Exécution de

(pour $i = 2..4$ et $j = 1..i - 1$) et inversement lorsqu'un opérateur a fini l'exécution d'une tâche il peut superviser la suite de la mission qui sera réalisée par les autres opérateurs. En considérant que le temps de la prise en main du robot réel soit $\delta_{i_i} \ll te_i$ (pour $i = 1..4$) pour chaque opérateur OP_i (le temps nécessaire pour libérer la commande du robot réel par le client maître plus le temps de la prise de la commande par un autre client) alors le temps total de l'exécution de la mission en télécoopération est

$$tmc = \left(\sum_{i=1}^4 tp_i\right)/4 + \sum_{i=1}^4 te_i \tag{3.23}$$

ce qui donne $tmc < tm$.

Cette exemple illustre l'avantage que peut offrir un travail coopératif pour la réalisation des missions complexes. En effet, la charge de travail induite par la complexité de la mission est partagée entre plusieurs opérateurs et le temps de réalisation de la mission est nettement réduit.

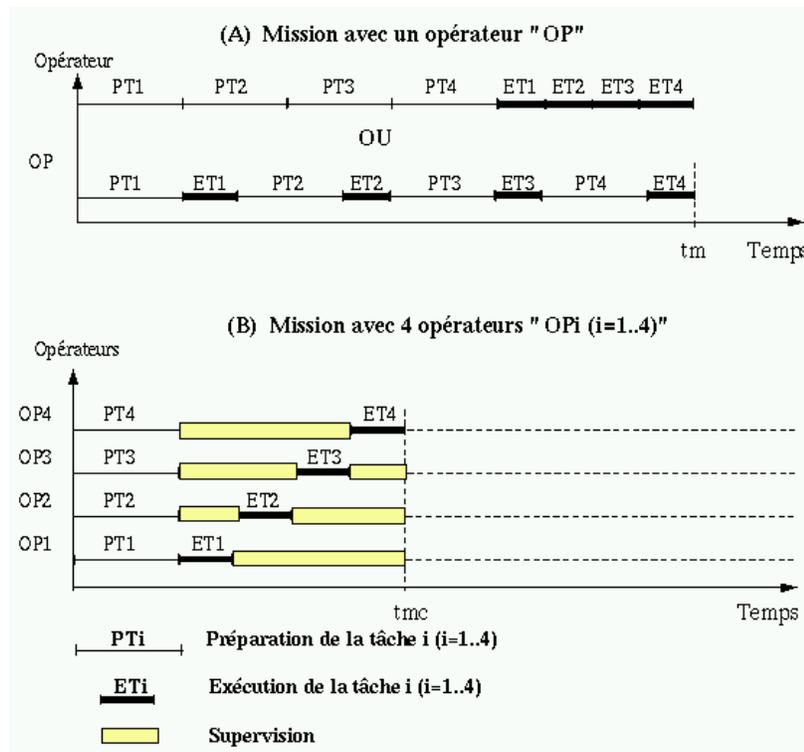


FIG. 3.20 – Exemple de préparation et d'exécution d'une mission, (A) : préparation et exécution par un seul opérateur, (B) : Télétravail coopératif avec 4 opérateurs.

3.4 Architecture réseau du système ARITI

Dans cette section nous présentons le premier niveau de l'architecture réseau du système ARITI (figure 3.21), ensuite nous montrons comment les différents modules (client/serveur) échangent des informations afin d'assurer une bonne communication. Il s'agit ici de décrire le protocole de communication entre les serveurs (images et commandes) et les clients correspondants.

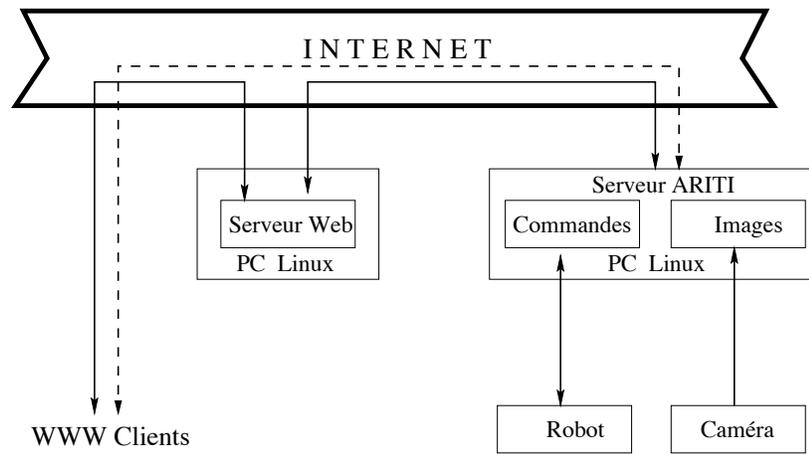


FIG. 3.21 – Architecture générale de communication avec le système ARITI : En pointillés, une possibilité d'accéder directement au serveur ARITI sans passer par le serveur Web

Le serveur image permet d'enregistrer une image capturée par la caméra, puis de l'envoyer, via le protocole TCP/IP, dans un format compressé. Il se peut aussi que le serveur envoie les modifications de l'image (différence par rapport à l'image précédente) pour diminuer la taille du fichier compressé par exemple. Le serveur de commande quant à lui permet de récupérer des commandes envoyées par le client et les transmet au robot réel pour exécution.

Pour pouvoir effectuer une requête au serveur (réception d'images, envoi de commandes), le client doit avant tout être identifié par un numéro qu'il obtient en faisant une demande au serveur (images ou commandes) (en envoyant un caractère "0" par exemple) (figure 3.22). Le nombre de clients maximum est 255 (car ce numéro de client est codé sur un octet allant de 1 à 255, il peut être augmenté si le besoin se fait sentir).

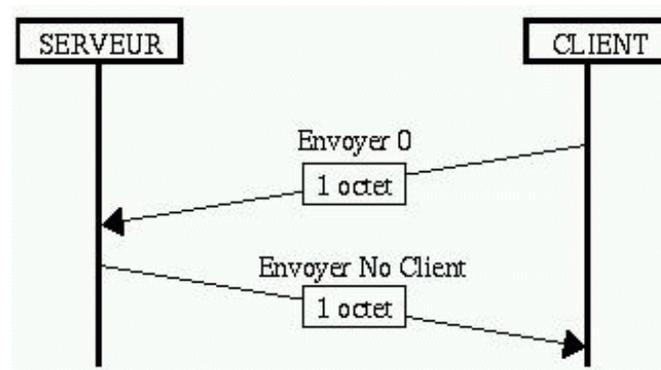


FIG. 3.22 – Protocole de demande d'un numéro d'identification

3.4.1 Protocole de communication client/serveur image

Dans cette partie, nous allons décrire le principe d'une demande d'une image vidéo par un client et les différents niveaux des schémas de fonctionnement du serveur image.

3.4.1.1 Demande d'une image vidéo

Dès que le client a récupéré son numéro, il le renvoie au serveur pour obtenir une image ou les modifications de l'image en format compressé (figure 3.23). En effet l'image vidéo capturée par le serveur peut subir des pré-traitements et des compressions différentes. Le client est informé du type du pré-traitement et/ou de la compression réalisée sur l'image afin de la décompresser correctement. Nous appelons pré-traitement la différence entre deux images I et I_{-1} (I_{-1} étant l'image déjà envoyée au client et I représente l'image capturée). Avant d'envoyer une image au client, le serveur envoie en premier un octet représentant le type de compression de l'image.

Nous présentons ici le codage des différents types de compression d'images envoyés au client:

- **“0” : Erreur** : Aucune image n'est envoyée, dans le cas où on ne désire pas l'envoyer à un client donné par exemple.
- **“1” : Image Brut** : L'image est envoyée sans compression.
- **“2” : Compression Gzip** : Un format de compression qui utilise la librairie “Zlib” supportée par la plupart des navigateurs Internet. Ce type de compression est généralement utilisé pour envoyer la première image aux clients. (voir annexe A pour plus d'information sur cette méthode).
- **“3” : Compression Gzip avec pré-traitement** : L'image subit un pré-traitement avant d'être compressée par Gzip. Ce format est utilisé généralement lorsqu'il y a beaucoup de mouvements dans l'image (voir chapitre résultat pour plus de détails).
- **“4” : Compression Huffman** : C'est un algorithme de compression rapide et qui fournit une bonne compression dans le cas où il y a peu de mouvements dans l'image.
- **“5” : Compression Huffman avec pré-traitement** : L'image subit un pré-traitement avant d'être compressée avec Huffman. Ce format permet d'avoir un taux de compression de l'image meilleur que le format Gzip avec pré-traitement. Il est utilisé dans le cas où il y a peu de mouvements dans l'image.

3.4.1.2 Schéma fonctionnel du serveur image

Le serveur image est composé de 3 modules: **“initialisation”**, **“attente d'une commande”** et **“gestion de la commande avec le client”**. Le schéma de fonctionnement simplifié du serveur image est donné par la figure 3.24.

Le module **“initialisation”** permet la création d'une socket TCP/IP, l'attachement de la socket sur un port connu et l'ouverture du service (figure 3.25 partie 1).

Le module **“attente d'une commande”** accepte la connexion avec le client et réceptionne son numéro d'identification (figure 3.25 partie 2).

Le module **“gestion de la commande avec le client”** peut être divisé en deux parties suivant le numéro du client :

Si le numéro vaut **“0”**, dans ce cas, le serveur cherche un numéro de libre (entre 1 et 255 par exemple). Si un numéro de libre est trouvé alors il est envoyé au client, sinon le

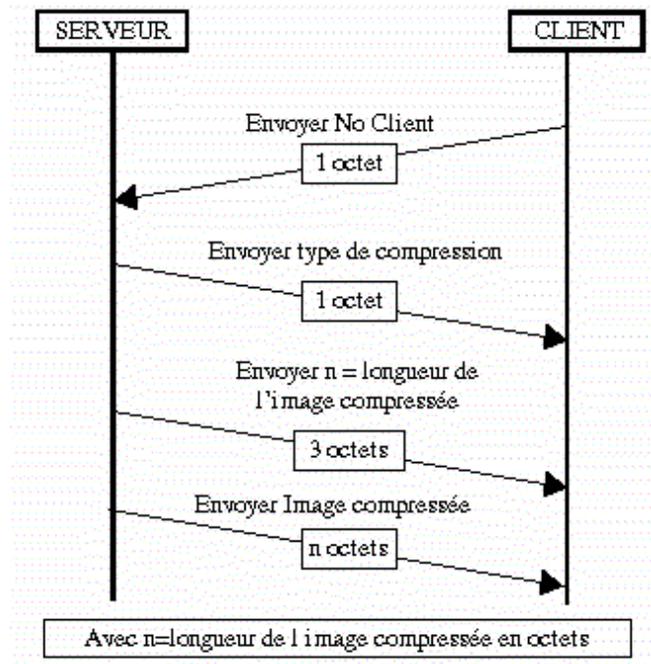


FIG. 3.23 – *Protocole de communication client/serveur image*

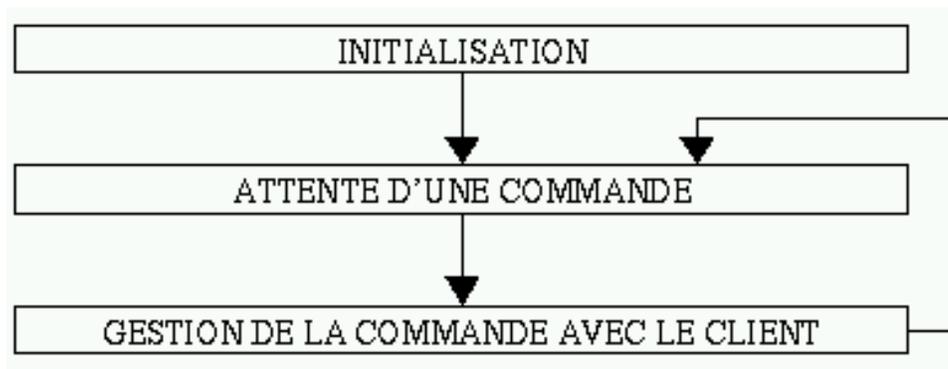


FIG. 3.24 – *Schéma de fonctionnement simplifié du serveur image*

serveur renvoi le numéro “0” pour informer le client qu’il ne peut pas recevoir d’images pour l’instant (figure 3.25 partie 3-A).

Si le numéro est supérieur à “0” (le client dans ce cas est autorisé à recevoir des images), alors le serveur prend l’image en cours, la compresse et l’envoi au client (figure 3.25 partie 3-B).

Dans la figure 3.25, est présenté le schéma de fonctionnement du serveur image, avec en caractères gras, les fonctions réseaux C/C++ utilisées.

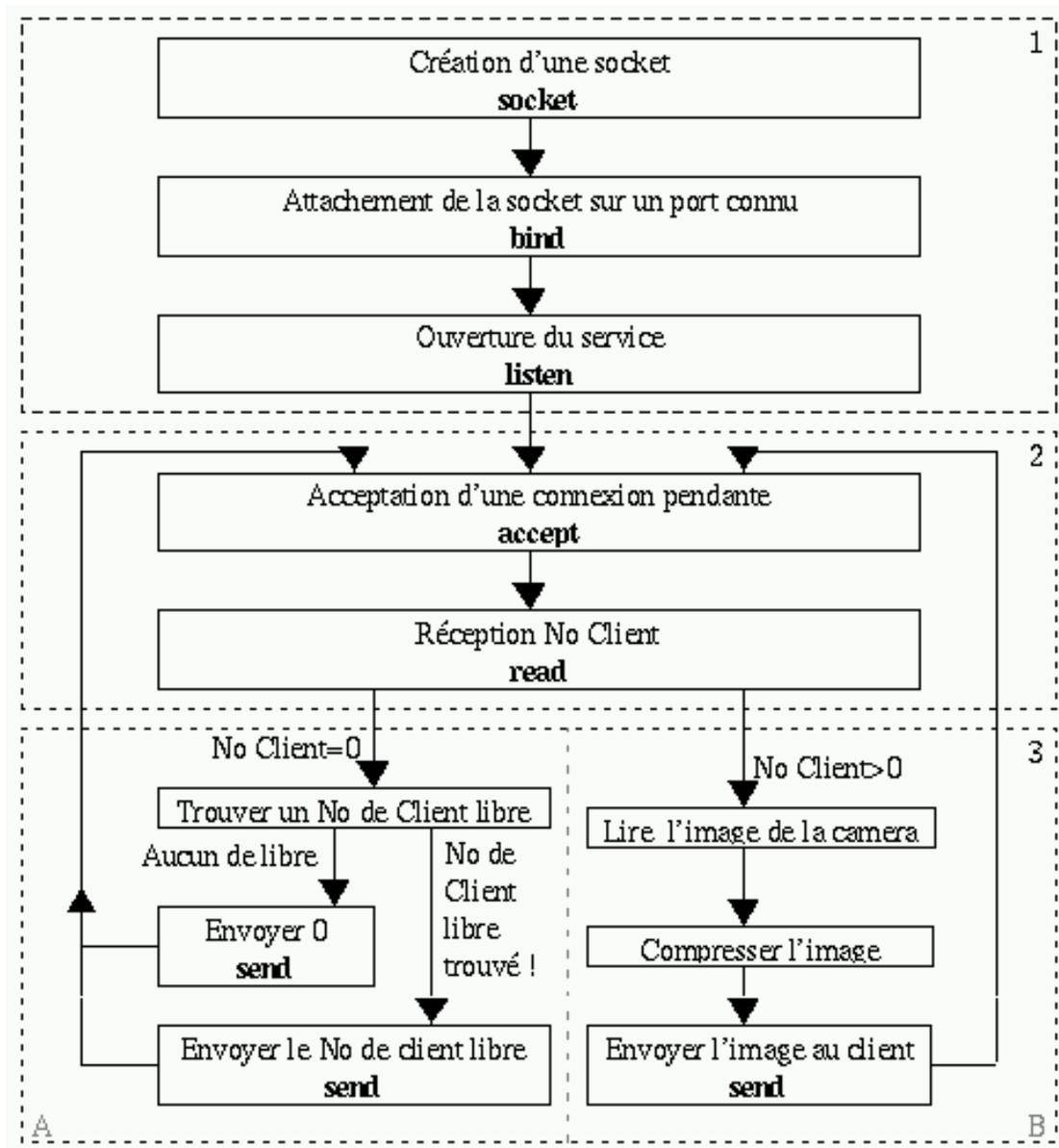


FIG. 3.25 – Schéma détaillé du fonctionnement du serveur image

3.4.2 Protocole de communication client/serveur commande

3.4.2.1 Protocole de connexion pour la commande du robot réel

Pour pouvoir contrôler le robot, il faut qu'aucun autre client ne l'utilise. Si le robot n'est pas libre, le serveur envoie le caractère "U" (pour dire que le robot est en cours d'utilisation) et interrompt la connexion. Sinon le serveur vérifie le mot de passe (password): s'il est bon, il autorise le contrôle du robot en envoyant le caractère "B" (pour dire que le mot de pass est bon), sinon le serveur envoie le caractère "M" (pour dire que le mot de passe est mauvais) et interrompt la connexion. La fin d'utilisation du robot se fait à la coupure de la connexion TCP/IP. La figure 3.26 illustre ce protocole de communication client/serveur commande.

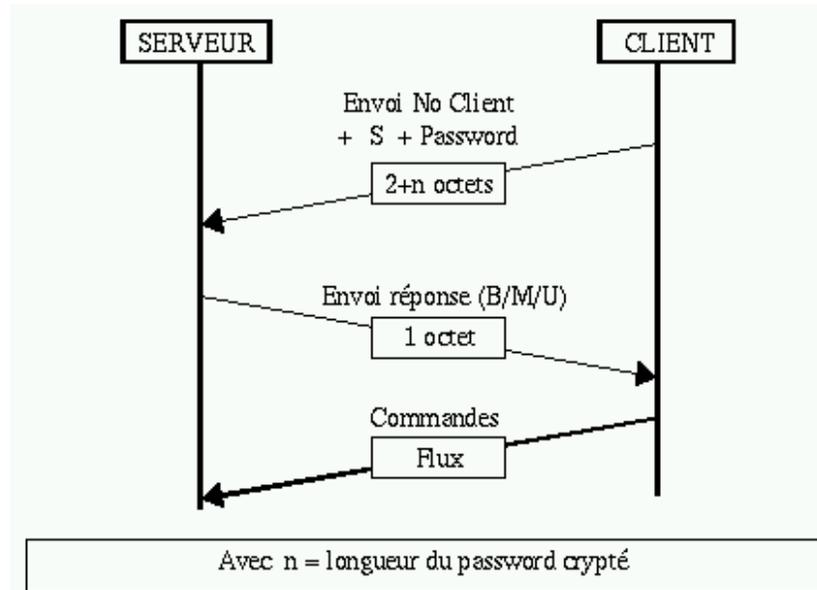


FIG. 3.26 – *Protocole de communication client/serveur commande. Le caractère "S" désigne la demande de contrôle du robot*

3.4.2.2 Protocole de télésupervision

Ce protocole est généralement utilisé lorsque un ou plusieurs opérateurs utilisent le mode de télésupervision et télécoopération. En effet, le serveur de commande se charge de renvoyer l'état final du robot et d'autres informations utiles à tous les opérateurs ayant choisi ce mode de télétravail.

Par ailleurs, ces opérateurs devraient voir en permanence la correspondance du robot virtuel avec le réel pendant toute la durée de contrôle du robot réel par le client maître. La figure 3.27 illustre ce protocole de communication entre le serveur commande et les clients dits superviseurs lorsque le client maître contrôle un robot à 4 degrés de libertés (ou PX et PY désignent deux rotations et PW et PZ les deux translations)

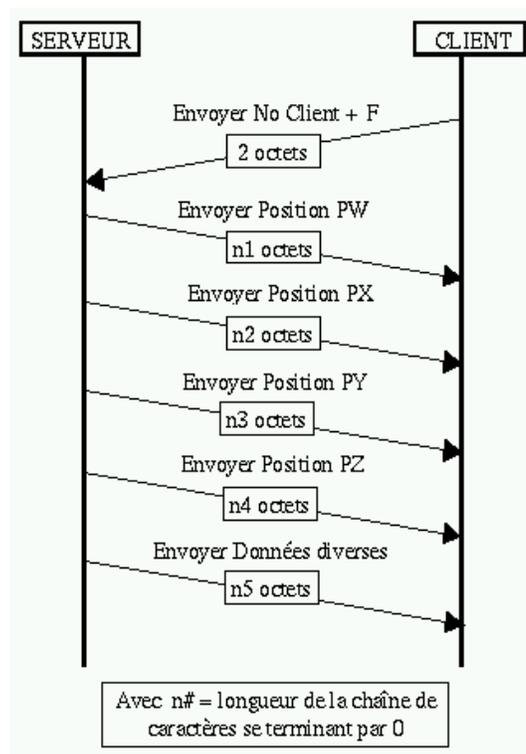


FIG. 3.27 – Protocole de communication entre le serveur de commande et les autres clients superviseurs : le caractère “F” désigne la demande de l’état final du robot

3.4.2.3 Schéma fonctionnel du serveur commande

De la même façon que le serveur image, le serveur commande peut être décomposé en 3 modules : “**initialisation**”, “**attente d’une commande**” et “**gestion de la commande avec le client**” voir la figure 3.24.

Le module “**initialisation**” est le même que celui du serveur image et il est représenté dans la figure 3.28 partie 1.

Le module “**attente d’une commande**”, détermine si un client se connecte pour la première fois (nouvelle connexion) ou s’il est déjà connecté et désire contrôler le robot réel. Dans le cas d’une nouvelle connexion, ce module réceptionne le numéro du client et vérifie si ce numéro est supérieur ou égale à zéro. Si ce numéro est supérieur à zéro, alors ce module réceptionne la demande de supervision ou de contrôle du robot envoyée par le client. Dans le cas où le client désire contrôler le robot réel, ce module se charge de récupérer les consignes du robot. Ce module est représenté dans la figure 3.28 partie 2.

Le module “**gestion de la commande avec le client**” peut être divisé en quatre sous modules **A**, **B**, **C** et **D** représentés par la figure 3.28 partie 3. Nous décrivons ci-dessous le fonctionnement de ces différents sous modules.

- **Le sous module A** : Alloue un numéro d’identification libre au client (figure 3.28 partie 3-A).

- **Le sous module B**: Enregistre les consignes envoyées par le client et les envoie ensuite au robot réel pour exécution (figure 3.28 partie 3-B).
- **Le sous module C**: Vérifie si le robot est libre (n'est pas pris par un autre client), dans ce cas, il procède à la vérification du mot de passe. Si le mot de passe est correct alors il envoie au client une autorisation pour contrôler le robot (et la connexion avec le robot est établie) sinon il informe le client que le mot de passe est mauvais. Dans le cas où le robot est en cours d'utilisation par un autre client, il informe le client que le robot est déjà pris (figure 3.28 partie 3-C).
- **Le sous module D**: Envoie la position finale du robot suite à une demande de supervision de la part d'un client (figure 3.28 partie 3-D).

Le schéma détaillé du fonctionnement du serveur de commande est donnée par la figure 3.28, avec en caractères gras, les fonctions réseaux C/C++ utilisées:

3.4.3 Limites et Sécurité du protocole

Le serveur peut être saturé rapidement si un client est défaillant (il demande toujours un nouveau No de client). Une solution serait de limiter le nombre de clients par adresse IP (non intégrée).

Le client (sauf s'il contrôle le robot) est déconnecté pour le serveur si il n'a pas envoyé de commande après un certain laps de temps (par défaut: 60 secondes). Dans ce cas, le numéro de ce client est libéré.

Chaque numéro de client est lié à une adresse IP, donc un client ne peut pas perturber un autre client qui se trouve sur une autre machine (au cas où un client défaillant utilise un No de client déjà pris par un autre!).

Le mot de passe est crypté pour éviter qu'il soit facilement lu sur le réseau. Le cryptage ne garantit pas à cent pour cent la confidentialité du mot de passe.

3.5 Bilan

Dans ce chapitre, nous avons présenté les méthodes et les outils nécessaires pour un contrôle en réalité augmentée, à savoir la modélisation de l'environnement et la calibration de la caméra et du robot. Nous avons aussi présenté le système expérimental de télétravail baptisé ARITI. L'interface de ce système est implantée en langage JAVA dans le but de la rendre portable et indépendante des systèmes d'exploitation sur lesquels elle va s'exécuter. Nous avons insisté sur des détails de l'architecture client/serveur réalisée afin de montrer d'un côté, sa complexité et de l'autre côté, la faisabilité conceptuelle et technique d'une architecture destinée à supporter un télétravail d'une part et un travail coopératif d'autre part.

Par ailleurs, nous avons présenté les trois modes de télétravail, à savoir la téléopération, la téléprogrammation de tâches et la télécoopération/télé supervision. Ce dernier mode uti-

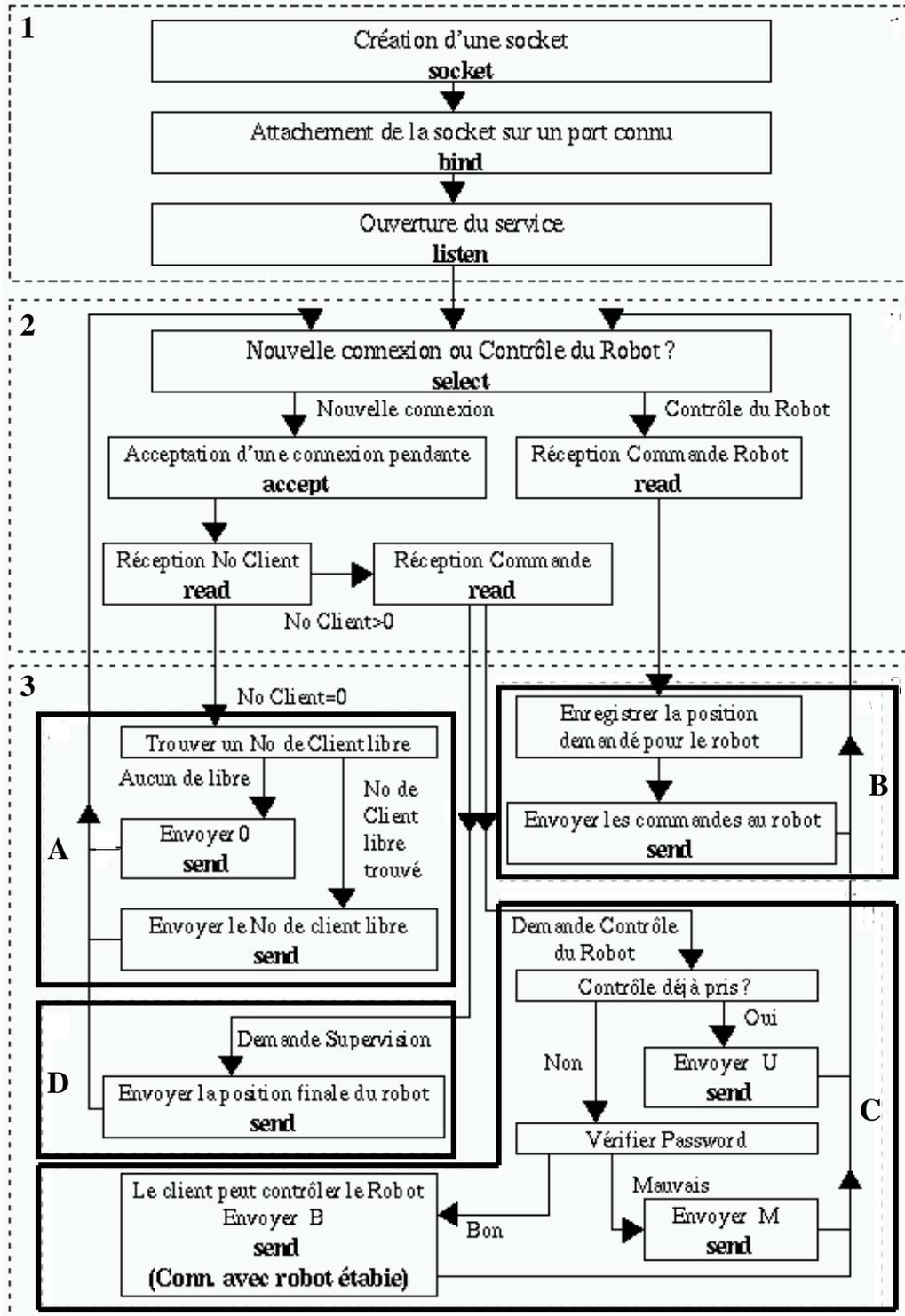


FIG. 3.28 – Schéma détaillé du fonctionnement du serveur de commande

lise un retour prédictif distribué, afin de permettre aux différents opérateurs de percevoir à la fois, les images vidéo de l'environnement réel distant et la mise à jour 3D de l'environnement virtuel au niveau de chaque client superviseur.

Dans le chapitre suivant, nous allons montrer quelques applications potentielles et en particulier pour la téléopération via Internet, d'un robot à 4 degrés de libertés utilisant le système ARITI.

Chapitre 4

Applications

Le chapitre précédent nous a permis de présenter les différentes méthodes géométriques utilisées pour un contrôle en réalité augmentée ainsi que le système expérimental de télétravail proposé. Nous avons cependant, introduit les différents modes de télétravail, téléopération, téléprogrammation de tâches et la télécoopération/télesupervision.

Dans ce chapitre nous présentons une application de télémanipulation, une autre de téléopération d'un robot mobile et les différentes applications potentielles pouvant bénéficier du système de télétravail décrit dans le chapitre précédent.

La première application est celle réalisée avec un banc d'essai à 4 degrés de libertés transformé en robot téléopérable via Internet grâce au système ARITI. En effet, nous décrirons les tâches réalisées suivant les différents modes de télétravail ainsi que les guides virtuels que nous avons utilisé pour réaliser quelques tâches en téléopération. Nous présentons l'interface homme-machine pour cette application ainsi que le modèleur 3D que nous avons réalisé pour créer et manipuler ces guides virtuels.

Nous présentons ensuite une application d'un projet en cours de réalisation pour l'assistance aux personnes handicapées (un projet en collaboration avec l'Association Française contre la Miopathie (AFM)) utilisant un robot mobile.

Ce chapitre s'achève par la description d'autres domaines d'applications où l'utilisation d'un tel système de télétravail peut être intéressante.

4.1 Télétravail avec un robot à 4 ddl

4.1.1 Description du site esclave

Dans ce cas, le site esclave est constitué d'un robot, d'une zone de manipulation et d'une caméra (figure 4.1).

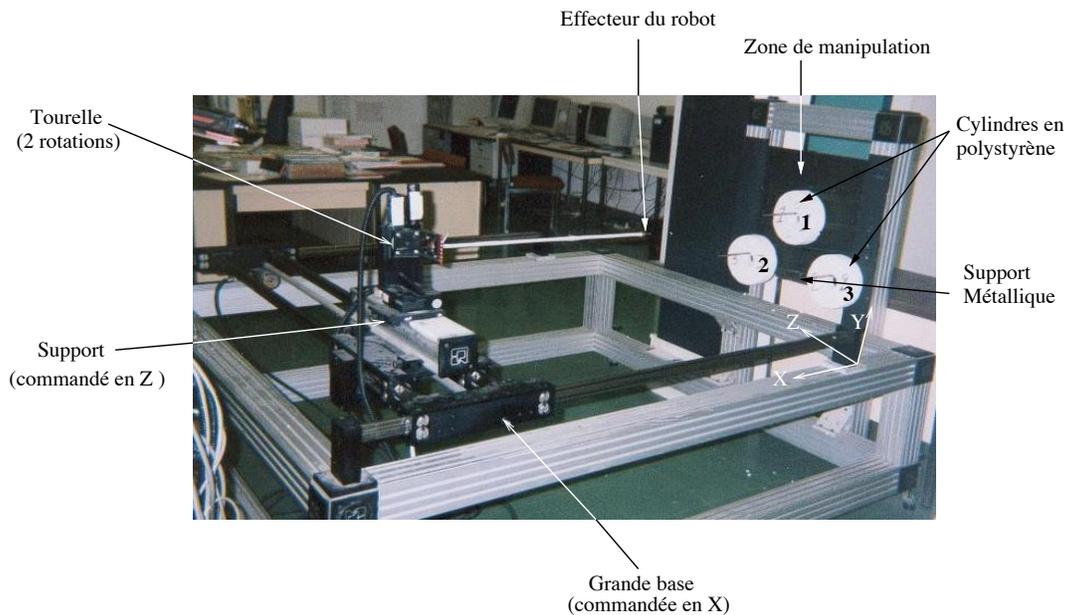


FIG. 4.1 – Image du robot et de la zone de manipulation.

Le robot utilisé est un banc d'essai à 4 ddl, il peut effectuer deux translations parallèles au sol et deux rotations en site et azimut. Chaque composante du robot est commandée par un moteur pas à pas via une armoire de commande (figure 4.2 (a)), le robot peut aussi être commandé par une liaison série RS232 en positionnant l'aiguille ("switch") en mode réseau.

La zone de manipulation est constituée de 3 cylindres en polystyrène numérotés (1,2 et 3) de 20 cm de diamètre accrochés sur des supports métalliques (figure 4.1). La figure 4.3 montre les caractéristiques de l'objet et de son support. Ces supports sont fixés à une mire métallique située à 1 mètre environ de l'effecteur du robot.

La visualisation du robot et de son environnement est réalisée par une caméra noir et blanc (focale de 8,5 mm) montée sur une tourelle (figure 4.2 (b)).

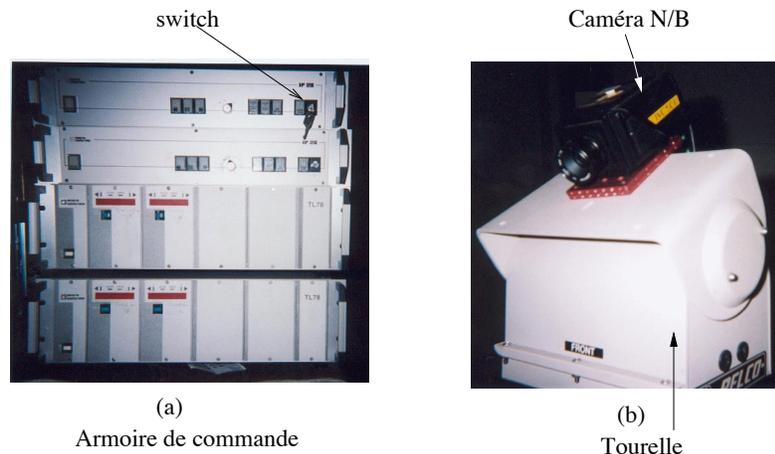


FIG. 4.2 – Image de l'armoire de commande (a) et de la caméra montée sur une tourelle (b)

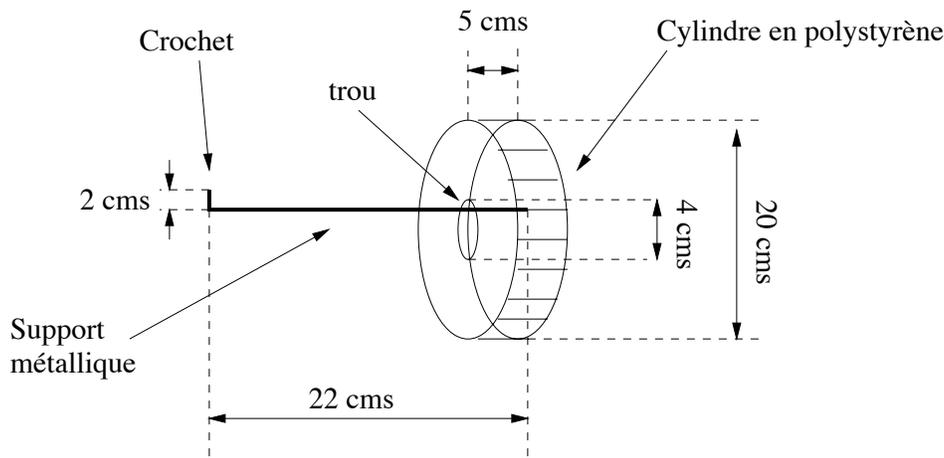


FIG. 4.3 – Caractéristiques d'un objet à manipuler et de son support

4.1.2 Description du site client

Le site client est généralement constitué d'un opérateur et d'une machine connectée à Internet (ordinateur ou une Station de travail disposant d'un navigateur Internet).

Comme pour tout système accessible sur réseau en particulier Internet, et vue le nombre important de clients qui peuvent se connecter en même temps, il est nécessaire d'avoir un administrateur pour ce type de système. En effet, dans notre cas, la création de guides virtuels par exemple, ne peut pas se faire par plusieurs clients connectés au système ARITI en même temps (pour éviter des conflits d'accès aux ressources du système). Ne peut créer des guides virtuels que le client qui s'est connecté en tant qu'administrateur. Les autres clients, peuvent les utiliser pour réaliser des tâches en téléopération.

Nous définissons alors deux types d'opérateurs (clients) :

- **Administrateur** : Est un client pouvant travailler avec une version de ARITI réservée uniquement pour l'administrateur. Ce dernier réalise la mise à jour de la version d'ARITI destinée aux autres clients.
- **Utilisateurs ou superviseurs** : Se sont des clients pouvant travailler avec une version de ARITI mise à jour par l'administrateur.

La figure 4.4 illustre ces deux types de clients pouvant télétravailler avec les deux types de versions du système ARITI.

Il existe deux versions d'Interface Homme Machine (IHM) permettant à un opérateur de télétravailler. La première version, est appelée "**IHM administrateur**" et la deuxième "**IHM utilisateur**".

4.1.2.1 IHM administrateur

Cette version est destinée à l'opérateur (client) dit "administrateur" qui a pour rôle la mise à jour éventuelle et l'administration du système ARITI. En effet, l'administrateur peut créer des guides virtuels d'assistance à la téléopération, les tester et mettre à jour la base de données des guides (comme indiqué dans les cas d'utilisation des guides virtuels au chapitre cf. § 2.3 et le tableau 2.3). Il peut aussi interdire complètement l'accès au système, ou partiellement, comme par exemple limiter le contrôle du robot réel à certains

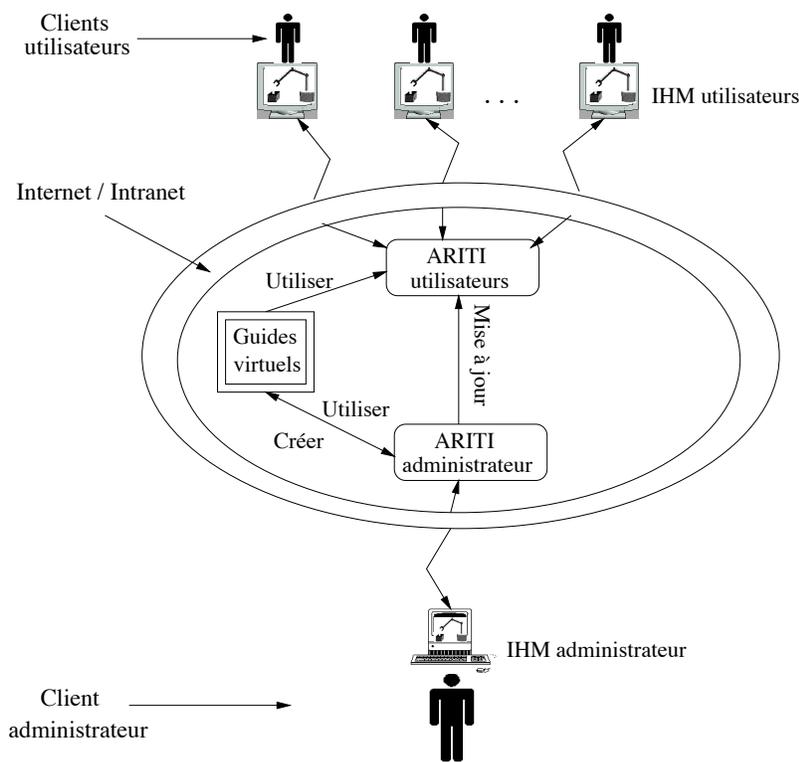


FIG. 4.4 – Illustration des deux types de clients “Administrateur” et “Utilisateur” du système ARITI

clients autorisés.

L’administrateur, peut aussi créer de nouvelles tâches de téléprogrammation et les mettre ensuite à la disposition des autres clients utilisateurs du système. En d’autres termes, L’IHM administrateur peut être considérée comme une version expérimentale avec laquelle l’opérateur peut créer de nouveaux modèles (robots, objets et guides virtuels) et les tester avant de les mettre à la disposition des utilisateurs.

La figure 4.5 montre l’interface destinée au client administrateur.

Nous décrivons dans le paragraphe qui suit l’interface du système ARITI. Cette description est commune aux deux versions de l’interface “administrateur” et “utilisateur”. L’interface du système ARITI est constituée de quatre parties. La première, en haut à gauche de la figure 4.5, permet le contrôle en réalité augmentée du site esclave (l’opérateur perçoit d’un côté l’environnement distant grâce au retour vidéo ainsi que la superposition du monde virtuel sur le réel et de l’autre côté, il perçoit une horloge lui indiquant le délai de transmission vidéo et le décalage horaire si le client se trouve à l’extérieur de la France).

La deuxième partie de l’interface (en haut à droite de la figure 4.5) est complètement virtuelle, elle permet de visualiser l’environnement virtuel grâce à une caméra virtuelle que l’on peut déplacer ou tourner suivant les 3 axes X, Y et Z.

La troisième partie (en bas à gauche de la figure 4.5) permet à l’opérateur de voir ce que voit une caméra virtuelle fixée sur l’effecteur du robot (eye-in-hand).

La quatrième partie (en bas à droite de la figure 4.5) représente le tableau de bord de l’interface de ARITI. En effet, elle permet à l’opérateur de choisir son mode de télétravail (téléopération, téléprogrammation de tâche ou encore télésupervision et télécoopération). Elle permet aussi à l’opérateur de changer de point de vue de la caméra virtuelle, de

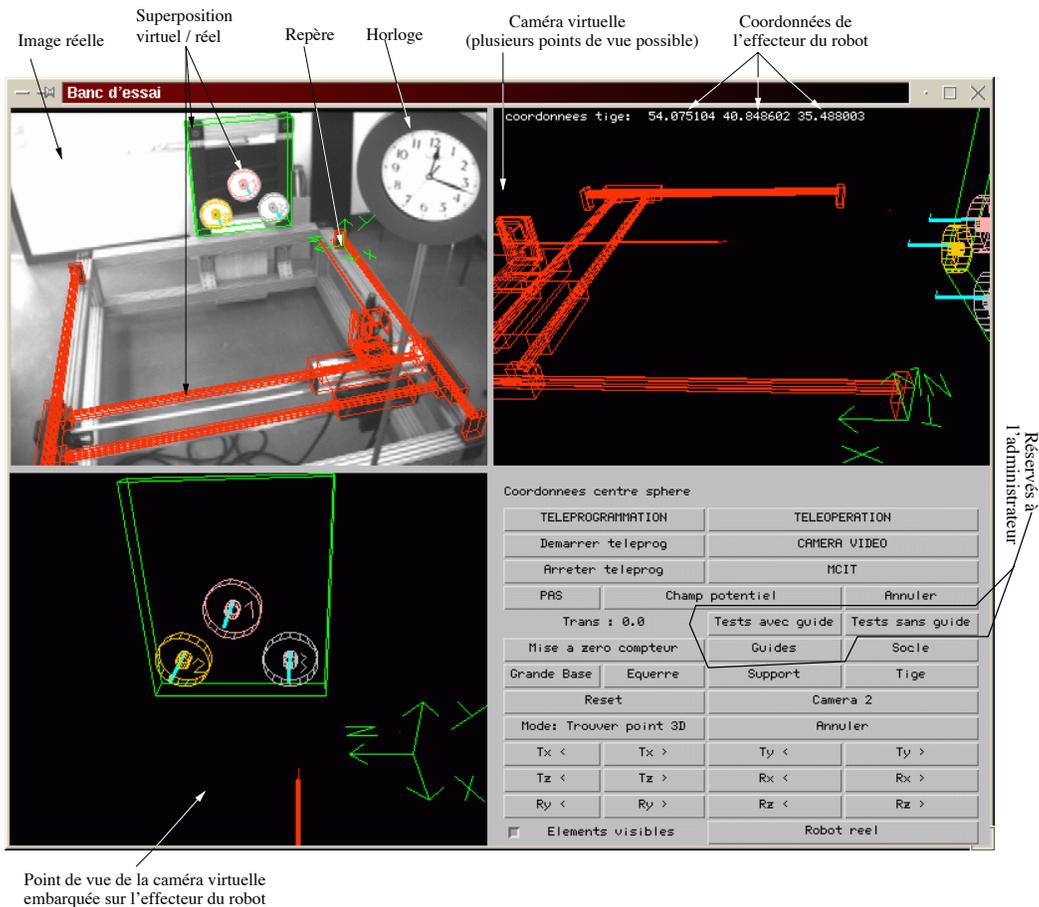


FIG. 4.5 – l'IHM destinée au client administrateur, avec en haut à gauche, une image vidéo de résolution 288×384 , en bas à droite la partie réservée à l'administrateur pour créer des guides virtuels.

choisir le pas de déplacement du robot, etc.

La figure 4.7 montre comment choisir le mode de télétravail à partir du tableau de bord de l'interface de ARITI.

4.1.2.2 IHM utilisateur

Cette deuxième version de l'interface, est destinée à tous les autres opérateurs (clients) appelés "utilisateurs" du système ARITI. Elle est accessible sur le site web de notre laboratoire au <http://lsc.cemif.univ-evry.fr:8080/> ou encore sur le site de la NASA "NASA Space Telerobotics Program" au http://ranier.oact.hq.nasa.gov/telerobotics_page/realrobots.html).

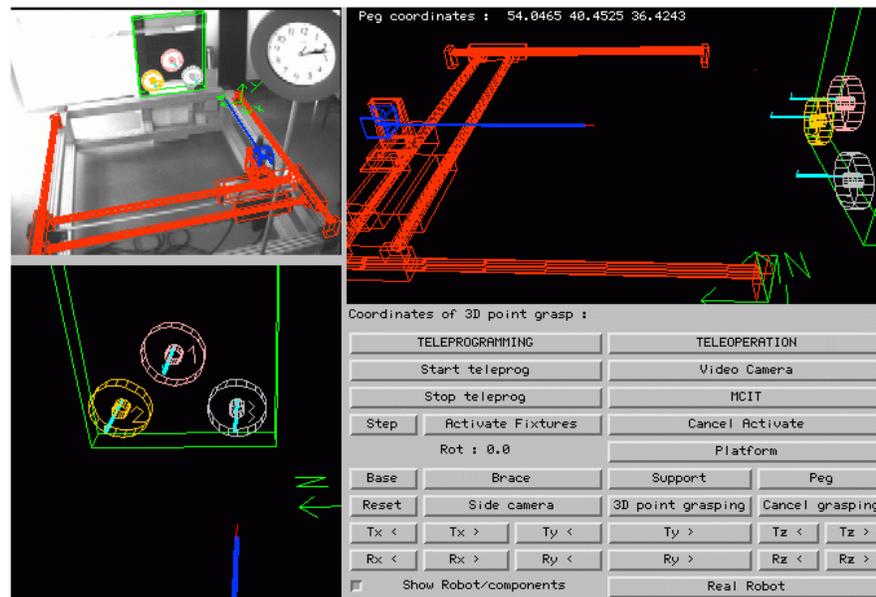


FIG. 4.6 – l'IHM destinée au client utilisateur, avec en haut à gauche, une image vidéo de résolution 192×256 , et l'utilisateur ne peut pas créer de guides virtuels.

Les différences principales entre cette version et la précédente, sont situées au niveau de la taille de l'image vidéo et de la possibilité de créer des guides virtuels. Dans la version administrateur (figure 4.5), l'image vidéo est de 288×384 (110592 octets = 108 KO), alors que dans la version utilisateur (figure 4.6) la taille est de $2/3$ ($192 \times 256=49152$ octets = 48 KO).

4.1.2.3 Description de l'outil de création et de manipulation des guides virtuels

Comme nous l'avons décrit dans le second chapitre, l'opérateur peut être assisté par des guides virtuels lors de la téléopération. Il peut utiliser des guides virtuels existants dans la base de données des guides (figure 4.7-(B)). La description de ces guides sera donnée avec la description des tâches de téléopération dans la section suivante.

Dans le cas où l'opérateur désire créer de nouveaux guides virtuels correspondant à une nouvelle tâche, un outil de création et de manipulation de guides virtuels apparaît en

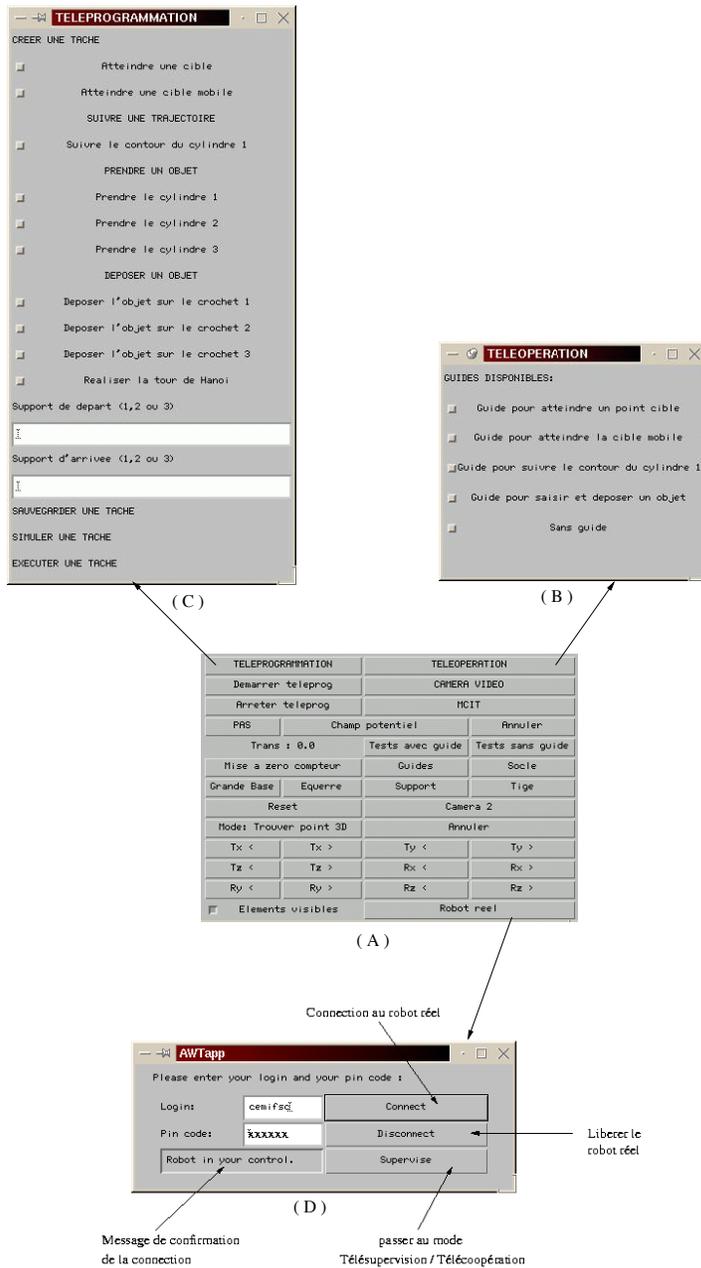


FIG. 4.7 – (A): Tableau de bord de l'interface de ARITI. (B): mode téléopération, (C): mode téléprogrammation de tâche, (D): contrôle du robot réel et le mode télésupervision ou télécoopération

activant le bouton “Guides” (voir la figure 4.7-(A)). Il s’agit d’un modèleur 3D (figure 4.8) permettant à l’opérateur de générer des objets virtuels en trois dimensions. Les méthodes utilisées sont celles décrites dans le second chapitre (surfaces, volumes ouverts, volumes fermés, etc.) et représentées par le générateur de guides virtuels dans la figure 4.9.

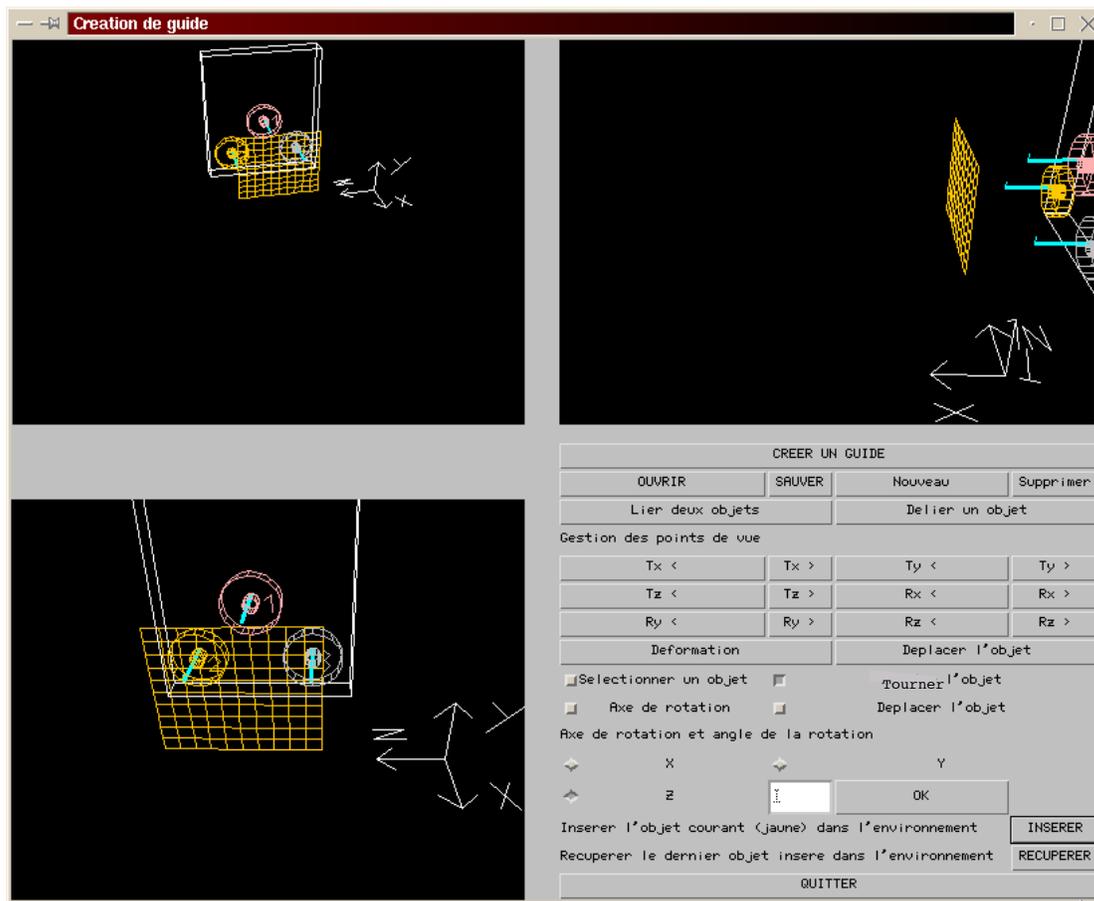


FIG. 4.8 – L’interface du modèleur 3D pour la création et la manipulation de guides virtuels.

A chaque guide virtuel est associé des propriétés permettant à l’opérateur et au système de l’identifier afin que ce guide puisse accomplir le rôle qui lui a été réservé. Les figures 4.11 et 4.10 montrent respectivement, un exemple d’un guide virtuel simple et de ses propriétés pour la tâche d’atteinte d’une cible. Un autre exemple de propriétés d’un guide virtuel (figure 4.14) pour la tâche de suivi de contour est donnée dans la figure B.41.

Nous verrons plus loin les propriétés des guides virtuels composés.

4.1.3 Mode Téléopération

Dans ce mode de télétravail, l’opérateur commande le robot réel via le robot virtuel. Il peut téléopérer le robot réel avec ou sans assistance des guides virtuels. Nous allons décrire quelques tâches de téléopération réalisées avec assistance des guides virtuels (Otmame *et al.*, 2000a) (Otmame *et al.*, 2000c). Ceci permettra de décrire les différents guides présentés dans la figure 4.7-(B).

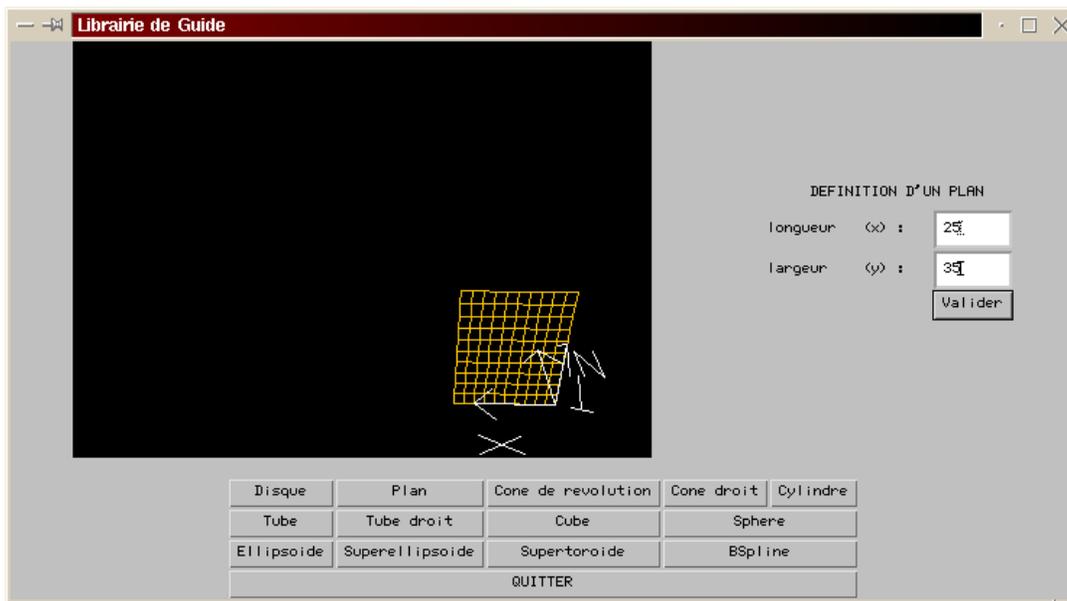


FIG. 4.9 – Générateur de guides virtuels: Exemple d'un plan généré à partir de deux paramètres, la longueur et la largeur.

Les tâches que nous allons décrire sont les suivantes :

- Atteinte de cible.
- Atteinte de cible mobile.
- Suivi de contour d'un objet.
- Saisie et dépôt d'un objet.

Pour chacune d'elles, nous décrivons le(les) guide(s) associé(s).

4.1.3.1 Tâche d'atteinte de cible

La cible en question est généralement un point 3D dans l'espace. Ce point peut être attaché à un objet de l'environnement à saisir par exemple. Le guide virtuel utilisé est un cône tronqué ouvert (entonnoir).

L'extrémité de ce guide est attachée à la périphérie d'un objet cylindre, dans ce cas le point 3D cible est le centre de cette extrémité représenté par le point P_1 dans la figure 4.10.

Le point " E " de coordonnées (X_e, Y_e, Z_e) désigne l'extrémité de l'effecteur du robot. Le point " A " de coordonnées (X_a, Y_a, Z_a) représente le centre du disque de rayon " r_1 " passant par le point " E ".

Le point " P_1 " de coordonnées (X_1, Y_1, Z_1) est le centre de la première extrémité du guide virtuel et de rayon " r ".

Le point " P_2 " de coordonnées (X_2, Y_2, Z_2) est le centre de la deuxième extrémité du guide et de rayon " R ".

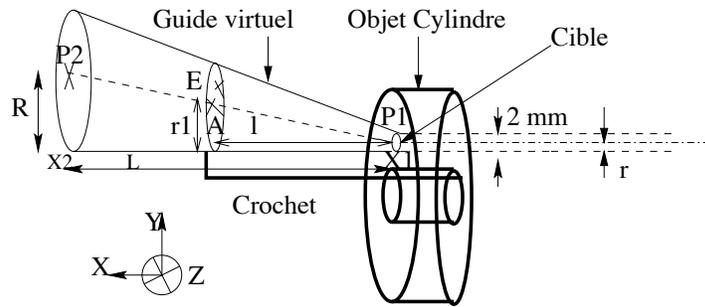


FIG. 4.10 – Guide virtuel utilisé pour atteindre une cible: ici la cible est un point 3D se trouvant sur la périphérie de l'objet cylindre

Type du guide	Nom du guide	Atteindre cible3.obj
	Composition	Simple Compose
	Guide de référence	
	Guide précédent	
	Guide suivant	
	Etat du guide	Actif Passif
Référentiel	Influence du guide	Attractif Repulsif
	Action du guide	Guide semi autonome
	Tx	-10.46325
	Ty	25.519833
	Tz	27.967573
	Rx	90.0
Première ouverture horizontale/verticale	Ry	0.0
	Rz	0.0
	Attachement	Objet_cylindre_3.obj
	Zone d'influence	Tube
	Deuxième ouverture horizontale/verticale	Paramètre 1:
Paramètre 2:		0.1
Hauteur	Paramètre 3:	20.0
	Paramètre 4:	20.0
	Paramètre 5:	58.0

Le guide est attaché à l'objet cylindre numéro 3

FIG. 4.11 – Propriétés du guide virtuel d'assistance à la téléopération pour atteindre une cible. La cible dans ce cas est l'objet cylindre numéro 3 (figure 4.1)

Nous décrivons ici les propriétés données par la figure 4.11 et le formalisme de ce guide :

- **Nom du guide :** L’identification de ce guide se fait par son nom (“*Atteindre_cible3.obj*” dans figure 4.11). Ceci permet à l’opérateur de retrouver ce guide, de changer ses propriétés pour saisir un autre objet par exemple. En effet ce guide peut être utilisé pour toutes les tâches de saisie.
- **Type du guide :** Ce guide virtuel est dit “simple” et “actif en attraction” (comme indiqué dans la figure 4.11). Il s’agit d’un guide d’assistance “semi-autonome”. Ce type est utilisé pour définir la fonction du guide.
- **Référentiel :** Donne la position et l’orientation de ce guide par rapport au repère de l’environnement virtuel.
- **Attachement :** L’attachement de ce guide à un objet virtuel, se fait en précisant le nom de cet objet dans le champ “Attachement” représenté dans la figure 4.11. Ici, ce champ contient “Objet_cylindre_3.obj” qui représente bien l’objet cylindre de numéro 3 appartenant à l’environnement virtuel.
- **Zone d’influence :** Constitue un volume de travail en dehors duquel l’extrémité de l’effecteur du robot “*E*” ne peut pas sortir (sauf si l’opérateur recule suffisamment pour faire disparaître le guide). Le nom de l’équation du guide (“*tube*”) et ses paramètres sont mémorisés afin de les utiliser pour définir la condition d’activation de ce guide (pré-condition).
- **Pré-condition :** Le guide virtuel est activé dès que l’effecteur du robot se rapproche suffisamment du support métallique (atteinte de la zone d’influence). Ceci se traduit par l’appartenance de l’extrémité de l’effecteur du robot “*E*” au volume de ce guide en vérifiant les contraintes suivantes :

$$\begin{cases} X_1 < X_e \leq X_2 \\ (Y_e - Y_a)^2 + (Z_e - Z_a)^2 \leq r_1^2 \end{cases} \quad (4.1)$$

avec $r_1 = l \times (R - r) / L + r$ et “*L*” représente la hauteur du guide virtuel, “*l*” la hauteur du sous cône dont l’extrémité est le disque contenant le point “*A*” (voir figure 4.10)

- **Fonction :** Elle est représentée par une combinaison des actions réalisées par l’opérateur (dans ce cas, “**Déplacer le robot**”) et d’une fonction de projection de l’extrémité de l’effecteur du robot “*E*” sur le point “*A*”. Cette projection peut s’écrire sous la forme suivante :

$$\begin{cases} X_e = X_a \\ Y_e = Y_a \\ Z_e = Z_a \end{cases} \quad (4.2)$$

avec

$$\begin{cases} X_a = X_e \\ Y_a = Y_e + r_1 - r \\ Z_a = Z_e \end{cases} \quad (4.3)$$

- **Post-condition :** Elle constitue la négation de la pré-condition. En effet, le guide est désactivé si par exemple l’opérateur fait reculer le robot suffisamment pour sortir du guide ou si la cible est atteinte. Ces deux conditions peuvent s’écrire sous la forme suivante :

$$\begin{cases} X_e \leq X_1 \\ \text{ou} \\ X_e > X_2 \end{cases} \quad (4.4)$$

4.1.3.2 Tâche de suivi de cible mobile

Cette tâche permet au robot de suivre une cible virtuelle, que l'opérateur peut déplacer dans son environnement, ou encore suivre un objet virtuel en mouvement lorsque la cible mobile est attachée à cet objet. La figure 4.12 illustre le principe de suivi d'objet en mouvement.

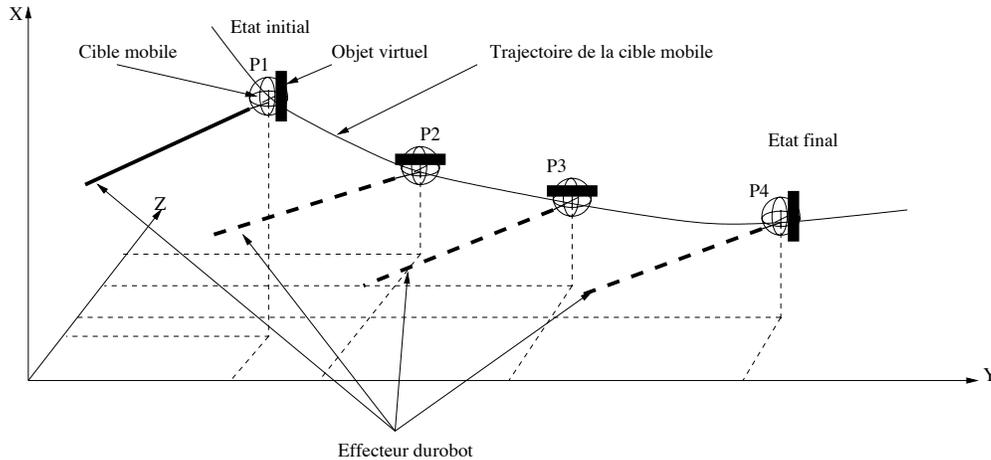


FIG. 4.12 – Illustration du principe de suivi de cible mobile par l'effecteur du robot.

En effet le déplacement de l'objet virtuel entraîne avec lui la cible qui lui est attachée. Cet objet se déplace suivant une trajectoire que réalise l'objet en passant par les points P_1 , P_2 , P_3 et P_4 par exemple. Dans ce cas les coordonnées de la cible mobile sont déterminées en fonction des coordonnées de l'objet virtuel en mouvement.

La réalisation de la tâche de suivi de cible se fait en deux étapes :

La première étape consiste à désigner sur écran la cible mobile, et à activer un guide virtuel pour atteindre cette cible. L'opérateur peut utiliser le même type de guide virtuel que pour la tâche précédente, sauf que ce guide (sous forme d'une sphère) sera attaché à un objet virtuel en mouvement. La méthode de désignation d'un point 3D à partir de deux points 2D est décrite dans cf. § 3.3, voir aussi la figure 3.17.

La deuxième étape consiste à déplacer la cible mobile dans son environnement en translation et/ou en rotation par rapport aux axes X , Y et Z . La figure 4.13, montre les captures d'écran de la deuxième fenêtre de l'interface de ARITI pendant le suivi de la cible par le robot. Ici l'opérateur ne contrôle pas directement le robot, mais ce dernier subi la fonction d'attraction du guide virtuel attaché à la cible mobile.

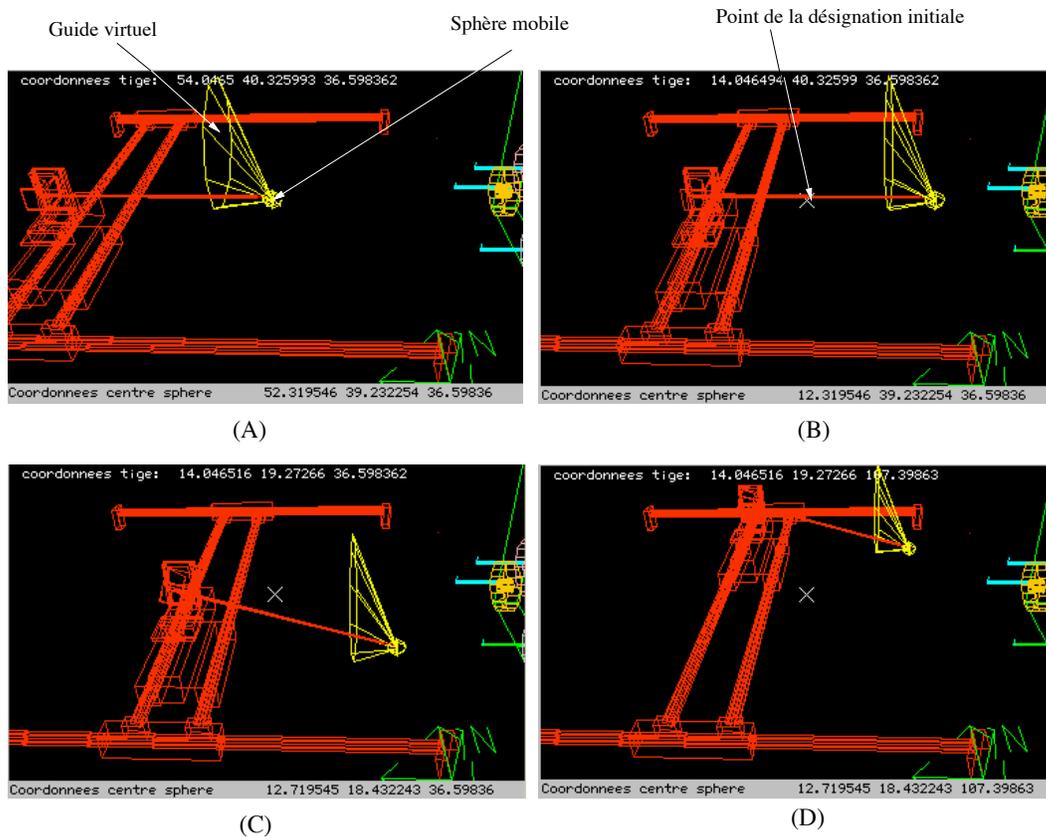


FIG. 4.13 – Exemple de suivi de cible mobile par le robot. La cible est atteinte par le robot (A), a effectué une translation par rapport à l'axe X (B), ensuite une rotation par rapport à l'axe Z (C) et enfin une translation par rapport à l'axe Z (D)

4.1.3.3 Tâche de suivi de contour d'un objet

Dans cette tâche, l'opérateur téléopère le robot pour atteindre le contour d'un objet cylindre et réalise ensuite le suivi de son contour.

Le guide virtuel utilisé pour assister l'opérateur à réaliser cette tâche est un cône de révolution. Une extrémité de ce guide est attachée au contour d'un objet cylindre et l'autre est destinée à recevoir l'extrémité de l'effecteur du robot (figure 4.14).

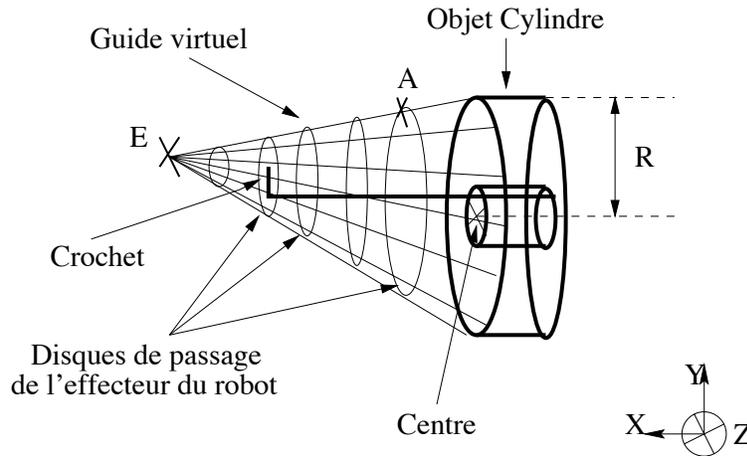


FIG. 4.14 – Le guide virtuel utilisé pour le suivi de contour. Le point “E” désigne l'extrémité de l'effecteur du robot et le point “A” appartient au contour du disque de passage de “E”.

En effet, avec ce type de guide (semi-autonome), l'opérateur va pouvoir déplacer l'effecteur du robot directement sur le contour de l'objet cylindre et ensuite il lui suffit de déplacer l'effecteur du robot de gauche à droite (ou inversement) pour parcourir le contour du cylindre.

Les propriétés de ce guide sont données par la figure B.41, et ci-dessous nous présentons essentiellement la pré-condition, la fonction et la post-condition de ce guide.

- **Pré-condition :** Le guide virtuel est activé dès que l'effecteur du robot se rapproche suffisamment du support métallique (atteinte de la zone d'influence *cône*). Ceci se traduit par l'appartenance de l'extrémité de l'effecteur du robot “E” au volume de ce guide en vérifiant les contraintes suivantes :

$$\begin{cases} X_e \geq X_{centre} \\ (Y_e - Y_{centre})^2 + (Z_e - Z_{centre})^2 \leq R^2 \end{cases} \quad (4.5)$$

- **Fonction :** Elle est représentée par une combinaison des actions réalisées par l'opérateur (dans ce cas, “Déplacer le robot”) et par une fonction de projection de l'extrémité de l'effecteur du robot “E” sur le point “A” qui appartient au contour du disque contenant “E”. Cette projection peut s'écrire sous la forme suivante :

$$\begin{cases} X_e = X_a \\ Y_e = Y_a \\ Z_e = Z_a \end{cases} \quad (4.6)$$

Nom du guide	Suivre contour1.obj
Composition	Simple Compose
Guide de reference	
Guide precedent	
Guide suivant	
Etat du guide	Actif Passif
Influence du guide	<input checked="" type="checkbox"/> Attractif <input type="checkbox"/> Repulsif
Action du guide	Guide semi autonome
Tx	28.182426
Ty	34.221222
Tz	57.128757
Rx	0.0
Ry	0.5
Rz	0.0
Attachement	Objet_cylindre_1.obj
Zone d'influence	Cone
Parametre 1:	10.0
Parametre 2:	10.0
Parametre 3:	10.0

FIG. 4.15 – Propriétés du guide virtuel utilisé pour le suivi de contour. Dans ce cas ce guide permet de suivre le contour du cylindre numéro 1

- **Post-condition :** Dans ce cas, le guide est désactivé seulement si l’opérateur recule le robot suffisamment pour sortir du guide.

4.1.3.4 Tâche de saisie et dépôt d’un objet

Cette tâche consiste à décrocher un objet cylindre depuis son support métallique (tâche de saisie) et l’accrocher ensuite sur un autre support (tâche de dépôt).

L’opérateur peut être assisté par des guides virtuels pour réaliser cette tâche. Nous décrivons ici, les guides virtuels utilisés pour chacune des tâches.

4.1.3.4.a Saisie d’un objet

Généralement, la tâche “saisie d’un objet” est toujours précédée de la tâche “atteinte d’un objet”. En effet, pour saisir un objet, il faut dans un premier temps atteindre cet objet par l’effecteur du robot. Pour cela, le guide virtuel donné dans la figure 4.10 est utilisé. Dès que cet objet est atteint, ce guide disparaît pour faire apparaître le guide de saisie proprement dit.

La figure 4.16 décrit le guide virtuel utilisé pour saisir un objet cylindre. Ce guide est composé de trois guides simples sous forme de cylindres appelés “Guide 1”, “Guide 2” et “Guide 3”. Ces trois guides sont disposés de telle sorte que la tâche de saisie d’objet ne

provoque pas la détérioration du robot ou de l'objet lui même.

Le "Guide 1", est positionné à l'intérieur de l'objet à saisir (formant ainsi une zone de prise de l'objet). En effet, pour prendre l'objet, l'effecteur du robot doit piquer l'objet en faisant une translation en profondeur de 2 cm.

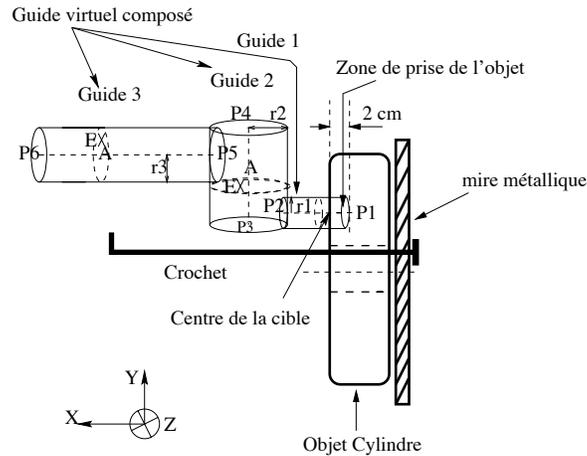


FIG. 4.16 – Le guide virtuel utilisé pour la saisie d'un objet cylindre. Le point "E" désigne l'extrémité de l'effecteur du robot et le point "A" désigne le centre du disque passant par le point "E".

Les propriétés de ce guide sont données par la figure 4.17, et nous présentons ensuite la pré-condition, la fonction et la post-condition de ces guides.

Dans ce qui suit, le point P_i (figure 4.17) a pour coordonnées (X_i, Y_i, Z_i) pour $i = 1..6$

Nom du guide	Saisie guide3.obj	Nom du guide	Saisie guide2.obj	Nom du guide	Saisie guide1.obj
Composition	Simple	Simple	Simple	Simple	Simple
Guide de reference					
Guide precedent					
Guide suivant					
Etat du guide	Actif	Actif	Actif	Actif	Actif
Influence du guide	Attractif	Attractif	Attractif	Attractif	Attractif
Action du guide	Guide semi autonome	Guide semi autonome	Guide semi autonome	Guide semi autonome	Guide semi autonome
Tx	-6.494344	28.295198	-31.509954		
Ty	24.045022	33.120077	21.003782		
Tz	38.238252	38.595398	39.063774		
Rx	0	0	0		
Ry	90	90	90		
Rz	0	0	0		
Attachement	Saisie_guide2.obj	Saisie_guide1.obj	Objet_cylindre_3.obj		
Zone d'influence	Cylindre	Cylindre	Cylindre		
Parametre 1:	0	1.5	0.25		
Parametre 2:	0	1.5	0.25		
Parametre 3:	0	1.5	0.25		
Parametre 4:	0	1.5	0.25		
Parametre 5:	42	3.75	5		

FIG. 4.17 – Les propriétés des guides virtuels utilisés pour la saisie de l'objet cylindre numéro 3.

- **Pré-condition :** Les guides virtuels sont activés dès que le premier guide “Guide 1” est activé. En effet, ce guide est attaché à l’objet à saisir (figure 4.17-(C)), alors que le “Guide 2” est attaché au “Guide 1” (figure 4.17-(B)) et le “Guide 3” est attaché au “Guide 2” (figure 4.17-(A)). Ceci forme une chaîne de guides virtuels permettant ainsi l’activation de toute cette chaîne dès que le premier guide est activé.

Au moment de l’apparition de ces guides, l’effecteur du robot se trouve déjà dans la zone d’influence du premier guide (le volume du cylindre) et sur la cible atteinte précédemment. Les contraintes qui déterminent la pré-condition sont données par l’appartenance de l’extrémité de l’effecteur du robot “E” aux volumes de chaque guide virtuel :

- ❶ Dans “Guide 1” :

$$\begin{cases} X_1 \leq X_e \leq X_2 \\ (Y_e - Y_a)^2 + (Z_e - Z_a)^2 \leq r_1^2 \end{cases} \quad (4.7)$$

avec $A(X_a, Y_a, Z_a)$ le centre du disque de rayon r_1 contenant “E” et

$$\begin{cases} X_a = X_e \\ Y_a = Y_1 \\ Z_a = Z_1 \end{cases} \quad (4.8)$$

- ❷ Dans “Guide 2” :

$$\begin{cases} Y_3 \leq Y_e \leq Y_4 \\ (X_e - X_a)^2 + (Z_e - Z_a)^2 \leq r_2^2 \end{cases} \quad (4.9)$$

avec $A(X_a, Y_a, Z_a)$ le centre du disque de rayon r_2 contenant “E” et

$$\begin{cases} X_a = X_3 \\ Y_a = Y_e \\ Z_a = Z_3 \end{cases} \quad (4.10)$$

- ❸ Dans “Guide 3” :

$$\begin{cases} X_5 \leq X_e \leq X_6 \\ (Y_e - Y_a)^2 + (Z_e - Z_a)^2 \leq r_3^2 \end{cases} \quad (4.11)$$

avec $A(X_a, Y_a, Z_a)$ le centre du disque de rayon r_3 contenant “E” et

$$\begin{cases} X_a = X_e \\ Y_a = Y_5 \\ Z_a = Z_5 \end{cases} \quad (4.12)$$

- **Fonction :** Chaque guide virtuel étant semi-autonome, la fonction de chaque guide représente alors une combinaison des actions réalisées par l’opérateur “**Déplacer le robot**”) et de la fonction de projection de l’extrémité de l’effecteur du robot “E” sur le point “A”. Cette projection est représentée pour chaque guide comme suit :

$$\begin{cases} X_e = X_a \\ Y_e = Y_a \\ Z_e = Z_a \end{cases} \quad (4.13)$$

- **Post-condition :** La chaîne de guides virtuels est désactivée, lorsque l’extrémité de l’effecteur du robot se trouve à l’extérieur du troisième guide. En effet, ceci signifie que l’objet vient d’être saisi (décroché) de son support. Cette post-condition peut se résumer à :

$$X_e > X_6 \quad (4.14)$$

4.1.3.4.b Dépôt d’un objet

Pour déposer un objet saisi, l’opérateur rapproche l’effecteur du robot vers un des supports métalliques sur lequel doit être accroché cet objet.

Au moment où l’effecteur du robot se rapproche de l’un des supports, une chaîne de guides virtuels apparaît pour l’aider à accomplir correctement cette tâche. Il s’agit d’un guide virtuel composé de quatre guides simples, un cône représenté par “Guide 4”, suivi de trois cylindres imbriqués “Guide 3”, “Guide 2” et “Guide 1” (figure 4.18-(A)).

Ces quatre guides sont disposés de telle sorte que l’effecteur du robot et l’objet saisi puissent se déplacer sans provoquer des collisions avec le support métallique.

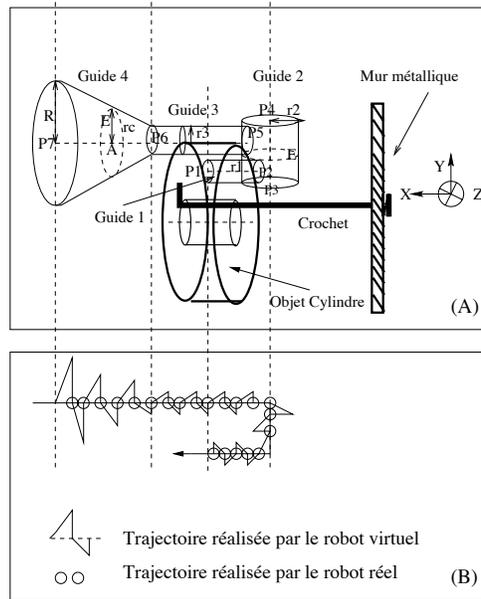


FIG. 4.18 – Le guide virtuel utilisé pour le dépôt d’un objet cylindre. (A) : Positionnement des différents guides, avec le point “E” qui désigne l’extrémité de l’effecteur du robot et le point “A” le centre du disque passant par le point “E”; (B) : Exemple d’influence des guides sur les trajectoires réalisées par les robots virtuel et réel.

Les propriétés de ces guides sont données par la figure 4.19, et nous présentons ensuite la pré-condition, la fonction et la post-condition de ces guides.

Dans ce qui suit, le point P_i (figure 4.18-(A)) a pour coordonnées (X_i, Y_i, Z_i) pour $i = 1..7$.

- **Pré-condition :** La chaîne de guides virtuels est activée dès que le quatrième guide “Guide 4” est activé. En effet, ce guide est attaché à un autre guide “Guide 3” (figure 4.19-(D)), le “Guide 3” est attaché au “Guide 2” (figure 4.19-(C)) et le “Guide

2” est attaché au “Guide 1” (figure 4.19-(B)). L’assistance à l’opérateur prend son effet à la fin du parcours du guide “Guide 1”, c’est pour cette raison que le champ “Attachement” des propriétés de ce guide contient “Guide libre” (figure 4.19-(A)).

De la même façon que les précédentes tâches, les contraintes qui déterminent la pré-condition sont données par l’appartenance de l’extrémité de l’effecteur du robot “E” au volume de chaque guide virtuel par exemple :

Dans “Guide 4” :

$$\begin{cases} X_6 \leq X_e \leq X_7 \\ (Y_e - Y_a)^2 + (Z_e - Z_a)^2 \leq r_c^2 \end{cases} \quad (4.15)$$

avec $A(X_a, Y_a, Z_a)$ le centre du disque de rayon r_c contenant “E” et

Pour les guides 3, 2 et 1, les pré-conditions sont déterminées avec le même principe.

- **Fonction :** De la même façon que la tâche de saisie, la fonction de chaque guide représente une combinaison des actions réalisées par l’opérateur (“Déplacer le robot”) et de la fonction de projection de l’extrémité de l’effecteur du robot “E” sur le point “A”. La figure 4.18-(B), illustre un exemple d’influence des guides virtuels sur les déplacements des robots virtuel et réel. En effet, les actions de l’opérateur sur le robot virtuel sont corrigés par le guide avant l’exécution par le robot réel. Cette correction est représentée pour chaque guide par cette fonction de projection :

$$\begin{cases} X_e = X_a \\ Y_e = Y_a \\ Z_e = Z_a \end{cases} \quad (4.16)$$

- **Post-condition :** La chaîne de guides virtuels est désactivée, lorsque l’extrémité de l’effecteur du robot se trouve, soit à l’extérieur du “Guide 4” (généralement en reculant, avant le dépôt de l’objet), soit à l’extérieur du guide libre “Guide 1”. En effet, ce dernier cas signifie que l’objet vient d’être déposé (accroché) sur un support. Cette post-condition peut se résumer à :

$$\begin{cases} X_e > X_7 \\ \quad \quad \quad \text{ou} \\ X_e > X_1 \quad \text{et} \quad Y_e \leq Y_1 \end{cases} \quad (4.17)$$

4.1.4 Mode Téléprogrammation de tâches

Les tâches présentées dans la section précédente (mode téléopération), sont réalisées par l’opérateur en téléopérant les robot virtuel et réel. Il est aussi possible d’exécuter ces tâches d’une manière automatique par les robots virtuel et réel.

En effet, dans le mode de téléprogrammation de tâches (désignation d’objectif), l’opérateur désigne les tâches qu’il désire faire réaliser par les robots virtuel et réel en choisissant l’une des tâches dans le menu “TELEPROGRAMMATION” (figure 4.7-(C)).

Après avoir choisi une tâche, l’opérateur lance l’exécution en activant le bouton “Démarrer téléprog” (figure 4.7-(A)). Le rôle de l’opérateur consiste donc à superviser l’exécution de cette tâche par les deux robots virtuel et réel. Il peut aussi intervenir pour arrêter cette exécution s’il le désire, en activant le bouton “Arrêter téléprog” dans le tableau de bord de l’interface du système ARITI (figure 4.7-(A)).

(A)		(B)		(C)		(D)	
Nom du guide	Depot guide1.obj	Nom du guide	Depot guide2.obj	Nom du guide	Depot guide3.obj	Nom du guide	Depot guide4.obj
Composition	Simple Compose						
Guide de référence		Guide de référence		Guide de référence		Guide de référence	
Guide précédent		Guide précédent		Guide précédent		Guide précédent	
Guide suivant		Guide suivant		Guide suivant		Guide suivant	
Etat du guide	Actif Passif						
Influence du guide	Attractif Repulsif						
Action du guide	Guide semi autonome						
Tx	-19.886972	Tx	-22.5759	Tx	-16.924686	Tx	-0.08054916
Ty	31.443226	Ty	33.268994	Ty	34.015682	Ty	32.873344
Tz	36.321148	Tz	37.11286	Tz	36.033344	Tz	35.88948
Rx	0	Rx	0	Rx	0	Rx	0.5
Ry	0.5	Ry	0.5	Ry	0.5	Ry	1.5
Rz	0	Rz	0	Rz	0	Rz	0
Attachement	Guide libre	Attachement	Depot_guide1.obj	Attachement	Depot_guide2.obj	Attachement	Depot_guide3.obj
Zone d'influence	Cylindre	Zone d'influence	Tube	Zone d'influence	Tube	Zone d'influence	Cylindre
Parametre 1:	0.5						
Parametre 2:	0.5						
Parametre 3:	0.5	Parametre 3:	0.5	Parametre 3:	0.5	Parametre 3:	0
Parametre 4:	0.5	Parametre 4:	0.5	Parametre 4:	0.5	Parametre 4:	0
Parametre 5:	0.5	Parametre 5:	0.5	Parametre 5:	0	Parametre 5:	0.5

FIG. 4.19 – Les propriétés des guides virtuels utilisés pour le dépôt de l’objet cylindre numéro 3 sur le support numéro 1.

Les schémas de fonctionnement des différentes tâches de téléprogrammation sont données dans l’annexe B

4.1.5 Mode Télésupervision et Télécoopération

Ce mode de télétravail, permet à plusieurs opérateurs connectés au système ARITI de participer à la réalisation d’une tâche. Ce principe de télétravail est décrit dans le chapitre précédent (3.3).

Il existe plusieurs possibilités permettant aux opérateurs de participer à une mission. Dans cette section, nous considérons que la mission fait participer une équipe de quatre opérateurs.

Considérons que la mission est constituée par les tâches suivantes :

- T_1 : Atteindre l’objet numéro 3 se trouvant sur le support numéro 3.
- T_2 : Saisir l’objet numéro 3 depuis le support numéro 3.
- T_3 : Déposer l’objet saisi numéro 3 sur le support numéro 1.
- T_4 : Remettre l’objet déposé sur son support d’origine (Atteindre l’objet numéro 3 sur le support numéro 1, le saisir et le déposer sur le support numéro 3). Cette dernière tâche consiste à annuler les tâches T_1, T_2 et T_3 .

Considérons l’organisation de l’équipe comme suit :

L’opérateur OP_i réalise la tâche T_i pour $i = 1..4$. Il est appelé “**Maître**” s’il a le contrôle du robot réel, sinon il est appelé “**Superviseur**”. Le tableau suivant illustre l’exécution de cette mission avec ce mode de télétravail.

Opérateur/Tâche	T_1	T_2	T_3	T_4
Op_1	Maître	Superviseur	Superviseur	Superviseur
Op_2	Superviseur	Maître	Superviseur	Superviseur
Op_3	Superviseur	Superviseur	Maître	Superviseur
Op_4	Superviseur	Superviseur	Superviseur	Maître

TAB. 4.1 – Exemple d'organisation d'une équipe pour la réalisation d'une mission de téléopération en coopération

4.2 Télétravail avec un robot mobile

Il s'agit d'une application en cours de réalisation qui consiste à téléopérer un robot mobile en utilisant le système ARITI. En effet il est possible d'ajouter au système existant quelques modules (modèle virtuel du robot mobile, module client/serveur commande) pour pouvoir contrôler un robot mobile, ou encore contrôler les deux robots simultanément. La figure 4.20 montre une vue générale de la mise à jour du système ARITI, avec en pointillé les modules ajoutés.

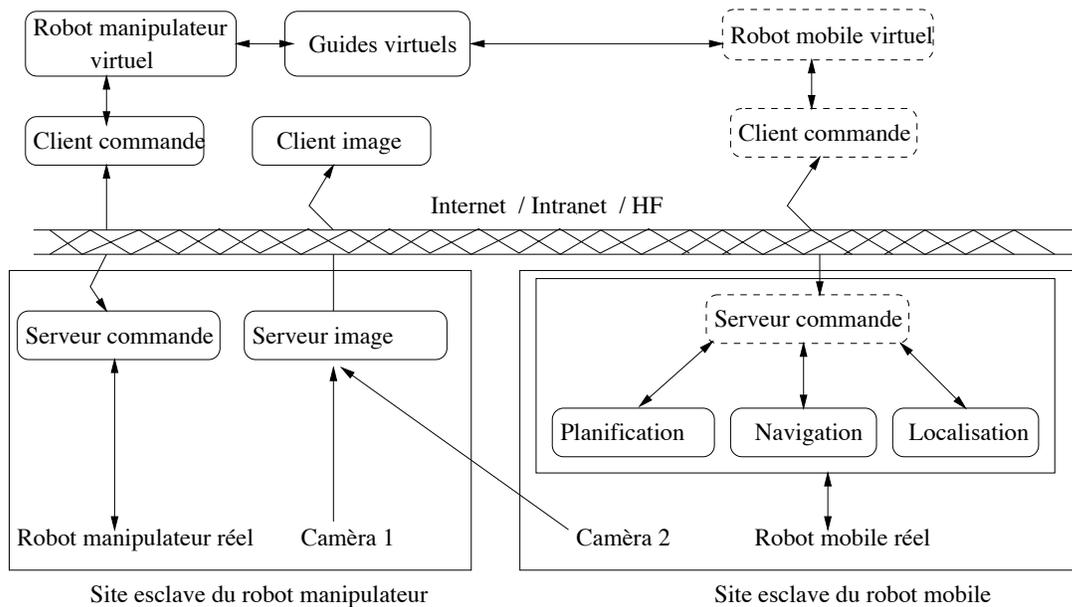


FIG. 4.20 – Mise à jour de l'architecture du système ARITI pour le contrôle d'un robot mobile

Cette application nous permet d'un côté, d'étudier le comportement du système ARITI vis à vis d'un robot mobile téléopérable et pouvant être autonome (dans ce cas le contrôle devient semi-autonome). De l'autre coté, elle permet d'étudier comment une IHM basée sur des techniques de la réalité virtuelle/augmentée peut contribuer à l'amélioration des performances des personnes handicapées (Otmane *et al.*, 2000d).

En effet, le système devient semi-autonome, du moment que le robot mobile possède déjà une certaine autonomie, telle que l'évitement d'obstacle (capteurs ultrason), planification, navigation et localisation. En effet, la planification permet au robot de déterminer une trajectoire à suivre pour atteindre un but (un point dans l'environnement) désigné par l'opérateur. La navigation, permet au robot mobile d'atteindre le but avec évitement

d'obstacles. La localisation quant à elle, permet de déterminer la position et l'orientation du robot dans son environnement grâce aux capteurs odométriques.

Dans cette section, nous décrivons deux robots mobiles avec lesquelles nous travaillons ainsi que l'interface homme machine permettant le contrôle du robot mobile utilisé avec ARITI. Nous présentons ensuite quelques guides virtuels utilisés afin d'assister l'opérateur à téléopérer le robot mobile pour traverser une porte.

4.2.1 Description des robots mobiles

Le premier robot est une base mobile qui fait 30 cm de diamètre (figure 4.21-(A)) et le deuxième est constitué d'une base mobile (qui a la largeur d'un fauteuil roulant) sur laquelle est embarqué un bras manipulateur de type MANUS (figure 4.21-(B)). Les deux robots possèdent une ceinture de huit capteurs ultrason et une caméra qui peut réaliser deux rotations.

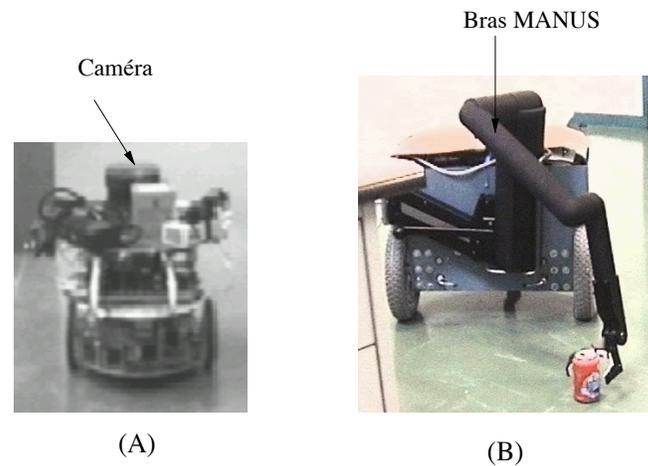


FIG. 4.21 – Les images des deux robots mobiles utilisés. (A) : le robot utilisé avec le système ARITI, (B) : le robot destiné au projet d'assistance aux personnes handicapées

4.2.2 Description de l'IHM

De la même façon que l'interface donnée dans la section précédente (4.1 dans la figure 4.5), l'interface pour contrôler le robot mobile (figure 4.21-(A)) est constituée de quatre parties.

Une partie pour la réception d'images vidéos (pas de superposition du modèle virtuel sur l'image réelle), deux parties pour représenter le robot mobile et son environnement dans un mode virtuel et la quatrième partie constitue le tableau de bord de l'interface. Ce dernier, permet à l'opérateur de :

- Choisir l'un des deux modes de contrôle, manuel (téléopération) ou navigation (semi-autonome).
- Changer de points de vue de la caméra virtuelle.
- Créer et utiliser des guides virtuels.

La figure 4.22 montre L'IHM utilisée pour le contrôle et la supervision du robot mobile. La fenêtre en haut à gauche, réceptionne des images vidéo envoyées par une caméra qui visualise le robot réel et son environnement (cette caméra peut être remplacée par celle embarquée sur le robot mobile). La fenêtre en haut à droite (figure 4.22), représente un point de vue virtuel de la scène obtenue par une caméra virtuelle placée en hauteur (cette caméra peut être déplacée ou tournée suivant les trois axes X, Y, Z). La fenêtre en bas à gauche, représente un point de vue virtuel donné par une caméra embarquée sur le robot mobile (nous permet de voir ce que voit le robot mobile virtuel).



FIG. 4.22 – L'IHM utilisée pour le contrôle et la supervision du robot mobile.

4.2.3 Guides virtuels pour traverser une porte

Dans cette section, nous présentons deux types de guides virtuels pouvant être utilisés pour traverser une porte par le robot.

Le premier type est constitué de deux plans parallèles et répulsifs, placés de telle sorte que le robot mobile ne puisse pas toucher les deux murs qui forment un passage représentant une porte. Le deuxième type de guide est dit attractif, représenté sous forme d'une trajectoire générée par une B-Spline.

4.2.3.1 tâche avec deux plans répulsifs

Dans cette tâche, l'opérateur téléopère le robot réel via le robot virtuel. Il déplace le robot virtuel jusqu'au couloir formé par les deux guides virtuels et tente de traverser la porte. En effet, ce couloir virtuel permet au robot de se déplacer uniquement suivant des trajectoires qui sont possibles à l'intérieur. Les deux plans répulsifs qui forment ce couloir n'autorisent pas le robot virtuel à les traverser.

La figure 4.23 montre l'intervention des deux guides virtuels lorsque le robot mobile tente de traverser une porte.

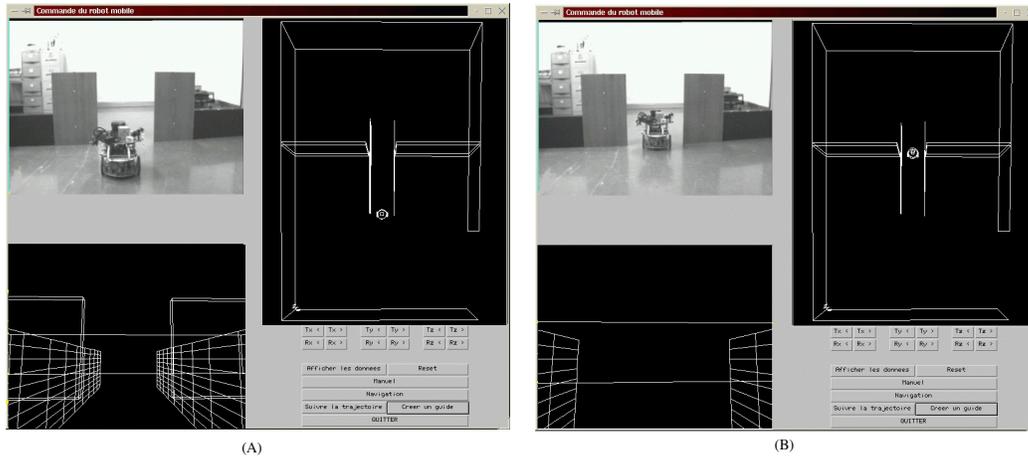


FIG. 4.23 – Illustration de la téléopération d'un robot mobile avec assistance de deux guides virtuels répulsifs sous forme de plans. (A) : Le robot mobile subit l'influence des deux guides, (B) : le robot traverse la porte.

Les propriétés des deux guides virtuels sont données par la figure 4.24-(A) et 4.24-(B)

4.2.3.2 tâche avec une courbe B-Spline attractive

Dans ce cas, l'opérateur téléopère le robot en lui faisant suivre une seule et unique trajectoire. Cette trajectoire est générée initialement par une B-Spline (présentée dans la section 2.2.1.3) et ensuite elle est modifiée (déformée) à la convenance de l'opérateur.

Ce type de guide virtuel, contraint le robot mobile à suivre une trajectoire précise que l'opérateur a choisi. Cette trajectoire peut être modifiée en ligne par l'opérateur s'il le désire.

La figure 4.25 montre une simulation du parcours réalisé par le robot mobile le long de ce guide virtuel. Nous présentons ici les captures d'écran de la fenêtre en haut à droite de l'interface (figure 4.22).

Les propriétés de ce guide sont données par la figure 4.26.

Nom du guide	Plan gauche.ob.i	Nom du guide	Plan droit.ob.i
Composition	Simple Compose	Composition	Simple Compose
Guide de reference		Guide de reference	
Guide precedent		Guide precedent	
Guide suivant		Guide suivant	
Etat du guide	Actif Passif	Etat du guide	Actif Passif
Influence du guide	Attractif Repulsif	Influence du guide	Attractif Repulsif
Action du guide	Guide libre	Action du guide	Guide semi autonome
Tx	211.44194	Tx	266.78984
Ty	365.7984	Ty	365.8981
Tz	-23.17554	Tz	-23.874849
Rx	0.0	Rx	0.0
Ry	0.5	Ry	0.0
Rz	0.0	Rz	0.0
Attachement	Guide libre	Attachement	Guide libre
Zone d'influence	Plan	Zone d'influence	Plan
Parametre 1:	300.0	Parametre 1:	300.0
Parametre 2:	50.0	Parametre 2:	50.0

(A)

(B)

FIG. 4.24 – Propriétés des deux guides virtuels utilisés pour traverser une porte par le robot mobile.

4.3 Autres applications potentielles

Il existe de nombreux secteurs d'activité (nucléaire, spatial, médecine, maintenance, enseignement...) où ce type de télétravail (télétravail, télétravail coopératif et l'utilisation des techniques de réalité virtuelle sur des machine à faible coût) peut apporter beaucoup.

4.3.1 Nucléaire

Sur les sites nucléaires, les robots et les systèmes téléopérés sont utilisés pour diminuer les doses de rayonnement, les contraintes physiques et/ou les risques pour les intervenants. La plupart des machines utilisées (pour l'IHM, Client/Serveurs, etc.) sont trop onéreuses du fait que les outils de développement et de simulation en réalité virtuelle sont fiables et stables sur ces machines. Par conséquent, les IHMs ne sont pas portables et donc la supervision et le contrôle d'un système (robot esclave par exemple) ne peut se faire que sur la même machine cliente (machine maître). Si cette dernière tombe en panne lors d'une mission de téléopération, il devient très difficile et encore très coûteux de reprendre à nouveau la mission surtout si la mission est vraiment urgente. Par ailleurs, il est difficile et très coûteux d'envisager un télétravail coopératif sachant que le système maître ne fonctionne que sur le même type de machine.

De nos jours, il existe des ordinateurs personnels plus performants (avec une puissance qui ne cesse d'augmenter), sous des environnements multi-tâches et multi-utilisateurs (tels que *Linux* ou encore *Linux temps réel*). De même, la maturité des outils de simulation sur PC (OpenGL, VRML, etc) et la stabilité des outils développement sur Internet en particulier le Langage *Java* offre la possibilité d'étendre les techniques de téléopération vers le télétravail coopératif.

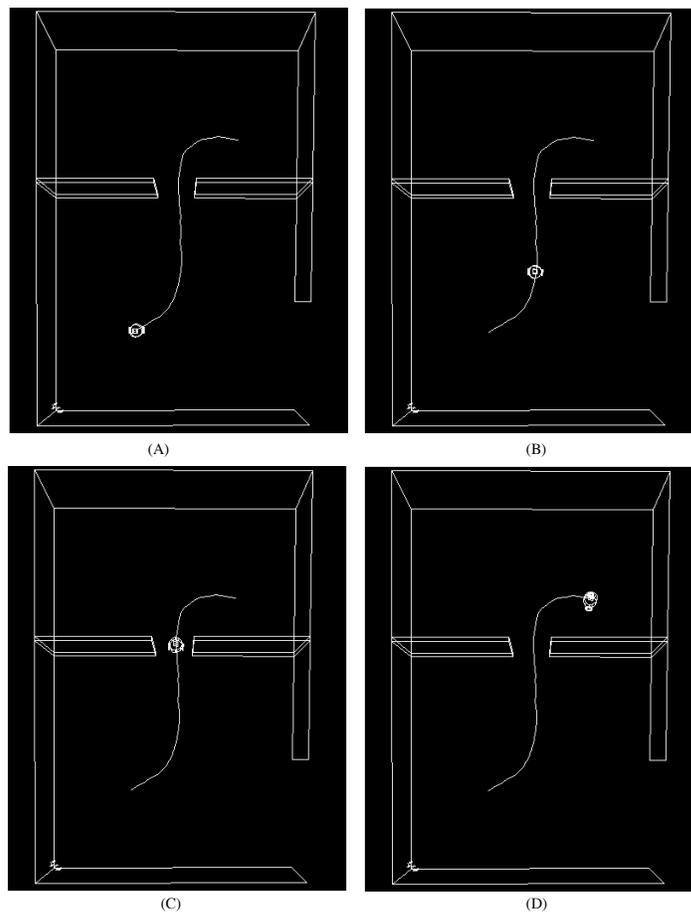


FIG. 4.25 – Simulation du parcours d'une B-Spline par le robot mobile virtuel. (A) : l'état initial, (B) : pendant le parcours, (C) : le robot traverse la porte, (D) : l'état final

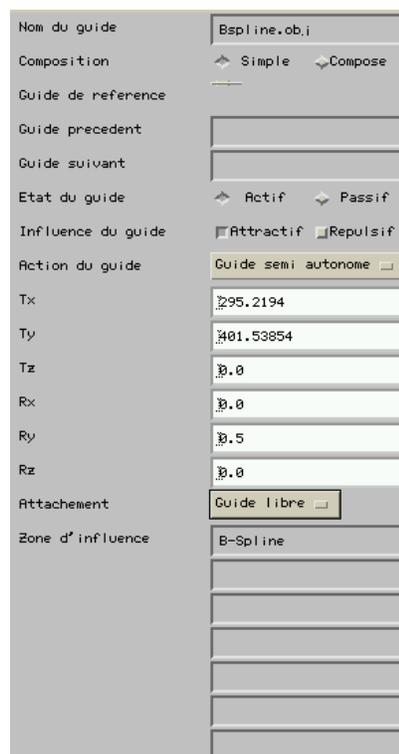


FIG. 4.26 – Propriétés du guide virtuel généré par une B-Spline

Le laboratoire CEREM¹ (Centre d'Etude et de Recherche des Matériaux) de la DTA (Direction des Technologies Avancées) envisage en perspective l'exploitation de cette voie pour étendre leurs techniques de téléopération. En effet ils ont annoncé que *dans un futur proche, le développement des communications sur les réseaux généralistes comme Internet, sont une formidable opportunité pour étendre les techniques de téléopération, afin d'aborder le domaine en pleine expansion du télétravail et du travail coopératif.*

CEREM participe en particulier sur ce sujet, au pôle TRV (Téléopération et Réalité Virtuelle) du programme de recherche CNRS "*machines Intelligentes*" avec le Laboratoire de Robotique de Paris (LRP de l'Université de Versailles) et le Centre d'Etudes en Mécanique d'Ile de France (CEMIF de l'Université d'Evry Val d'Essonne).

4.3.2 Médical

L'utilisation des systèmes de télétravail, peut être intéressante dans différents niveaux d'applications médicales. En effet, la Télésupervision par exemple permettra aux étudiants en médecine (se trouvant dans le même établissement ou encore dans le monde entier) d'assister en direct à une opération chirurgicale sans pour avoir à se déplacer en salle d'opération.

Un autre exemple est celui d'assister le chirurgien par d'autres chirurgiens plus expérimentés ne se trouvant pas au même endroit (leurs conseils peuvent être utiles dans le cas d'une opération délicate).

La téléchirurgie semble avoir un avenir prometteur, bien que l'idée de remplacer un chirurgien par un robot n'est pas la priorité de tout le monde. Mais ce domaine en cours de maturité peut trouver rapidement sa place en mettant le système expérimental à la disposition des autres chercheurs dans le but d'un télétravail coopératif.

4.3.3 Enseignement

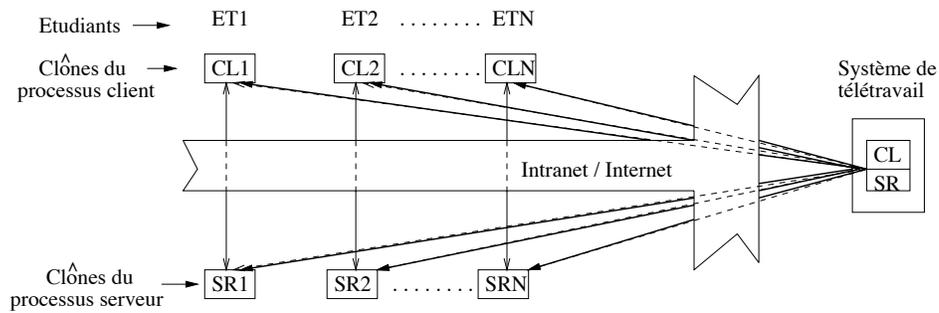
En plus de l'enseignement théorique donné aux étudiants (formation aux méthodes et aux techniques de la robotique, réseau et informatique, etc.), des Travaux Pratiques (TPs) moins coûteux peuvent être réalisés en utilisant de tels systèmes.

Prenons par exemple un cours de robotique sur la calibration de caméras, après avoir étudié les différentes méthodes, les étudiants peuvent expérimenter leurs méthodes et les tester en temps réel pendant une séance de TP. En effet, les étudiants ne sont pas obligés d'avoir une caméra réelle à disposition, puisque une image réelle capturée suffit.

Comme il s'agit du télétravail, d'autres établissements ne disposant pas de moyens (caméras, robot, etc.) peuvent très bien profiter de cet enseignement à condition que l'établissement propriétaire du système les autorise.

Il ne peut pas y avoir de défaillance du système original du moment que les étudiants travailleront sur des processus clones (une copie du système original, figure 4.27) et éventuellement un clone par étudiant afin d'éviter d'un côté, les mélanges et les conflits d'accès aux ressources du système et de l'autre côté un travail individuel sécurisé.

1. <http://www-dta.cea.fr/CEREM/UK/Pages/teleoper.htm>

FIG. 4.27 – *Illustration du télé-enseignement*

4.4 Bilan

Dans ce chapitre, nous avons présenté quelques applications pouvant bénéficier d'un système de télétravail via Internet. Nous avons cependant souligné la nécessité d'avoir d'un côté, une interface administrateur pour la maintenance et la mise à jour d'un tel système et de l'autre côté une interface destinée aux utilisateurs du système.

Nous avons décrit certaines tâches réalisées suivant le mode de télétravail choisi, d'un côté, les tâches de téléopération (saisie/dépôt d'objets, etc.) où l'opérateur s'il le désire peut être assisté par des guides virtuels. D'un autre côté, l'opérateur peut télétravailler à niveau plus haut, pour cela, il spécifie seulement la tâche à faire exécuter par les robots virtuel et réel simultanément (il s'agit dans ce cas des tâches pré-programmées). Enfin, si une mission nécessite l'intervention de plusieurs opérateurs simultanément dans le but d'un travail coopératif par exemple, un opérateur prend le contrôle du robot réel pendant que les autres supervisent en virtuel et en réel les actions de celui-ci. Dans ce dernier cas, le contrôle des robots virtuel et réel est partagé entre les différents opérateurs.

Nous avons montré la flexibilité et l'ouverture du système ARITI pour d'autres applications, en utilisant un robot mobile. En effet, une application sur un robot mobile a été réalisée en s'inspirant de l'existant et en rajoutant essentiellement le module contenant les nouveaux modèles virtuels (environnement et robot mobile) ainsi que le module de client/serveur commande (pour gérer les interactions avec le robot mobile réel).

Nous avons cité d'autres domaines d'application dans lesquels un système de télétravail ou de travail coopératif via Internet peut être intéressant, mais cette liste n'est pas exhaustive.

Par ailleurs, il est nécessaire d'une part, d'évaluer les performances des tâches réalisées avec ce système ainsi que les différents algorithmes de compression d'images utilisés. D'autre part, d'analyser le comportement de ce système et les performances de l'architecture client/serveur du système ARITI à la fois, en fonction de la distance séparant le site maître (client) du site esclave et en fonction du nombre d'opérateurs utilisant simultanément le système pour un travail coopératif. Cette partie est présentée et discutée dans le chapitre suivant.

Chapitre 5

Evaluation et Expérimentation

Le chapitre précédent, nous a permis de présenter essentiellement, deux applications réalisées en utilisant le système de télétravail ARITI, à savoir, la télémanipulation avec un robot à 4 ddl et la téléopération d'un robot mobile. Nous avons présenté certaines tâches suivant le mode de télétravail choisi, téléopération, téléprogrammation de tâches et télécoopération/télesupervision (dans le but d'un travail coopératif par exemple). Nous avons décrit les guides virtuels d'assistance à la téléopération ainsi que les propriétés associées.

Dans ce chapitre, nous présentons les expérimentations réalisées avec le système ARITI utilisant un robot esclave à 4 ddl (site expérimental se trouvant dans notre laboratoire). Les modes de télétravail utilisés pour ces expérimentations sont la téléopération et la télécoopération/télesupervision.

Dans la première section de ce chapitre, nous évaluons les performances de certaines tâches avec le mode de téléopération en utilisant l'interface de ARITI. Pour cela, plusieurs sujets (opérateurs humains) ont participé aux différentes expérimentations qui consistent à réaliser des missions de téléopération d'une part, sans assistance des guides virtuels et d'autre part, avec assistance des guides virtuels répulsifs ou attractifs.

La deuxième section consiste à évaluer les différents algorithmes de compression d'images vidéo et ceci suivant que l'image vidéo présente peu ou beaucoup de mouvements. Il s'agit d'étudier d'une part, les taux de compression fournis par les différentes méthodes de compression d'images, sachant que les images compressées seront envoyées via Internet aux différents clients connectés au système ARITI. D'autre part, nous étudions les temps de compression d'images pour chaque méthode utilisée. Ces évaluations vont nous permettre de déterminer un algorithme de compression implanté dans le serveur d'images permettant ainsi de choisir automatiquement la méthode à utiliser suivant la dynamique des images (images avec peu ou beaucoup de mouvements).

Dans la troisième section, nous étudions le comportement du système et les performances de l'architecture client/serveur du système ARITI d'une part, en fonction de la distance séparant le site maître (client) du site esclave. D'autre part, en fonction du nombre d'opérateurs utilisant simultanément le système pour un travail coopératif avec un retour prédictif distribué.

5.1 Evaluation des performances des tâches avec l'interface de ARITI

5.1.1 Conditions expérimentales

Le site maître est un PC cadencé à 233 MHz connecté à Internet sur lequel un opérateur contrôle un robot virtuel. L'interface utilisée est celle destinée à l'opérateur dit "administrateur" décrite dans le chapitre précédent (chapitre Applications 4.1) et représentée par la figure 4.5.

Le robot esclave utilisé est un banc d'essai à 4 ddl (2 rotations et deux translations), transformé en un robot manipulateur décrit aussi dans le chapitre 4.1 et présenté par la figure 4.1. Les objets à manipuler sont des cylindres en polystyrène (numérotés 1, 2 et 3) accrochés sur des supports métalliques.

L'effecteur du robot est constitué d'une pointe métallique avec laquelle s'effectue la saisie des objets en polystyrène. La figure 5.1 donne une vue rapprochée de l'effecteur du robot et des trois cylindres en polystyrène.

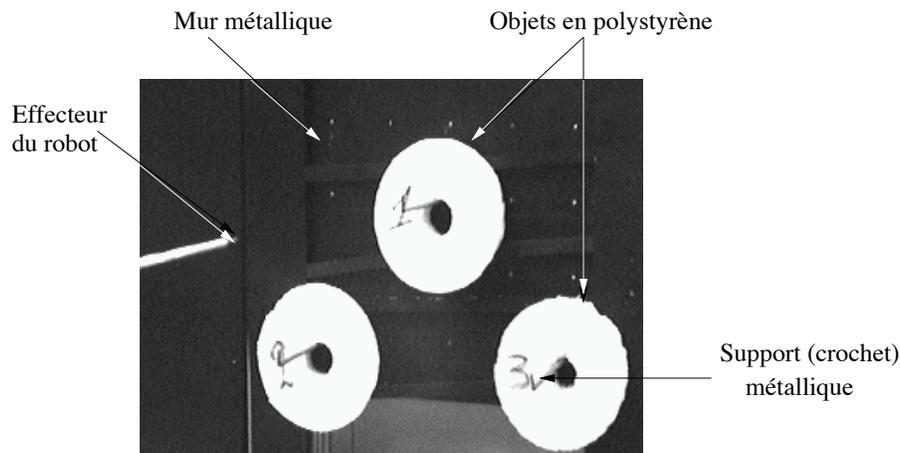


FIG. 5.1 – Vue rapprochée de l'effecteur du robot et des trois cylindres en polystyrène.

L'expérimentation est réalisée avec trois missions qui consistent à réaliser les tâches suivantes :

- Atteindre le cylindre numéro 1.
- Saisir le cylindre numéro 1 depuis son support.
- Déposer le cylindre numéro 1 sur le support numéro 3.

Le déroulement de chaque mission se fait généralement en deux étapes, préparation de la mission par le robot virtuel, ensuite l'exécution par le robot réel.

Soit T_p , le temps nécessaire à l'opérateur pour préparer une mission avec le robot virtuel, et T_e , le temps nécessaire pour exécuter la mission par le robot réel. Le temps de la réalisation de la mission est donné alors par le temps total de la mission T_m :

$$T_m = T_p + T_e \quad (5.1)$$

Les vitesses de déplacement et de rotation des différentes composantes du robot réel sont constantes (16.6 cm/s pour les translations et 16.6 degrés/s pour les rotations, voir le tableau 5.1).

Composantes du robot réel	Vitesse
Grande base	16.6 cm/s
Support	16.6 cm/s
Tige	16.6 degrés/s
Equerre	16.6 degrés/s

TAB. 5.1 – Vitesses de déplacement et de rotation des composantes du robot réel

Etant donné que la cible à atteindre est un point 3D situé à une distance d (constante) par rapport à la position de l'effecteur du robot. Alors le temps d'exécution T_e est identique pour chaque mission consistant à atteindre le cylindre numéro 1. Cependant, le temps de réalisation de la mission (T_m) pour atteindre le cylindre numéro 1 par exemple, dépend complètement du temps de la préparation de cette mission avec le robot virtuel (T_p).

La vitesse du déplacement du robot virtuel se fait suivant l'endroit où se trouve l'extrémité de son effecteur. En effet, l'environnement virtuel est découpé en trois zones (zone 1, zone 2 et zone 3) représentées dans la figure 5.2. Les pas de déplacement et de rotation du robot virtuel varient selon que l'effecteur du robot se rapproche ou s'éloigne de la cible.

Soit (X_2, Y_2, Z_2) les coordonnées du point P_2 appartenant au premier plan de la deuxième zone, et (X_3, Y_3, Z_3) , les coordonnées du point P_3 appartenant au premier plan de la troisième zone. Soit aussi $(X_{cible}, Y_{cible}, Z_{cible})$ les coordonnées du point cible situé à la périphérie de l'un des objets cylindres. Si (X_e, Y_e, Z_e) sont les coordonnées du point "E" représentant l'extrémité de l'effecteur du robot virtuel, alors les conditions pour que l'extrémité de la tige se trouve dans chacune des zones sont données par :

$$\begin{cases} X_e \in Zone1 & X_e \geq X_2 \\ X_e \in Zone2 & X_1 \leq X_e < X_2 \\ X_e \in Zone3 & X_{cible} \leq X_e < X_1 \end{cases} \quad (5.2)$$

les caractéristiques de déplacement et de rotation du robot virtuel sont données par le tableau 5.2.

Pas	Zone 3	Zone 2	Zone 1
Translation (cm)	2	0.4	0.1
Rotation (degrés)	2	0.4	0.1

TAB. 5.2 – Les pas de déplacement et de rotation du robot virtuel

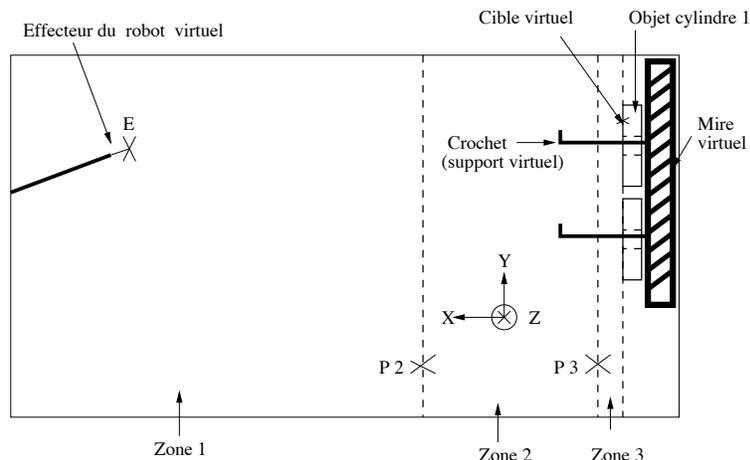


FIG. 5.2 – Illustration du découpage de l’environnement virtuel en trois zones

Dans tous les tests réalisés que nous allons présenter par la suite, dix opérateurs (étudiants IUP, stagiaires, doctorants) n’ayant subi aucun apprentissage des tâches ont participé à celles-ci. Chaque opérateur effectue une série de dix tests. Le mode de travail est la téléopération et le contrôle du robot se fait avec le clavier.

5.1.2 Atteinte d’une cible

La tâche consiste à faire atteindre l’objet cylindre numéro 1, par le robot à partir de sa position initiale. Cette position de repos se trouve à 1 m environ de l’objet à atteindre. Pour évaluer cette tâche, trois types de tests ont été réalisés. Le premier type de tests est réalisé sans guide virtuel. Dans le second, l’opérateur utilise un guide virtuel répulsif, l’effecteur du robot peut se déplacer librement à l’intérieur du guide, mais il ne peut pas le traverser. Le troisième type de tests est réalisé en utilisant un guide virtuel attractif, dans ce cas, les actions de l’opérateur sont combinées avec une fonction d’attraction vers l’axe central du guide. En effet, dans ce cas, l’extrémité de l’effecteur du robot est projeté à chaque mouvement de l’opérateur, sur l’axe central du guide virtuel.

5.1.2.1 Tests sans guide virtuel

Pour permettre à l’opérateur de voir où se trouve la cible, un signal visuel est affiché (petit cercle autour de la cible) à la périphérie du cylindre numéro 1 (voir la figure 5.3).

Pendant ces tests, les éventuelles collisions de l’effecteur du robot avec le support métallique (où se trouve l’objet à atteindre) ont été gérées. Le nombre de collisions est calculé pour chaque opérateur et pour chaque test (figure 5.4-(B)). Dès qu’une collision se produit, le message “collision” apparaît sur l’écran ainsi qu’une flèche à l’endroit de la collision (figure 5.4-(A)). Ces informations permettent à l’opérateur de corriger ses mouvements.

La figure 5.4-(B), montre qu’il y a en moyenne 14.9 collisions pour les dix tests soit une collision par test.

Nous avons aussi mesuré les imprécisions avec lesquelles la cible a été atteinte par l’effecteur du robot, ainsi que le temps mis par chaque opérateur pour atteindre la cible. La figure 5.5-(A), montre la moyenne des imprécisions (en mm) sur les dix tests pour chaque

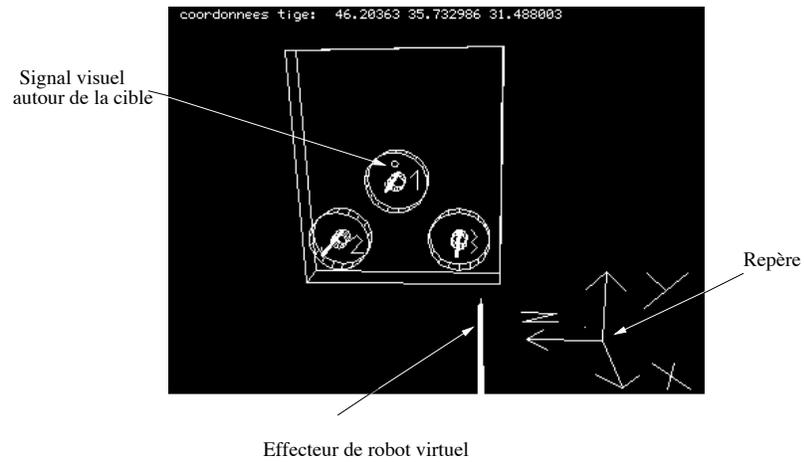


FIG. 5.3 – Illustration du signal visuel utilisé pour atteindre le cylindre numéro 1

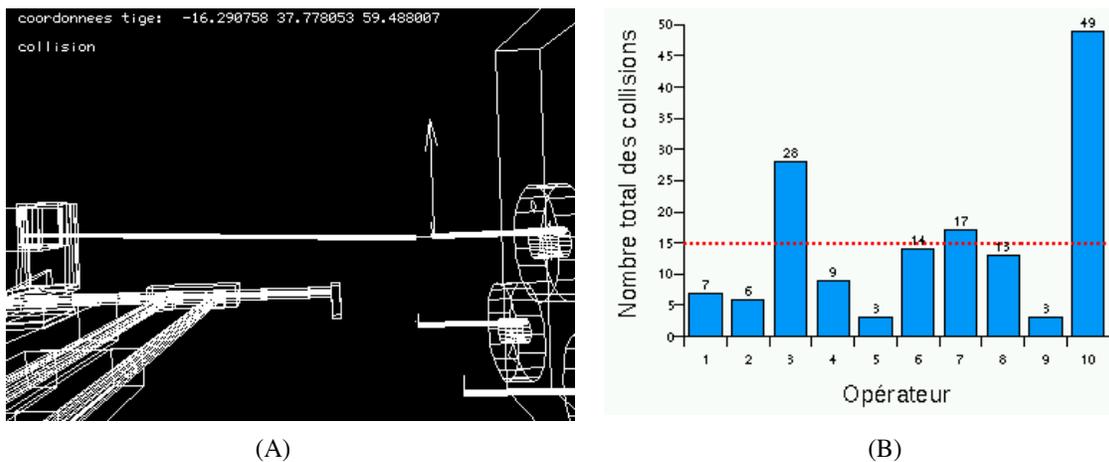


FIG. 5.4 – Gestion des collisions pendant que l'opérateur essaie d'atteindre le cylindre numéro 1. (A): détection des collisions, (B): le nombre total de collisions réalisé par chaque opérateur.

opérateur par rapport à X , Y et Z . L'imprécision moyenne de tous les opérateurs est donnée par la figure 5.5-(B).

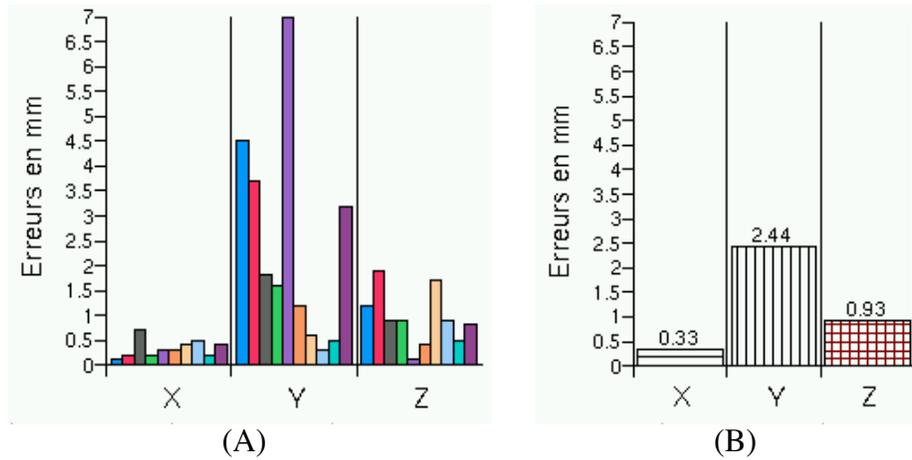


FIG. 5.5 – Les imprécisions obtenues lorsque la cible est atteinte. (A) : la moyenne des dix tests pour chaque opérateur suivant X , Y et Z , (B) : la moyenne sur les dix opérateurs suivant X , Y et Z .

Nous remarquons que les imprécisions sur l'axe "Y" sont plus importantes, elles correspondent aux erreurs de rotation en site de l'effecteur du robot. En effet, pour éviter de prendre le risque de provoquer des collisions entre l'effecteur du robot et le support où se trouve l'objet à atteindre, les différents sujets (opérateurs) préfèrent maintenir l'effecteur éloigné du support.

Nous avons mesuré les temps (en secondes) réalisés par chaque opérateur à chaque essai. La figure 5.6 donne la moyenne des temps obtenus par les dix opérateurs à chaque essai.

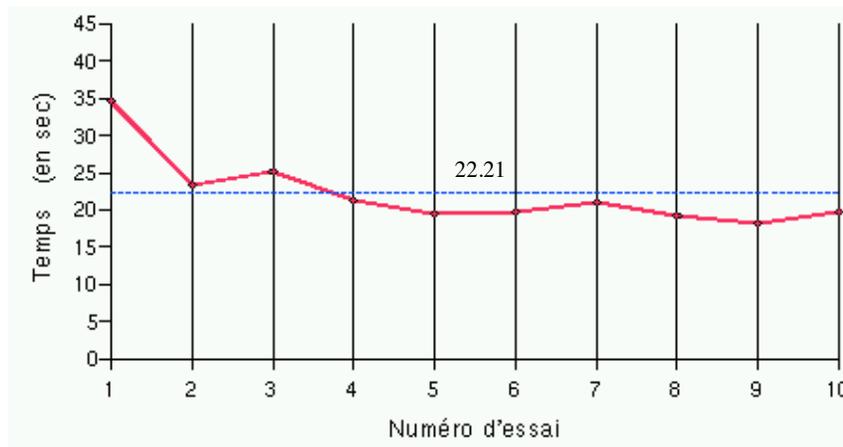


FIG. 5.6 – Le temps moyen pour atteindre la cible (cylindre numéro 1) à chaque essai.

La figure 5.6, montre que, pendant les trois premiers tests les temps moyen sur les dix opérateurs sont plus importants (35, 23 et 25 sec). Ce temps a tendance à diminuer dans la suite des autres tests, ce qui montre en effet, l'adaptation des opérateurs à l'interface de

commande ainsi que leur apprentissage. Cependant, le temps moyen sur les dix opérateurs et sur les dix tests pour atteindre le cylindre numéro 1, est de 22.21 sec.

5.1.2.2 Tests avec guide virtuel répulsif

Le guide virtuel utilisé est un cône (appelé “tube” dans la librairie des guides virtuels) attaché à l’objet cylindre numéro 1. Ce guide virtuel est activé en répulsion, cependant, lorsque l’effecteur du robot virtuel se trouve à l’intérieur du guide, ses degrés de liberté sont limités au volume de ce guide qui définit l’espace de travail (l’effecteur du robot ne peut pas traverser le guide).

La figure 5.7 montre ce guide virtuel attaché au cylindre numéro 1.

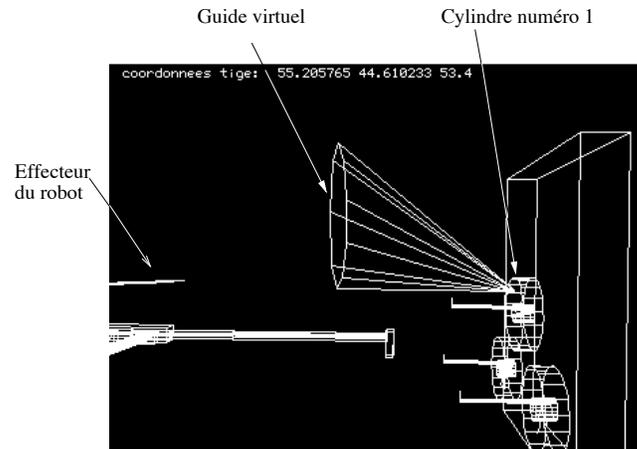


FIG. 5.7 – Utilisation d’un guide virtuel pour atteindre le cylindre numéro 1

Les imprécisions obtenues avec ce guide virtuel sont représentées dans la figure 5.8.

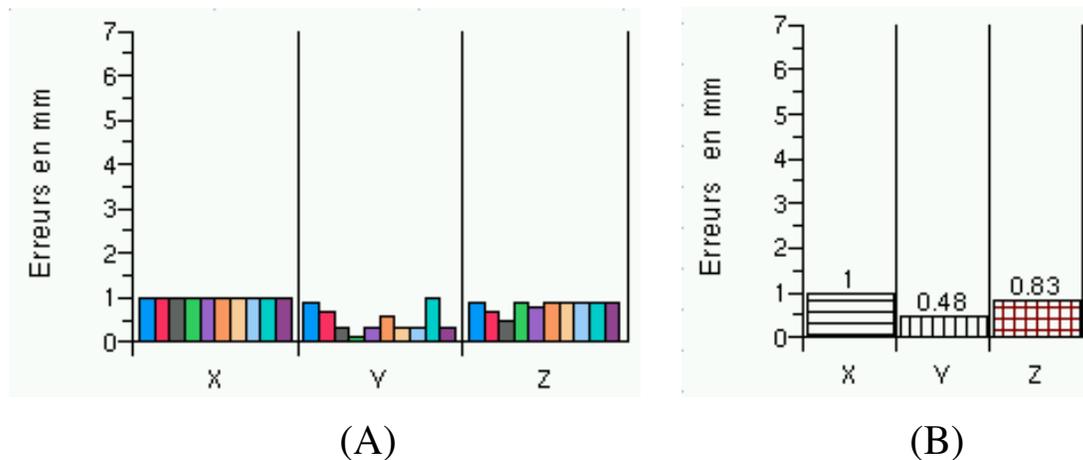


FIG. 5.8 – Les imprécisions obtenues lorsque la cible est atteinte en utilisant un guide virtuel répulsif. (A) : la moyenne des dix tests pour chaque opérateur suivant X, Y et Z, (B) : la moyenne sur les dix opérateurs suivant X, Y et Z

Contrairement aux tests réalisés précédemment (sans assistance d’un guide virtuel), les risques de collisions sont exclus. Par ailleurs, les imprécisions pour atteindre la cible avec assistance d’un guide virtuel répulsif, sont nettement inférieures par rapport à l’axe “Y” si nous les comparons aux tests réalisés sans guide virtuel. En effet, la moyenne des

imprécisions avec guide répulsif est de 0.48 mm contre 2.44 mm sans guide virtuel. Cette diminution s’explique par le fait que les opérateurs ont tendance à rapprocher davantage l’effecteur du robot vers le support virtuel, sachant que le guide les empêcherait de rentrer en collision avec ce support.

Nous remarquons aussi que les imprécisions sur l’axe des “X” est de 1 mm pour tous les opérateurs (la cible est atteinte de la même façon pour tous). Ceci est une conséquence directe sur le comportement identique concernant le positionnement de l’effecteur du robot à l’intérieur du guide virtuel. En effet, comme l’effecteur du robot est souvent positionné juste au dessus du support virtuel (mais toujours à l’intérieur du guide), pour déplacer l’effecteur vers la cible, une translation en “X” suffit. Cependant, comme le pas de déplacement est le même dans cette zone, alors ils y arrivent tous de la même façon.

Par contre, le temps nécessaire pour atteindre le cylindre numéro 1 avec un guide virtuel répulsif est nettement plus important que sans guide virtuel. Le temps moyen sur les dix opérateurs et sur les dix tests est de 31.48 sec (figure 5.9) contre 22.21 sec lorsque l’opérateur n’est pas assisté par un guide virtuel répulsif. Cette perte de temps, est tout simplement provoquée par des différentes collisions entre l’effecteur du robot et le guide virtuel répulsif. En effet, ce dernier a tendance à repousser l’effecteur du robot à chaque fois que l’opérateur tente de le traverser involontairement.

Par ailleurs, bien que ce type de guide (répulsif), fournit une sécurité pour l’effecteur du robot (pas de collisions avec le support), son utilisation rend l’apprentissage difficile (voir la courbe de la figure 5.9).

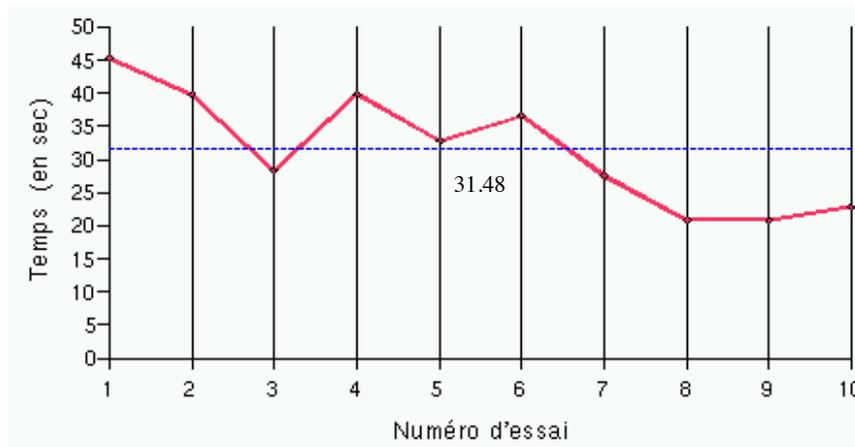


FIG. 5.9 – Le temps moyen pour atteindre la cible (cylindre numéro 1) à chaque essai en utilisant un guide virtuel répulsif

5.1.2.3 Tests avec guide virtuel attractif

Dans cette série de tests, les opérateurs utilisent un guide virtuel attractif. Il s’agit du même guide virtuel utilisé dans la série de tests précédente, sauf que dans ce cas le guide devient actif en attraction. En effet, dès que l’effecteur du robot se trouve à l’intérieur de ce guide, les actions de l’opérateur sont combinées avec une fonction d’attraction vers l’axe central du guide.

Les imprécisions obtenues avec ce guide virtuel sont représentées dans la figure 5.10.

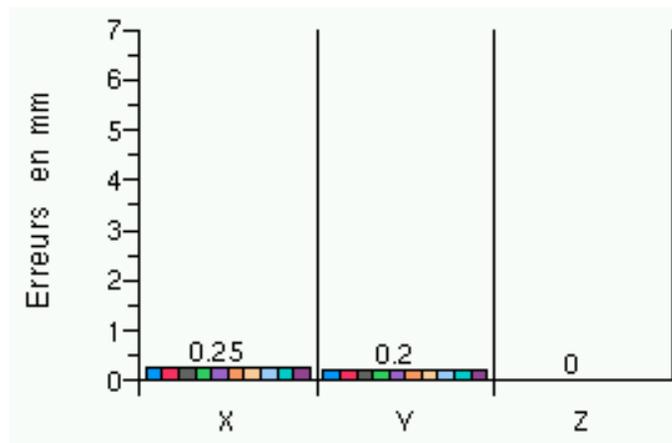


FIG. 5.10 – Les imprécisions obtenues suivant X , Y et Z , lorsque la cible est atteinte en utilisant un guide virtuel attractif

La figure 5.10, montre que les imprécisions pour atteindre le cylindre numéro 1 avec un guide virtuel attractif, sont très petites sur les axes “ X ”, “ Y ” (0.25 mm et 0.2 mm respectivement) et sont nulles sur l’axe “ Z ”. En effet, le fait d’avoir un guide virtuel attractif dont la fonction consiste à projeter l’extrémité de l’effecteur du robot sur l’axe central de ce guide et le fait, que cet axe passe par le point cible en question, alors la composante en “ Z ” de l’extrémité de l’effecteur du robot ne peut être que zéro. Par ailleurs, la cible est atteinte (en “ X ” et “ Y ”) de la même façon par tous les opérateurs, du moment qu’ils subissent tous la même fonction du guide.

En plus, l’utilisation de ce type de guide virtuel (attractif), fourni un temps d’atteinte de la cible très petit par rapport aux tests effectués sans guide et avec guide virtuel répulsif. Le temps moyen obtenu sur les dix opérateurs et sur les dix tests, est de 7.7 sec (figure 5.11) contre 31.48 sec (avec guide répulsif) et 22.21 sec (sans assistance du guide virtuel).

D’un autre côté, ce guide attractif, fourni un très bon apprentissage comme le montre la figure 5.11.

5.1.3 Saisie et dépôt d’un objet

Dans cette section, nous présentons les tests réalisés en utilisant des guides virtuels (répulsifs ou attractifs) pour saisir le cylindre numéro 1 depuis son support, pour ensuite le déposer sur le support numéro 3.

5.1.3.1 Saisie du cylindre numéro 1

Dès que le cylindre numéro 1 est atteint (tâche atteinte d’une cible), un guide virtuel composé de trois guides simples apparaît pour assister l’opérateur à prendre l’objet cylindre et le sortir de son support métallique.

Ce guide virtuel et son formalisme sont décrits dans le chapitre précédent, voir aussi la figure 4.16 dans la section 4.1.

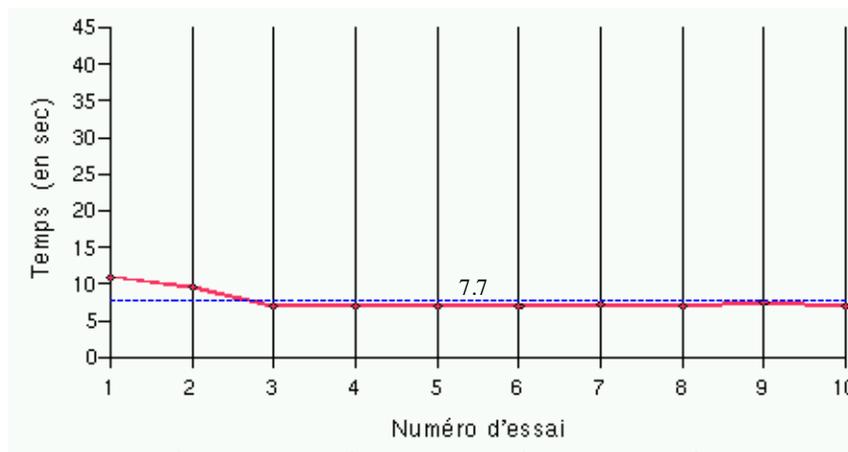


FIG. 5.11 – Le temps moyen pour atteindre la cible (cylindre numéro 1) à chaque essai en utilisant un guide virtuel attractif

La figure 5.12 montre ce guide virtuel attaché à l'objet cylindre numéro 1, permettant de prendre ce cylindre et de le sortir de son support métallique.

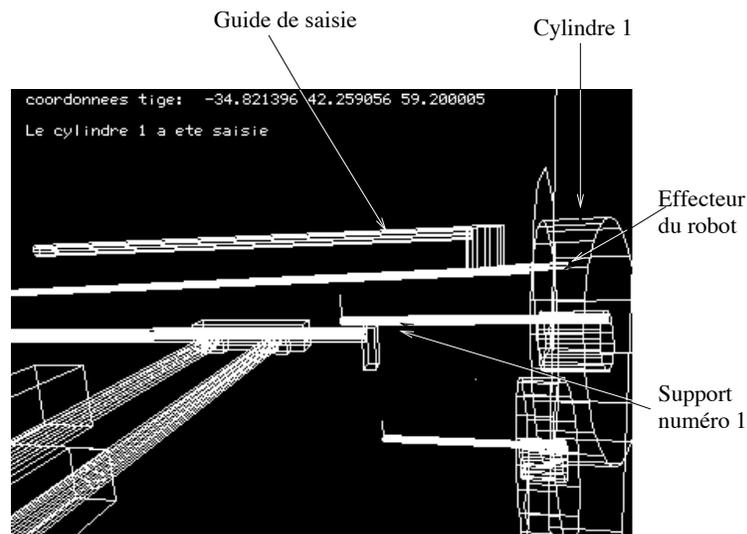


FIG. 5.12 – Utilisation d'un guide virtuel pour saisir et sortir le cylindre numéro 1 depuis son support métallique.

La moyenne des temps sur dix opérateurs et à chaque essai pour réaliser cette tâche est représentée par la figure 5.13. Avec un guide virtuel répulsif le temps moyen sur les dix essais est de 12.78 sec alors que avec un guide virtuel attractif ce temps est diminué (9.5 sec).

5.1.3.2 Dépôt du cylindre numéro 1 sur le support numéro 3

Afin de déposer l'objet saisi (cylindre numéro 1) sur un support métallique (numéro 3), l'opérateur dirige l'effecteur du robot (contenant l'objet) vers ce support. Un guide virtuel composé de quatre guides simples apparaît pour l'aider à déposer l'objet déjà saisi. Cette opération consiste donc à décrocher le cylindre de l'effecteur du robot en utilisant le support métallique numéro 3.

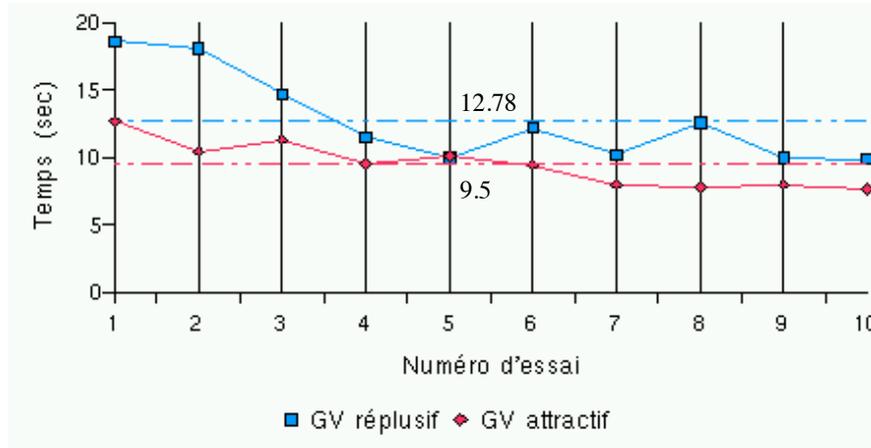


FIG. 5.13 – Le temps moyen pour saisir le cylindre numéro 1 à chaque essai en utilisant un guide virtuel répulsif ou attractif. En pointillé sont représentées les moyennes pour les dix opérateurs.

Ce guide virtuel apparaît au dessus du support numéro 3 comme le montre la figure 5.14.

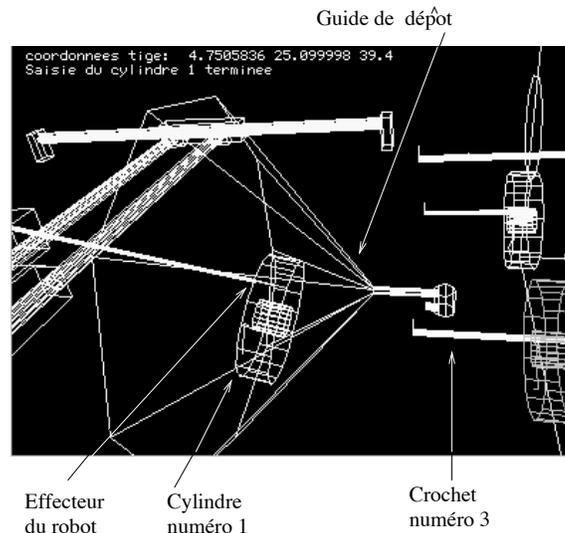


FIG. 5.14 – Utilisation d'un guide virtuel pour déposer le cylindre numéro 1 sur le support métallique numéro 3.

La moyenne des temps sur dix opérateurs et à chaque essai pour réaliser cette tâche est représentée par la figure 5.15. Avec un guide virtuel répulsif le temps moyen sur les dix essais est de 37.96 sec alors que avec un guide virtuel attractif ce temps est réduit à 16.86 sec.

A l'issue de ces différentes expérimentations, à savoir réaliser des missions de téléopération, sans assistance et avec assistance des guides virtuels (répulsifs ou attractifs), nous avons essentiellement souligné et évalué quatre critères qui sont :

- **Sécurité** : Protéger le robot et les objets manipulés contre d'éventuelles collisions vis à vis de l'environnement.
- **Précision** : Approcher l'objet à manipuler avec une bonne précision.

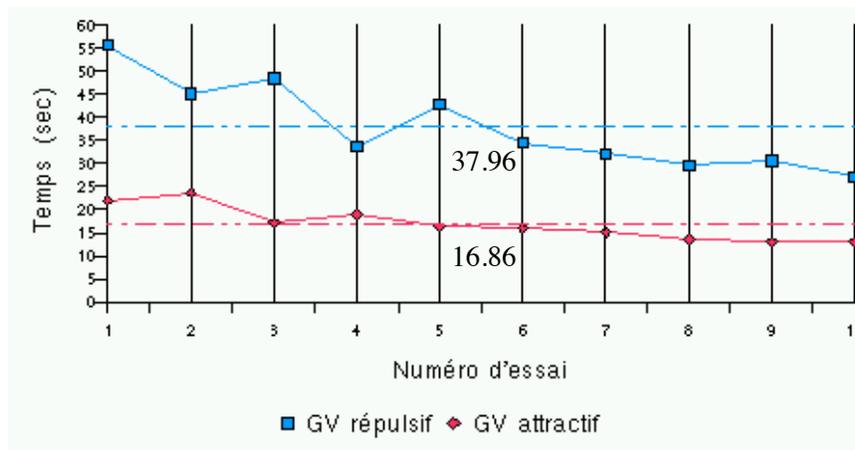


FIG. 5.15 – Le temps moyen pour déposer le cylindre numéro 1 sur le support numéro 3 à chaque essai en utilisant un guide virtuel répulsif ou attractif. En pointillé sont représentées les moyennes sur les dix opérateurs

- **Temps d'exécution :** Réaliser la mission avec un meilleur temps.
- **Apprentissage :** Faciliter l'apprentissage de la préparation de missions.

Par ailleurs, les résultats de ces expérimentations, ont montré d'une part, que l'interface du système ARITI permet un apprentissage facile pour la préparation des missions de téléopération. D'autre part, l'utilisation des guides virtuels garantit d'un côté, la sécurité du robot et de son environnement et de l'autre côté, lorsque les guides virtuels sont activés en attraction, ils fournissent une meilleure précision d'approche et un meilleur temps d'exécution.

D'une manière générale, nous avons évalué l'interface et les outils d'assistance virtuels pour la préparation et l'exécution des missions de téléopération. Nous allons maintenant nous intéresser au retour d'images vidéo et essentiellement à l'évaluation des algorithmes de compression d'images utilisés. Cette partie est présentée dans la section suivante.

5.2 Évaluation des algorithmes de compression d'images vidéo

Dans cette section nous présentons les résultats d'évaluation des algorithmes de compression d'images utilisés. Nous allons évaluer essentiellement les taux et les temps de compression d'image vidéo.

5.2.1 Conditions expérimentales

La capture d'images se fait sur un PC Pentium 233 MHZ via une carte d'acquisition vidéo Matrox Meteor reliée à une caméra noir et blanc. La taille des images est de 49152 octets (256).

Deux algorithmes de compression sont utilisés (pour plus de détails, voir annexe A):

- **Gzip :** Utilise la librairie "Zlib" supportée par la plupart des navigateurs Internet (Gzib, www).

- **Huffman** : Utilise un codage particulier, celui de représenter chaque élément de l'image par une série de "0" et de "1" (codage binaire) (Binstock et Rex, 1995) (Nelson et Gailly, 1996)

L'évaluation de ces algorithmes est faite sur une série de dix images vidéo capturées (généralement les dix premières images). Un pré-traitement (différence entre deux images) de l'image a été réalisé et combiné avec les deux méthodes de compression.

5.2.1.1 Evaluation des taux de compression d'image

Il s'agit ici de voir quelle méthode offre un meilleur taux de compression d'images. Nous avons testé trois types de compression, "Gzip", "Pré-traitement + Gzip" et "Pré-traitement + Huffman".

Des tests ont été réalisés suivant la dynamique de l'image (quantité de mouvements dans l'image) c'est à dire, lorsque l'image présente peu de mouvements ou beaucoup de mouvements.

Les résultats de compression donnés dans la figure 5.16, représente la taille moyenne des dix premières images compressées par chaque méthode et suivant la dynamique de l'image.

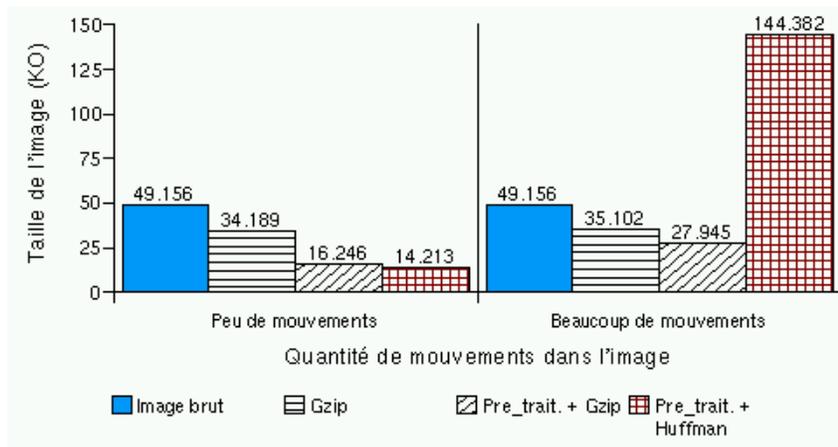


FIG. 5.16 – Evaluation des algorithmes de compression suivant la dynamique de l'image

Nous remarquons que l'algorithme de Huffman n'est utile que si les images à compresser ne contiennent pas beaucoup d'éléments différents (il est souvent précédé par un pré-traitement comme le fait JPEG, MPEG par exemple).

Les résultats obtenus montre que lorsque l'image vidéo présente peu de mouvements la méthode de compression "Pré-traitement + Huffman" donne un meilleur taux de compression, la taille de l'image est diminuée de 71%. Par contre, lorsque l'image présente beaucoup de mouvements, c'est la méthode "Pré-traitement + Gzip" qui fournit le meilleur taux de compression (44%).

Nous allons maintenant évaluer les temps de compression pour chaque méthode.

5.2.1.2 Evaluation des temps de compression d'image

Le temps que met chaque méthode pour compresser l'image est évalué sur les dix premières images capturées. Cette évaluation fait intervenir les trois méthodes de compression dans le cas où l'image présente peu de mouvements. Par contre lorsque l'image présente beaucoup de mouvements, seulement deux méthodes sont utilisées ("Gzip", "pré-traitement + Gzip"). En effet, dans ce dernier cas la méthode "pré-traitement + Huffman" ne présente pas d'intérêt étant donné que son utilisation fournit un taux de compression largement supérieur à la taille originale de l'image (300%).

Les figures 5.17 et 5.18, montrent respectivement les temps de compression nécessaires pour chaque méthode avec peu de mouvements et beaucoup de mouvements dans les images.

Nous remarquons que lorsque l'image présente peu de mouvements, la compression "Pré-traitement + Huffman" fournit un meilleur temps, en moyenne "30ms" contre "59 ms" pour "Pré-traitement + Gzip" et "80 ms" pour "Gzip". En revanche, lorsque l'image présente beaucoup de mouvements, le temps de compression par "Pré-traitement + Gzip" est de "70 ms" contre "74 ms" pour "Gzip".

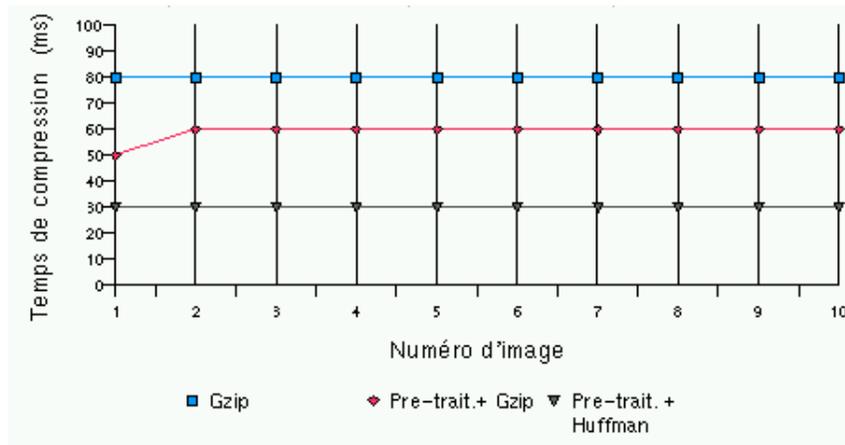


FIG. 5.17 – Les temps de compression lorsqu'il y a peu de mouvements dans l'image.

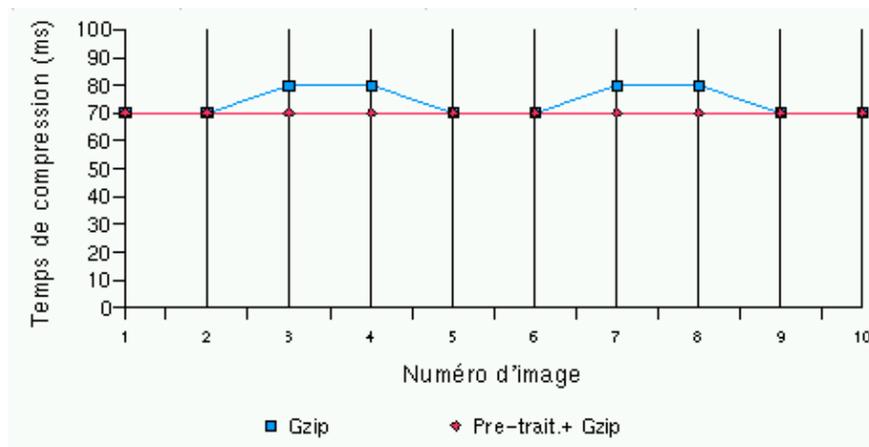


FIG. 5.18 – Les temps de compression lorsqu'il y a beaucoup de mouvements dans l'image.

L'analyse des résultats obtenus nous a permis d'implanter un algorithme de choix automatique de la méthode de compression à utiliser. En effet, il suffit de déterminer un seuil qui permet de départager l'appartenance de l'image à une des catégories peu ou beaucoup de mouvements. Si l'image présente peu de mouvements alors la méthode "pré-traitement + Huffman" est choisie, sinon c'est la méthode "pré-traitement + Gzip" qui sera utilisée.

Ce seuil, est déterminé expérimentalement par le nombre de pixels (niveau de gris) différents dans l'image que nous avons appelé Nombre de Types d'Eléments dans l'image (NTE).

Pour une image à 256 niveaux de gris ce seuil est "128", si NTE est supérieur à 128 alors l'image est considérée dans la catégorie des images présentant beaucoup de mouvements, sinon elle est dans celles présentant peu de mouvements.

L'organigramme de choix automatique de la méthode de compression est donné par la figure 5.19. Avec I désigne l'image brut, I_p l'image après pré-traitement, I_z image après Gzip, I_{pz} image après pré-traitement et Gzip, I_{ph} image après pré-traitement et Huffman. Les codes "2", "3" et "5", représentent respectivement les types de compression "Gzip", "Pré-traitement + Gzip" et "Pré-traitement + huffman".

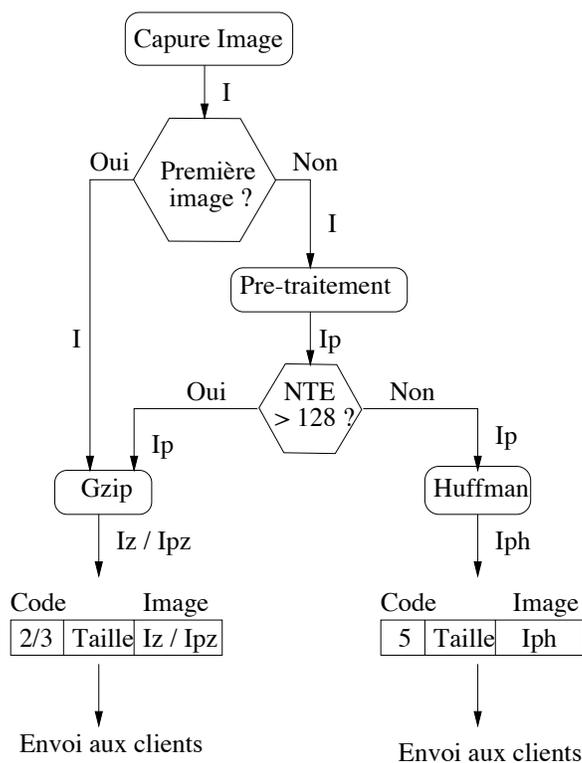


FIG. 5.19 – Organigramme de choix automatique de méthodes de compression d'image. l'image est envoyée aux clients avec le type de compression utilisé.

5.3 Evaluation des performances de l'architecture - client/serveur de ARITI

De nombreuses expériences ont été réalisées afin d'évaluer les performances de l'architecture client/serveur du système ARITI. Une série de tests a été réalisée en locale au sein même du laboratoire, et d'autres séries en moyenne et longue distance.

Nous avons étudié principalement les performances du serveur vis à vis de l'ensemble des connexions qui pourraient avoir lieu. Nous avons mesuré à chaque fois le temps serveur (T_s) et le temps de réponse du processus client (T_r). Nous considérons que le temps total effectif (T_t) est compris entre le temps de la réception d'une demande du client et le temps de la réception d'une nouvelle demande de ce client, ce qui donne :

$$T_t = T_s + T_r \quad (5.3)$$

Dans le cas du serveur d'image vidéo, le temps serveur est donné par :

$$T_s = T_i + T_c + T_e \quad (5.4)$$

avec

- T_i : temps de capture de l'image.
- T_c : temps de compression de l'image.
- T_e : temps de l'envoi de l'image dans le buffer (zone mémoire) de la socket TCP/IP.

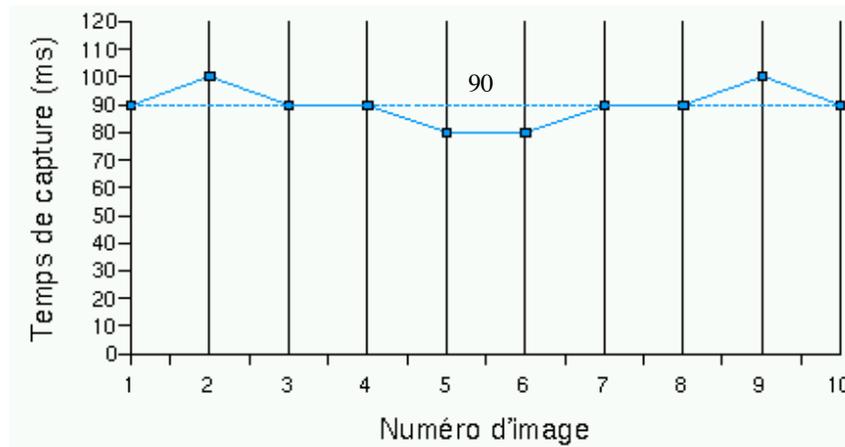


FIG. 5.20 – Evaluation du temps de capture d'images vidéo T_i avec dix images successives.

Le temps d'envoi dépend de la méthode de compression utilisée, et il varie de la même façon que le temps de capture d'image représenté dans la section cf. § 5.2 (voir aussi les figures 5.17, 5.18).

Le temps de réponse du processus client est donné par :

$$T_r = \delta_T + T_d + \delta_t \quad (5.5)$$

avec

- δ_T : temps de transfert de l'image, il s'agit du délai Internet/Intranet et dépend complètement de la bande passante.

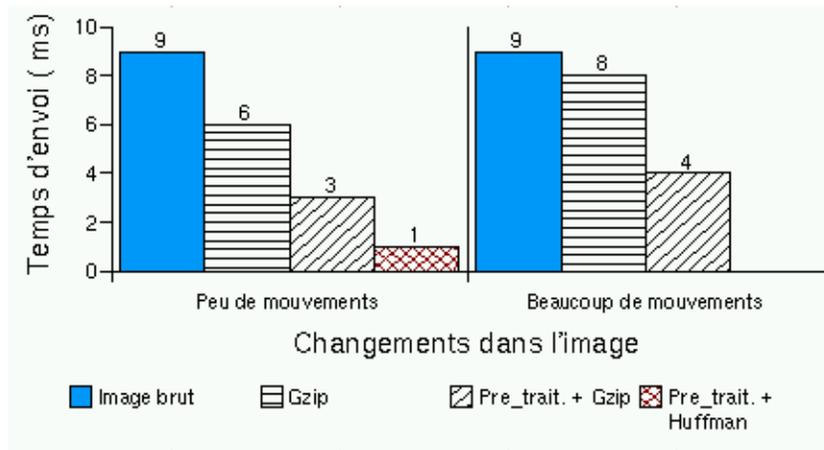


FIG. 5.21 – Evaluation du temps d’envoi T_e suivant la dynamique de l’image. Le temps représenté est une moyenne sur dix images envoyées.

- T_d : temps de décompression de l’image du côté client.
- δ_t : temps d’envoi d’une nouvelle demande d’images au serveur. Il s’agit du temps nécessaire pour le transfert du numéro de client (1 octet) vers le serveur.

Dans un premier temps nous présentons les résultats obtenus lorsque le système ARITI est utilisé par un seul opérateur se connectant en local depuis une autre machine interne au laboratoire. Par la suite, nous présentons les résultats obtenus lorsque l’opérateur se trouve à moyenne (France, Italie, Espagne, etc.) et à longue distance (Canada, Washington, Argentine, Brésil, etc.).

Dans le but d’un télétravail coopératif, des tests avec plusieurs opérateurs connectés simultanément au système, ont été réalisés en locale, moyenne et longue distance.

L’objectif de ces tests est d’un côté, d’analyser le comportement et les performances du serveur image de ARITI, lorsque les sites clients se trouvent à des endroits différents dans le monde, et pour cela, nous avons évalué le temps serveur (T_s) et le temps de réponse des clients (T_r). De l’autre côté, nous avons mesuré les temps moyens nécessaires pour la mise à jour, des images vidéo et des environnements virtuels par rapport aux différents site clients.

5.3.1 Connexion avec un seul opérateur

5.3.1.1 Connexion locale

Dans ce cas, l’opérateur est connecté au système ARITI en utilisant un PC Pentium 350 MHz. Le débit réseau théorique en interne est de 1.25 MO/s (méga octets par seconde), alors qu’en pratique ce débit est inférieur à 700 KO/s (Kilo Octets par seconde).

Suivant la méthode de compression d’image utilisée et suivant la dynamique de l’image (peu ou beaucoup de mouvements), les temps serveur (T_s) et réponse du client (T_r) ont été calculés sur une série de dix premières images vidéo (comptée à partir de la deuxième image car la première ne subit pas de pré-traitement) envoyées au site client.

Dans ce type de connexion (locale), nous constatons que le fait d'utiliser des images avec peu ou beaucoup de mouvements, avec des méthodes de compression différentes, ne présentent pas un écart important sur le temps total T_t (Éq. (5.3)). En effet, pour des images brutes (sans compression), le temps total moyen est de 576 ms (0.104 sec + 0.472 sec, voir la figure 5.22-(A)).

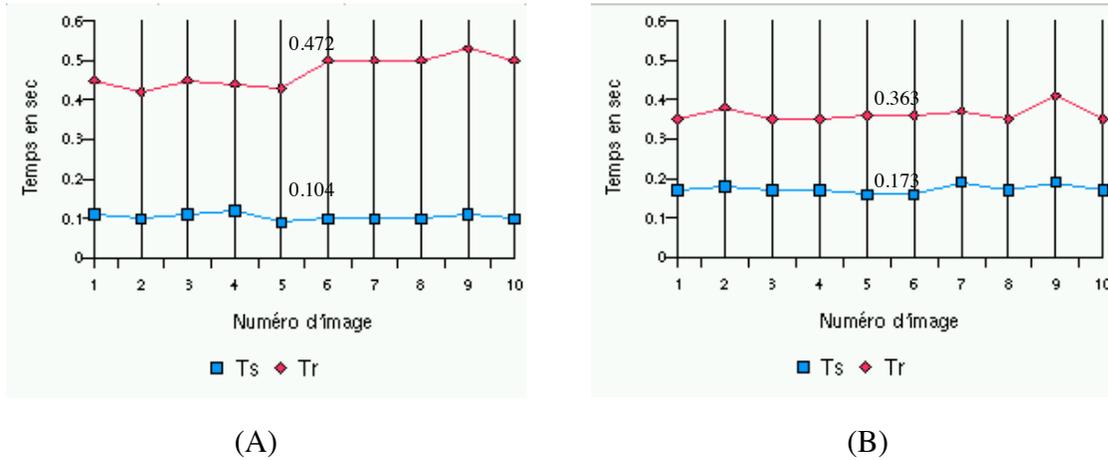


FIG. 5.22 – Evaluation des temps serveur et réponse du client lorsqu’il y a peu de mouvements dans l’image. (A): Pour des images brutes, (B): Pour des images compressées avec Gzip.

Pour des images avec peu de mouvements, ce temps avec une compression “Gzip” est de 536 ms (0.173 sec + 0.363 sec voir la figure 5.22-(B)) alors qu’il est de 446 ms (0.153 sec + 0.293 sec, voir la figure 5.23-(A)) lorsque “Gzip” est précédée d’un pré-traitement. Par contre ce temps est de 591 ms (0.138 sec + 0.453 sec, dans la figure 5.23-(B)) lorsque la méthode de Huffman précédée d’un pré-traitement est utilisée sachant qu’elle fournit un meilleur taux de compression (environ 71% de la taille de l’image brute).

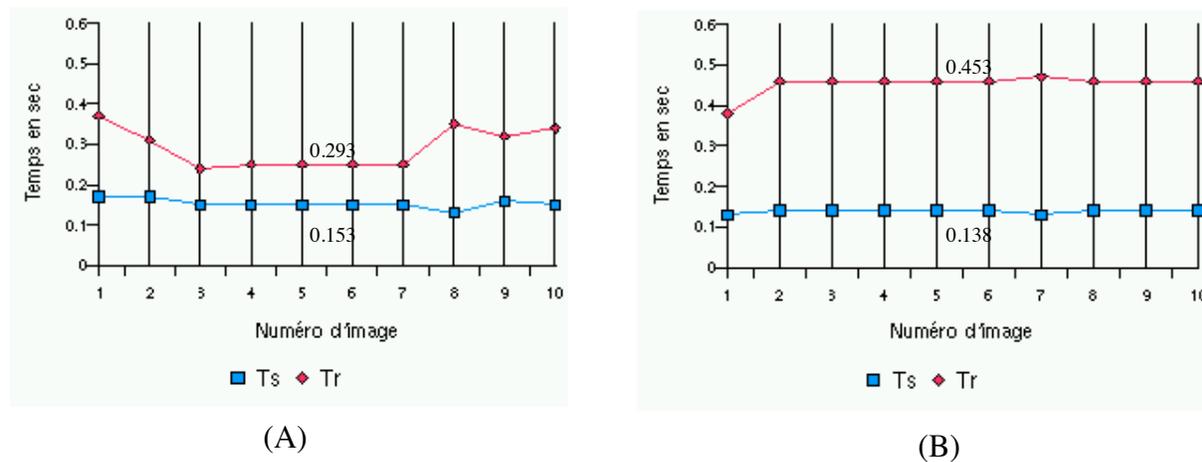


FIG. 5.23 – Evaluation des temps serveur et réponse du client lorsqu’il y a peu de mouvements dans l’image. (A): Pour des images prétraitées ensuite compressées en Gzip, (B): Pour des images prétraitées en compressées avec Huffman.

Pour des images avec beaucoup de mouvements, le temps total moyen est de 535 ms pour une compression “Gzip” (figure 5.24-(A)) et vaut 531 ms (figure 5.24-(B)) lorsque la compression “Gzip” est précédée d’un pré-traitement.

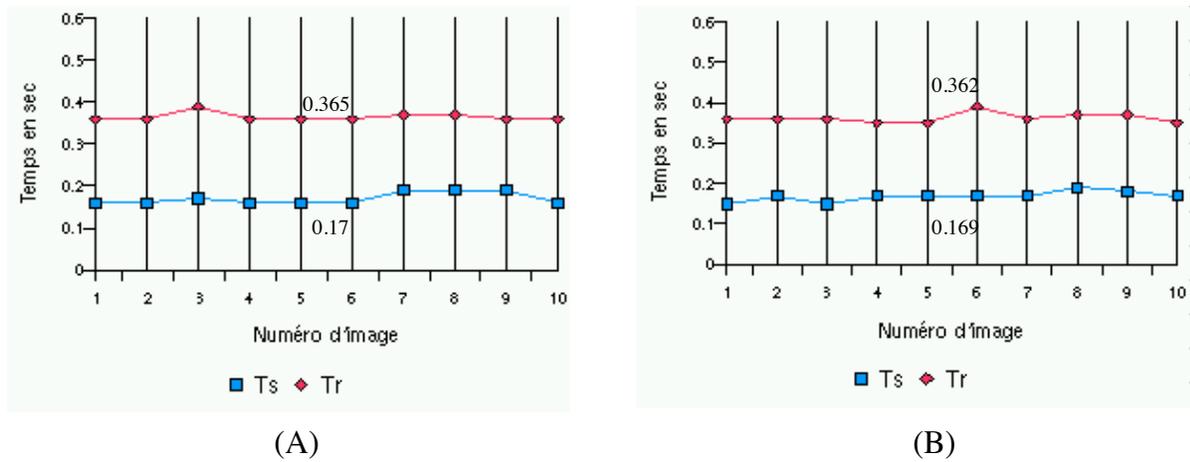


FIG. 5.24 – Evaluation des temps serveur et réponse du client lorsqu’il y a beaucoup de mouvements dans l’image. (A) : Pour des images compressées avec Gzip, (B) : Pour des images prétraitées et ensuite compressées avec Gzip.

Dans cette expérimentation, les faibles variations du temps total moyen, quelque soit la méthode de compression d’image, est une conséquence directe du fait que le site opérateur n’est pas éloigné du site esclave. En effet, en connexion locale, les délais de communications sont très faibles malgré leurs instabilités (par exemple ce délai augmente brusquement lors de l’expérimentation avec la méthode de Huffman précédée d’un pré-traitement représenté par la figure 5.23-(B)).

Expérimentalement, les images qui résultent des déplacements réalisés par le robot, sont considérées comme étant des images présentant peu de mouvements. Dans la majorité des cas, c’est la méthode de Huffman précédée d’un pré-traitement qui est utilisée.

Nous présentons maintenant, les résultats obtenus lorsque le site opérateur se trouve à moyenne distance.

5.3.1.2 Connexion moyenne distance

De nombreux opérateurs en Europe ont utilisé le système ARITI et réalisé des tâches. Nous présentons ci-dessous, quelques résultats de connexion lorsque le site opérateur se trouve en France, en Italie, en Espagne et au Royaume-Uni.

La figure 5.25, montre le résultat d’une connexion d’un opérateur se trouvant à “Toulouse”. Le temps moyen serveur est de 127 ms et le temps moyen de réception d’une nouvelle demande du client est de 2.49 sec, ce qui donne un temps total (T_t) moyen de 2.617 sec.

Dans la figure 5.26, l’opérateur se trouve à l’INRETS (Institut National de Recherche sur les Transports et leurs Sécurité) et le temps total moyen est de 3.178 sec (0.108 sec + 3.07 sec)

Un autre type de connexion est celui où le site opérateur est relié à Internet par un modem, il s’agit d’une connexion dont le débit réel est inférieur à 6 KO/s. Le temps total moyen est de 5.312 sec (0.112 sec + 5.2 sec voir la figure 5.27).

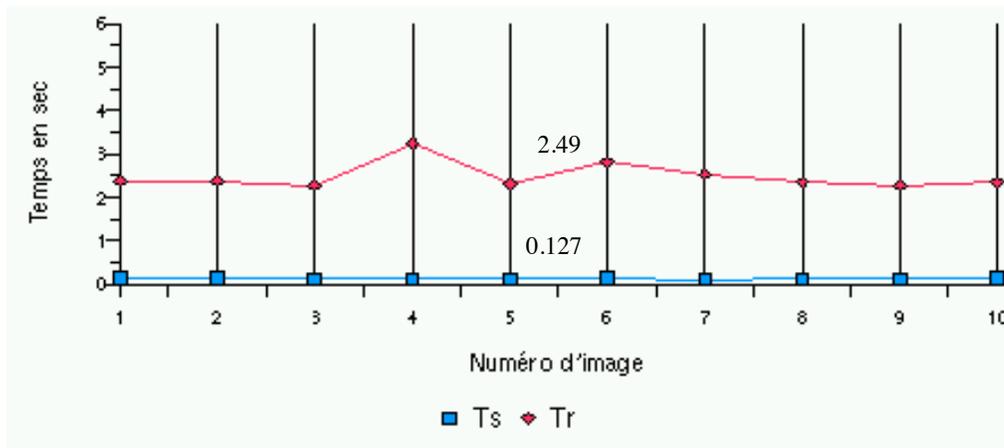


FIG. 5.25 – Evaluation des temps serveur et réponse du client à moyenne distance. L'adresse Internet du site client est "Toulouse-17-133.abo.wanadoo.fr".

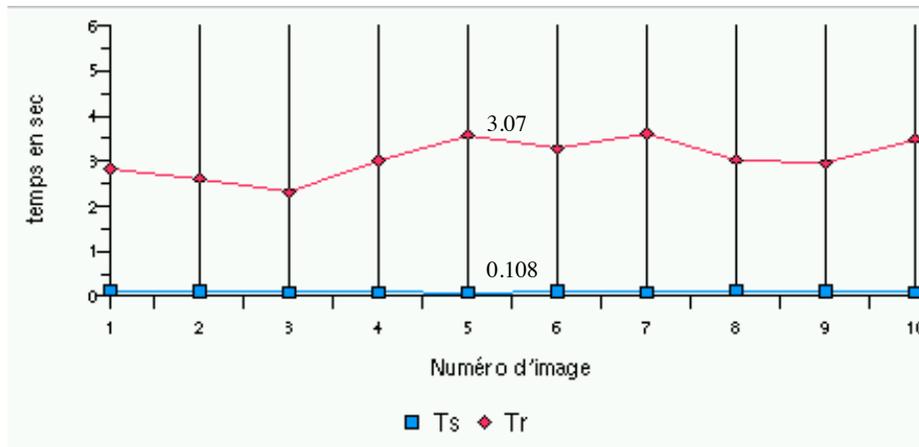


FIG. 5.26 – Evaluation des temps serveur et réponse du client à moyenne distance. L'adresse Internet du site client est "pc-41-13.inrets.fr".

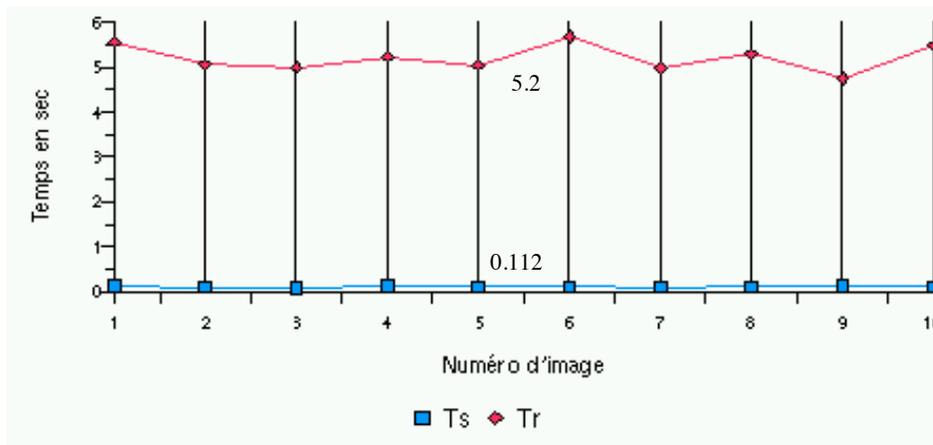


FIG. 5.27 – Evaluation des temps serveur et réponse du client à moyenne distance. L'adresse du site client est "dyn-213-36-102-192.ppp.libertysurf.fr".

Une connexion depuis l'Italie a permis d'avoir un temps total moyen intéressant de l'ordre de 1.085 sec, avec $T_r = 0.95$ sec et $T_s = 0.135$ sec (figure 5.28). Dans ce cas, le débit du côté du site client est de 1 GB/s (Giga Bits par seconde).

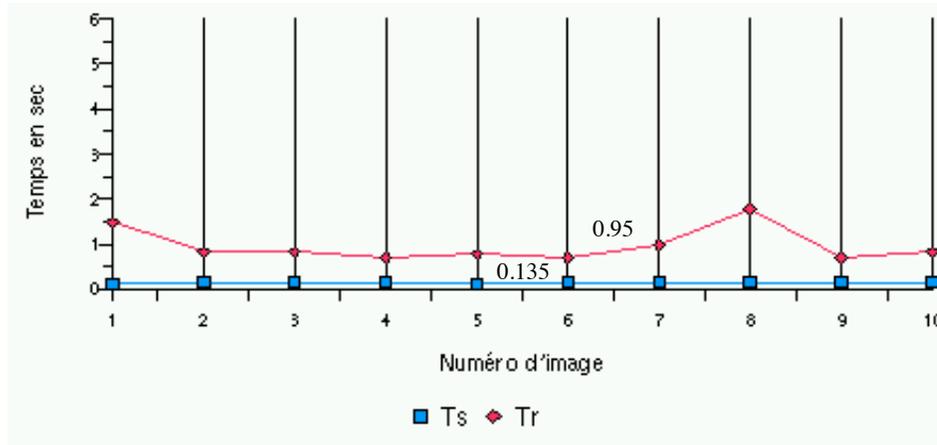


FIG. 5.28 – *Evaluation des temps serveur et réponse du client à moyenne distance. L'adresse Internet du site client est "pc.area.ba.cnr.it".*

Il est évident que le temps total T_t dépend essentiellement de la bande passante des deux sites maître et esclave, et aussi suivant le nombre de personnes connectées au réseau Internet à un instant donné. Nous constatons aussi que le temps serveur (T_s) moyen subit une faible variation (de 108 ms à 135 ms suivant les expérimentations présentées par les figures 5.25, 5.26, 5.27 et 5.28).

Dans le tableau 5.3, nous présentons les résultats pour d'autres connexions réalisées à moyenne distance. La première ligne représente les résultats de connexion depuis l'IRISA¹ de Rennes (avec une bande passante de 1 GB/s). La deuxième et la troisième ligne, pour une connexion de l'Espagne et la dernière du Royaume-Uni.

Les résultats représentés dans ce tableau sont obtenus sur une série d'envoi de 30 images vidéo, avec

- MT_s : Moyenne du Temps serveur.
- MT_r : Moyenne du Temps de réponse du client (réception d'une nouvelle demande du client)
- TMI : Taille Moyenne de l'image vidéo
- HC : Heure de Connexion (heure locale)

Adresse du site distant	MT_s (ms)	MT_r (sec)	TMI (KO)	HC
castor.irisa.fr	101.13	1.10	14.696	17:08:03
usuario1-36-131-230.dialup.uni2.es	115.09	4.69	15.366	15:51:13
usuario1-36-142-66.dialup.uni2.es	114.40	7.18	16.570	11:30:25
user142008.dial.netline.net.uk	109.90	4.72	14.525	16:35:49

TAB. 5.3 – *Quelques résultats de connexion à moyenne distance*

Nous présentons dans ce qui suit, les résultats de connexion lorsque le site opérateur se trouve à une longue distance.

1. Institut de Recherche en Informatique et Systèmes Aléatoires

5.3.1.3 Connexion longue distance

La téléopération du robot à aussi été réalisée par des opérateurs se trouvant dans d'autres continents, tels que l'Amérique, l'Asie et l'Océanie. Nous présentons ici quelques résultats de connexion depuis "San Francisco", "Washington", "Canada", "Brésil", "Argentine".

La figure 5.29, montre les résultats de connexion d'un opérateur se trouvant à "San Francisco". Le temps total T_t moyen est de 5.9 sec (0.12 sec + 5.78 sec)

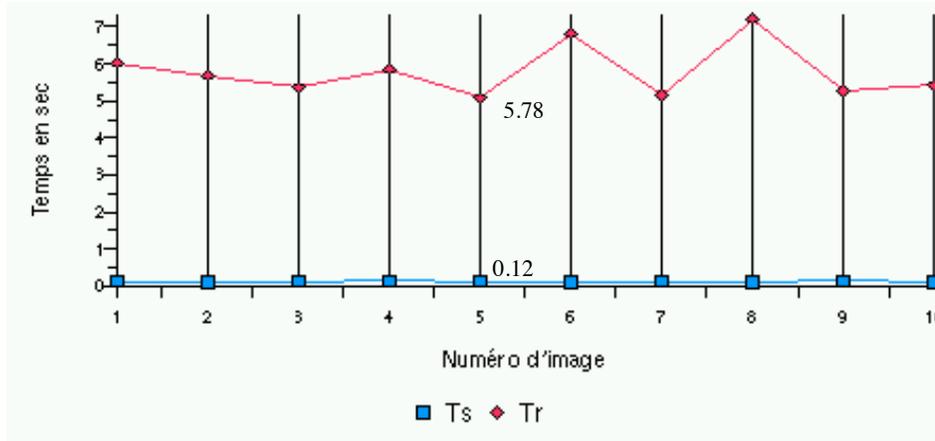


FIG. 5.29 – *Évaluation des temps serveur et réponse du client à longue distance. L'adresse du site client est "1Cust161.tnt1.san-francisco3.ca.da.uu.net"*

Pour une connexion depuis l'école polytechnique de Montréal "Canada", Le temps total moyen est de 5.624 sec (0.144 sec + 5.48 sec voir la figure 5.30).

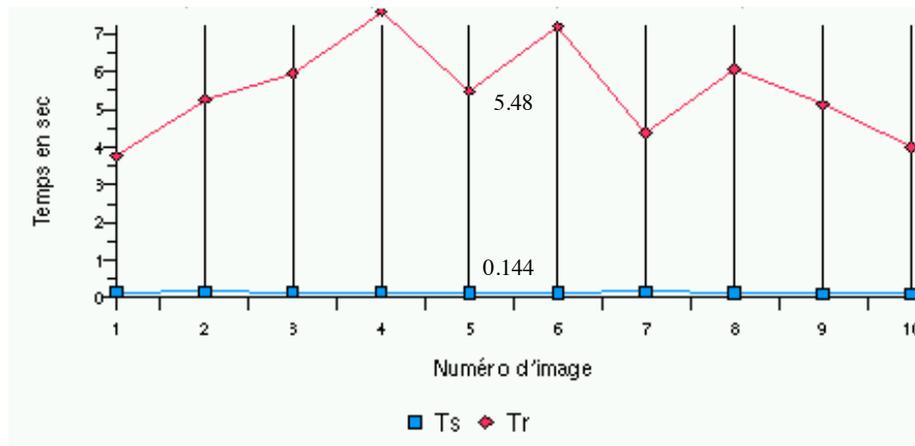


FIG. 5.30 – *Évaluation des temps serveur et réponse du client à longue distance. L'adresse Internet du site client est "internet-pub2.biblio.polymtl.ca"*

D'autres résultats de connexion à longue distance, sont représentés dans le tableau 5.4. La première ligne l'opérateur se trouve à "Washington", la deuxième et la troisième ligne depuis le "Brésil". Dans les deux dernières lignes, les opérateurs n'ont pas contrôlé le robot réel (ils n'ont pas demandé le "Login" et le "Pin code" pour avoir le contrôle du robot réel), mais ils ont reçu des images vidéo et utilisé le robot virtuel (ils ont réalisé des

Adresse du site distant	MT_s (ms)	MT_r (sec)	TMI (KO)	HC
Washington ^a	111.78	1.46	12.584	18:25:32
host052199.arnet.net.ar	109.26	5.69	14.348	16:54:20
line181.comsat.net.ar	119.41	7.45	16.581	19:08:27
maverick.furb.rct-sc.br	107.94	5.37	15.570	13:30:25
nc2608-50.cudenver.edu	111.78	1.27	13.696	20:08:03
141-211-31-173.bus.umich.edu	114.92	1.42	14.356	21:01:13

TAB. 5.4 – Quelques résultats de connexion à longue distance

^a dialup-209.244.215.139.Washington2.Level3.net

tâches seulement en virtuel).

avec MT_s , MT_r et TMI une moyenne sur 30 images vidéo envoyées.

Nous allons maintenant évaluer les performances de l'architecture client/serveur de ARITI, lorsque plusieurs opérateurs l'utilisent en même temps.

5.3.2 Connexion avec plusieurs opérateurs avec retour prédictif distribué

Dans cette section nous présentons quelques résultats lorsque par exemple un opérateur (appelé maître) contrôle le robot réel via le robot virtuel et les autres opérateurs (superviseurs), supervisent en virtuel et en réel les actions réalisées par l'opérateur maître. Un superviseur peut redevenir maître en prenant à nouveau le contrôle du robot réel. En effet, un seul opérateur peut avoir le contrôle du robot réel à un instant donné.

5.3.2.1 Connexion locale

Une expérimentation a été réalisée en locale au sein même de notre laboratoire Systèmes Complexes du CEMIF (Centre d'Etude de Mécanique d'Îles de France). Cinq opérateurs se sont connectés au système ARITI dans le but d'un télétravail coopératif (une mission partagée par cinq opérateurs).

Les machines maîtres sont des Pentiums, dont la vitesse du processeur est de 350 Mhz pour le maître et 500 Mhz pour les autres. Pour cette expérience, les connexions au système ARITI sont faites d'une manière séquentielle. En effet, le deuxième opérateur, attend environ 1 mn après la connexion du premier, etc. Ce temps d'attente nous l'avons imposé afin de pouvoir récupérer suffisamment de données nécessaires pour évaluer les performances du client/serveur en fonction du nombre de clients connectés au système. Nous présentons ici essentiellement l'expérimentation lorsque le premier opérateur contrôle le robot réel et les quatre autres supervisent en virtuel et en réel les actions réalisées par cet opérateur maître.

Nous avons mesuré les moyennes des temps, serveur et réponse du client, sur une série de 10 images vidéo envoyées. Cette évaluation est faite dans un premier temps avec un

seul opérateur connecté, ensuite avec 2, 3, 4 et 5 opérateurs simultanément.
avec :

- MTs désigne la Moyenne du Temps serveur en fonction du nombre de connexions simultanées.
- MTr désigne la Moyenne du Temps de réponse du client (ou reprise du serveur) en fonction du nombre de connexions simultanées.

Des résultats avec les différentes méthodes de compression d'images sont représentés dans les figure 5.31, 5.32 pour des images avec peu de mouvements et la figure 5.33 pour des images avec beaucoup de mouvements.

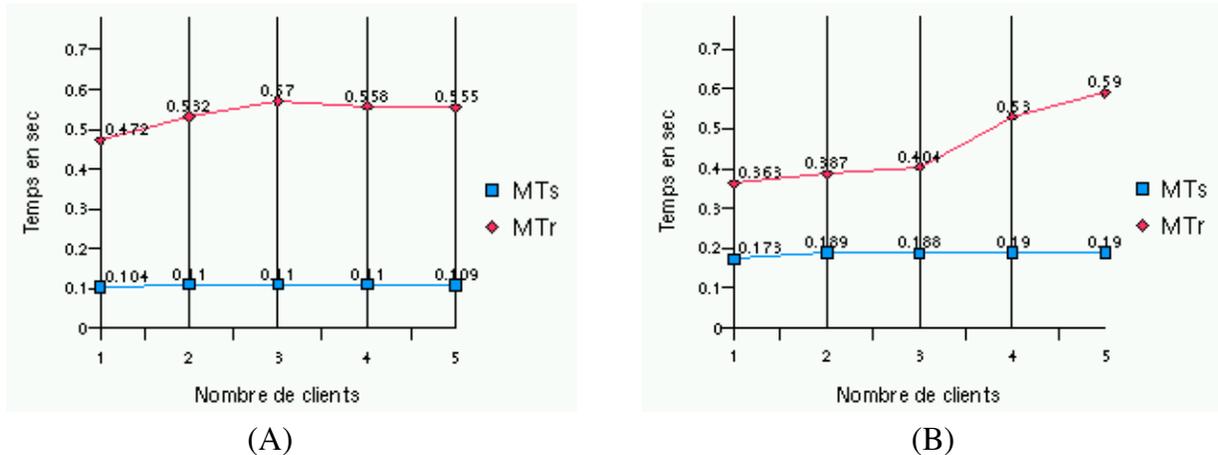


FIG. 5.31 – Les temps moyens, serveur et reprise du serveur, pour une connexion de 1 à 5 clients lorsqu'il y a peu de mouvements dans l'image. (A) : Pour des images brutes, (B) : Pour des images compressées avec Gzip.

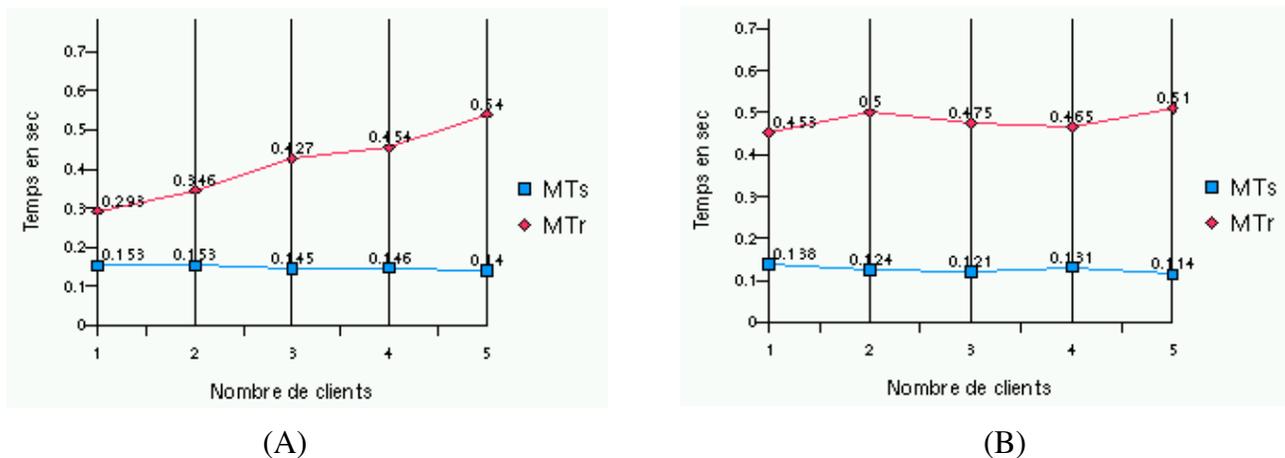


FIG. 5.32 – Les temps moyens, serveur et reprise du serveur, pour une connexion de 1 à 5 clients lorsqu'il y a peu de mouvements dans l'image. (A) : Pour des images prétraitées ensuite compressées en Gzip, (B) : Pour des images prétraitées et compressées avec Huffman.

Ces expérimentations montrent que le serveur d'image se comporte de la même façon pour une ou plusieurs connexions simultanées. En effet, pour des images brutes la moyenne du temps serveur (MTs) vaut 104 ms pour une seule connexion et 110 ms pour 2, 3, 4 et 5 connexions simultanées (figure 5.31-(A)). Pour des images avec peu de mouvement

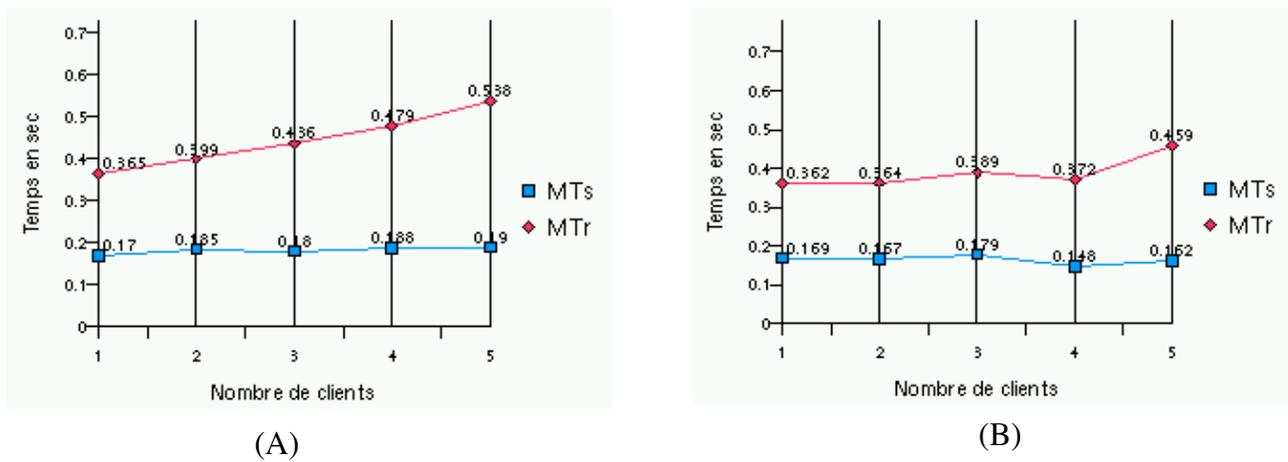


FIG. 5.33 – Les temps serveur, et reprise du serveur, pour une connexion de 1 à 5 clients lorsqu’il y a beaucoup de mouvements dans l’image. (A) : Pour des images compressées avec Gzip, (B) : Pour des images prétraitées et ensuite compressées avec Gzip.

et compressées en “Gzip” ce temps varie de 173 ms pour une connexion à 190 ms pour 5 connexions (5.31-(B)).

Par contre, cette variation est plus intéressante lorsque les images vidéo ont subi un pré-traitement avant compression. Nous constatons que ce temps passe de 153 ms pour un client à 140 ms pour 5 clients lorsque la méthode “Gzip” précédée d’un pré-traitement est utilisée (figure 5.32-(A)). Cette diminution de la moyenne du temps serveur est aussi plus importante lorsque la méthode de Huffman précédée d’un pré-traitement est utilisée, elle varie entre 138 ms pour un client et 114 ms pour 5 clients (figure 5.32-(B)).

Lorsque l’image subie beaucoup de mouvements, le temps serveur varie entre 170 ms pour un seul client et 190 ms pour 5 clients avec une compression “Gzip” (figure 5.33-(A)).

Nous présentons maintenant les résultats obtenus lors de la télésupervision avec retour prédictif distribué. Pour cela, nous avons mesuré les temps moyens nécessaires pour la mise à jour des images réelles et des environnements réels, en fonction du nombre de clients connectés au système ARITI (voir le tableau 5.5).

Soit :

- MR_{ER} : Moyenne du temps de Réponse du client pour la mise à jours des images Réelles
- MR_{EV} : Moyenne du temps de Réponse du client pour la mise à jours de l’environnement virtuel.

Ces résultats, montrent le gain de temps que peut fournir un retour prédictif lors d’un télétravail ou encore lors d’un télétravail coopératif. En effet, les actions réalisées par l’opérateur maître avec le robot virtuel dans son environnement virtuel, sont récupérées par le serveur de commande et envoyées aux différents superviseurs. Pour la mise à jours de l’environnement virtuel du côté de chaque client, le serveur de commande informe chaque client superviseur, de l’état actuel du robot et des tâches en cours. Dans cette expérimentation, les données nécessaires sont les coordonnées des quatre moteurs du robot (24 octets) et une chaîne de caractères représentant le codage de l’état des tâches réalisées (9 octets). Il est cependant évident que la taille totale des données nécessaires pour la mise à jour des environnements virtuels est très petite par rapport à la taille des

Nombre d'opérateurs	MR_ER (ms)	MR_EV (ms)
1	591	11.0266
2	624	24.7395
3	596	44.8783
4	596	71.7134
5	624	98.9045

TAB. 5.5 – La moyenne des temps pour la mise à jour des images réelles et des environnements virtuels lors d'un télétravail coopératif avec plusieurs opérateurs. Dans ce cas, les opérateurs se trouvent dans le même laboratoire.

images à envoyer pour la mise à jour de l'affichage réel (33 octets \ll 12584 octets dans le meilleur des cas).

Par contre, le temps moyen pour la mise à jour des environnements virtuels augmente en fonction du nombre d'opérateurs utilisant le système ARITI simultanément. Cette augmentation se justifie par le fait que le serveur envoie les données (pour la mise à jour de l'environnement virtuel) dans l'ordre de passage en mode "télé supervision". La figure 5.34 montre les deux courbes représentant les temps moyen de mise à jour des images réelles et des environnements virtuels.

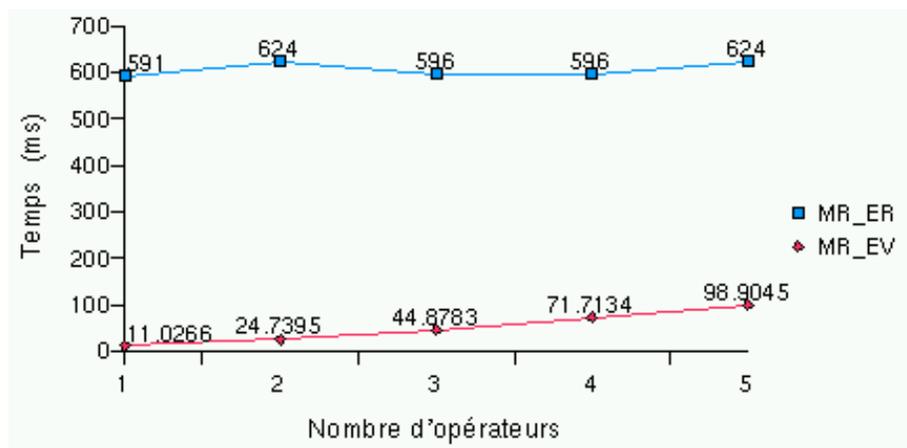


FIG. 5.34 – Temps de mise à jour des images réelles et des environnements virtuels en fonction du nombre d'opérateur utilisant le système ARITI en connexion locale.

Nous allons maintenant voir les résultats de connexion à moyenne et longue distance.

5.3.2.2 Connexion moyenne et longue distance

Deux expérimentations ont été réalisées en faisant participer seulement deux opérateurs à chaque fois.

La première expérimentation est réalisée avec deux opérateurs se trouvant en Espagne, le premier opérateur se connecte et prend la commande du robot réel, et le deuxième se connecte 2 mn plus tard et utilise le mode "télé supervision". De la même façon que précédemment, nous avons mesuré les temps moyen pour la mise à jour des images réelles

et des environnements virtuels, voir le tableau 5.6.

La deuxième expérimentation fait participer deux autres opérateurs, le premier se trouve en Italie et le second à Montréal (Canada). Dans les mêmes conditions d'expérimentation que précédemment, le premier opérateur prend le contrôle du robot réel et le second supervise en virtuel et en réel. Les résultats de cette expérience sont donnés dans le tableau 5.7.

Adresse ^a	ordre de connexion	MR_{ER} (s)	MR_{EV} (s)
usuario1-36-131-230.dialup.uni2.es	1	4.80	0.39
usuario1-36-142-66.dialup.uni2.es	2	5.14	0.48

TAB. 5.6 – La moyenne des temps pour la mise à jour des images réelles et des environnements virtuels lors d'un travail coopératif avec deux opérateurs distants. Dans ce cas, les opérateurs se trouvent en Espagne.

^aNom Internet du site opérateur distant

Adresse	ordre de connexion	MR_{ER} (s)	MR_{EV} (s)
pc.area.ba.cnr.it	1	1.15	0.14
internet-pub2.biblio.polymtl.ca	2	5.49	0.82

TAB. 5.7 – La moyenne des temps pour la mise à jour des images réelles et des environnements virtuels lors d'un travail coopératif avec deux opérateurs distants. Dans ce cas, le premier se trouve en Italie et le deuxième au Canada.

5.4 Bilan

Les expérimentations présentées dans ce chapitre, nous ont permis d'évaluer notre système de télétravail ARITI. En effet, dans la première section, nous avons montré que l'interface de ARITI permet un apprentissage des tâches et que l'utilisation des guides virtuels fournit une sécurité pour le robot et son environnement. De même, l'utilisation des guides virtuels attractifs permet d'améliorer les performances des opérateurs vis à vis des tâches de téléopération, en fournissant une meilleure précision d'approche des objets à manipuler (inférieure au mm) et un meilleur temps de réalisation des missions (7.7 sec pour une tâche d'atteinte d'une cible avec un guide virtuel attractif contre 22.21 sec sans guide virtuel).

Par ailleurs, l'évaluation des méthodes de compression d'images étudiées, nous a permis de déterminer un algorithme de choix automatique de la compression à réaliser en fonction de la dynamique de l'image. En effet, le choix s'est porté sur un "pré-traitement" suivi de la méthode de "Huffman" lorsque les images vidéo présentent peu de mouvements (avec un taux de compression de 71%). Par contre, lorsque les images présentent beaucoup de mouvements, le "pré-traitement" suivi de la méthode "Gzip" est choisie pour compresser les images vidéo (avec un taux de compression de 44%).

D'autres expérimentations ont été réalisées pour évaluer les performances du système ARITI et essentiellement l'architecture client/serveur. Nous avons montré la stabilité du

serveur vis à vis de nombreuses connexions, locales, moyennes et grandes distances. En effet, la téléopération en utilisant le système ARITI, essentiellement à moyenne et longue distance, est devenue possible grâce au retour prédictif permettant aux opérateurs d'une part, de contrôler le robot réel via sa représentation virtuelle et d'autre part, de superviser l'exécution des tâches en réalité augmentée. Bien que les délais de transmissions d'images vidéo en particulier, dépendent complètement de la distance et de la bande passante des deux sites et essentiellement des sites clients. Par exemple pour une téléopération en local avec un débit réseau inférieur à 700 KO/s, nous obtenons un peu plus de deux images par seconde. Pour une téléopération depuis "Toulouse" le délai moyen est de 2.5 sec pour une image, depuis l'IRISA de Rennes et l'Italie avec un débit de 1 Gbits/s, ce délai est autour d'une seconde (une image par seconde).

Pour une téléopération à longue distance, le même phénomène a été constaté, par exemple depuis "San-francisco", "Montréal" et "Argentine", ce délai est entre 5 et 7 secondes alors que depuis "Washington" le retour d'une images vidéo se fait toutes les 1.5 sec en moyenne.

D'autre part, la conception et la mise en œuvre d'une architecture de communication client/serveur, supportant de multiples connexions simultanées (jusqu'à 255 opérateurs) à permis à plusieurs clients distants de télétravailler en coopération avec un retour prédictif distribué. En effet, un travail coopératif avec deux opérateurs distants (dans la première expérience les opérateurs se trouvent en "Espagne" et dans la seconde, un se trouve en "Italie" et l'autre au "Canada") à été réalisé. Le premier opérateur a pris le contrôle des robots virtuel et réel et le second supervise toutes les actions réalisées par le premier, en virtuel et en réel et ensuite reprend à son tour, le contrôle des robots. Ces expérimentations ont montré, l'avantage d'un retour prédictif distribué (mise à jour des environnements virtuels des sites clients distants) lors d'un travail coopératif en présence de délais de transmission. En effet, les temps de mise à jour des environnements virtuels du côté des clients superviseurs est de 480 ms pour l'Espagne et de 820 ms pour le Canada, contre respectivement, 5.14 sec et 5.49 sec pour les retours d'images vidéo.

Conclusions

Tout au long de cette thèse, nous avons essayé d'atteindre nos objectifs. En effet, des réponses aux problématiques posées ont été apportées et présentent ainsi notre contribution, à savoir :

- ❶ Les outils logiciels et matériels nécessaires pour la mise en œuvre d'un système de télétravail ont été proposés et utilisés pour la réalisation du système ARITI (ce problème est traité dans le chapitre 3).
- ❷ Les nouveaux concepts et les solutions déjà apportées aux problèmes liés à la télérobotique, ont été intégrés sur une plate-forme universelle et à moindre coût, puisque l'interface du système ARITI a été développée sur un simple PC (ce problème est traité dans les chapitres 3 et 4).
- ❸ L'utilisation d'un système de télétravail est devenue souple et fournit un bon apprentissage pour l'opérateur. Les expérimentations réalisées avec le système ARITI l'ont montré dans le chapitre 5.
- ❹ Les types d'assistances nécessaires à l'opérateur afin d'améliorer ses performances, sont présentés dans le chapitre 3, il s'agit d'assistance à la perception de l'environnement, à la commande du robot et à la supervision des tâches. Pour l'assistance à la commande du robot, des guides virtuels sont utilisés, ces derniers sont présentés dans les chapitres 2 et 4, et leurs performances sont discutées dans le chapitre 5.
- ❺ Une architecture client/serveur supportant un travail coopératif est réalisée et est présentée dans le chapitre 3 et ses performances sont discutées dans le chapitre 5.
- ❻ Quelques avantages que peuvent offrir les systèmes de télétravail en particulier en utilisant le réseau Internet, ont été fournis dans le chapitre 4.

Le premier chapitre nous a permis d'un côté, de présenter les différents aspects du télétravail ainsi que les méthodes et les tendances technologiques qui contribuent énormément à son évolution. D'un autre côté, il nous a permis d'analyser les systèmes de téléopération et de téléprogrammation de robots existants, les méthodes et les techniques utilisées pour pallier les problèmes liés à la télérobotique en général et enfin d'analyser les récents travaux concernant le contrôle de systèmes robotique via Internet. A l'issue de ce chapitre, des problématiques ont été dégagées et ont fait objet d'études et de discussions dans les autres chapitres.

Dans le second chapitre, les notions de mécanisme et de guide virtuel indispensables à un système de réalité augmentée basé sur un retour prédictif ont été présentées. Nous avons aussi présenté les méthodes analytiques et géométriques de création, de représentation et de manipulation des objets virtuels en général et en particulier des guides virtuels. Nous avons ensuite proposé un nouveau formalisme pour les guides virtuels afin de les rendre paramétrables, évolutifs et utilisables comme de véritables outils d'assistance à la téléopération via Internet.

Dans le troisième chapitre, nous avons présenté les méthodes et les outils nécessaires pour un contrôle en réalité augmentée à savoir, la modélisation de l'environnement et la calibration de la caméra et du robot. Nous avons aussi présenté le système expérimental de télétravail baptisé ARITI. L'interface de ce système est implantée en langage JAVA dans le but de la rendre portable et indépendante des systèmes d'exploitation sur lesquels elle va s'exécuter. Nous avons insisté sur des détails de l'architecture client/serveur réalisée afin de montrer d'un côté, sa complexité et de l'autre côté, la faisabilité conceptuelle et technique d'une architecture destinée à supporter un télétravail d'une part et un travail coopératif d'autre part.

Par ailleurs, nous avons présenté les trois modes de télétravail utilisés qui sont la téléopération, la téléprogrammation de tâches et la télécoopération/télé supervision. Ce dernier mode utilise un retour prédictif distribué, afin de permettre aux différents opérateurs de percevoir à la fois, les images vidéo de l'environnement réel distant et la mise à jour 3D de l'environnement virtuel au niveau de chaque client superviseur.

Dans le quatrième chapitre, nous avons présenté quelques applications pouvant bénéficier d'un système de télétravail via Internet. Nous avons cependant souligné la nécessité d'avoir d'un côté, une interface administrateur pour la maintenance et la mise à jour d'un tel système et de l'autre côté une interface destinée aux utilisateurs du système.

Nous avons décrit certaines tâches réalisées suivant le mode de télétravail choisi, d'un côté, les tâches de téléopération (saisie/dépôt d'objets, etc.) où l'opérateur s'il le désire peut être assisté par des guides virtuels. D'un autre côté, l'opérateur peut télétravailler à un haut niveau (superviseur). Pour cela, il spécifie seulement la tâche à faire exécuter par les robots virtuel et réel simultanément. Enfin, si une mission nécessite l'intervention de plusieurs opérateurs simultanément dans le but d'un travail coopératif par exemple, un opérateur prend le contrôle du robot réel pendant que les autres supervisent en virtuel et en réel les actions de celui-ci. Dans ce dernier cas, le contrôle des robots virtuel et réel est partagé entre les différents opérateurs.

Nous avons montré la flexibilité et l'ouverture du système ARITI pour d'autres applications, en utilisant un robot mobile. En effet, une application sur un robot mobile a été réalisée en s'inspirant de l'existant et en rajoutant essentiellement le module contenant les nouveaux modèles virtuels ainsi que le module de client/serveur commande. Nous avons cité d'autres domaines d'application où un système de télétravail ou de travail coopératif via Internet peut être intéressant. La liste fournie n'est bien entendu pas exhaustive.

Dans le dernier chapitre, nous avons réalisé des expérimentations afin d'évaluer notre système de télétravail ARITI. En effet, nous avons montré que l'interface de ARITI permet un apprentissage des tâches et que l'utilisation des guides virtuels fournit une sécurité pour le robot et son environnement. Nous avons montré que l'utilisation des guides virtuels attractifs permet d'améliorer les performances des opérateurs vis à vis des tâches de téléopération, en fournissant une meilleure précision d'approche des objets à manipuler et un meilleur temps de réalisation des missions (7.7 sec pour une tâche d'atteinte d'une cible avec un guide virtuel attractif contre 22.21 sec sans guide virtuel).

Par ailleurs, l'évaluation des méthodes de compression d'images étudiées, nous a permis de déterminer un algorithme de choix automatique de la compression à réaliser en fonction de la dynamique de l'image. En effet, le choix s'est porté sur un "pré-traitement" suivi de la méthode de "Huffman" lorsque les images vidéo présentent peu de mouvements (avec un taux de compression de 71%). Par contre, lorsque les images présentent beaucoup de

mouvements, la méthode “pré-traitement” suivie de “Gzip” est choisie pour compresser les images vidéo (avec un taux de compression de 44%).

Les autres expérimentations présentées dans ce dernier chapitre, ont permis d'évaluer les performances du système ARITI et essentiellement l'architecture client/serveur. Nous avons montré la stabilité du serveur vis à vis de nombreuses connexions, locales, moyennes et grandes distances. En effet, le télétravail et le travail coopératif en utilisant le système ARITI (essentiellement à moyenne et longue distance) sont devenus possibles grâce au retour prédictif distribué. Ce dernier, permet aux opérateurs d'un côté, de contrôler le robot réel via sa représentation virtuelle et d'un autre côté, de superviser l'exécution des tâches en réalité augmentée.

Un système expérimental de télétravail via Internet, baptisé ARITI (Augmented Reality Interface for Telerobotic applications via Internet) a été réalisé. Il est accessible sur Internet d'une part, depuis 1999 sur le site Web de notre laboratoire (<http://lsc.cemif.univ-evry.fr:8080/Projets/ARITI>) et d'autre part, depuis Février 2000 sur le site de la NASA (*Nasa Space Telerobotics program* au [http : //ranier.oact.hq.nasa.gov/telerobotics_page/-realrobots.html](http://ranier.oact.hq.nasa.gov/telerobotics_page/-realrobots.html)).

Le système ARITI est le premier système en France qui permet la téléopération en réalité augmentée via Internet.

Perspectives

Système de télétravail multimodal

Il est possible de rajouter au système ARITI des retours d'informations multimodales afin d'augmenter le degré de téléprésence de l'opérateur. Il s'agit d'informations Sonore, Tactile et Kinesthésique. En effet, l'ajout d'un microphone (connecté à une carte son du PC serveur) sur le site esclave ainsi que des capteurs de contact et de force au niveau de l'effecteur du robot, permettrait de récupérer via des processus serveurs dédiés, toutes ces informations vers les machines clientes (sites maîtres) où se trouvent les opérateurs. Cependant, des questions se posent d'une part, sur le choix du format des fichiers audio ainsi que les méthodes de compression à utiliser. D'autre part, sur les dispositifs à retour tactile et d'effort pouvant être utilisés sur un système de télétravail via Internet. Des joysticks à retour d'effort à moindre coût, existent et fonctionnent sous l'environnement Linux.

La figure 5.35 illustre cette multimodalité avec des transferts d'informations audio et haptique, avec un processus dédié pour chaque type d'information.

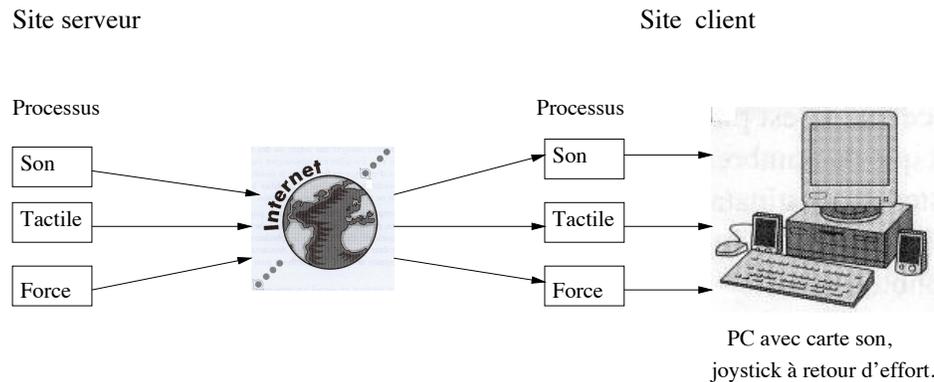


FIG. 5.35 – Illustration des transferts d'informations multimodales via Internet entre le site maître (client) et le site esclave (serveur).

Cette multimodalité impose la contrainte que la machine du client doit disposer du matériel adapté (organe à retour d'effort) pour réaliser une commande à retour d'effort du robot.

Interaction de ARITI avec d'autres systèmes de réalité augmentée et de réalité virtuelle

Il faut remarquer que dans l'état actuel du système ARITI, nous ne pouvons travailler qu'avec des environnements modélisés (robot, objets, etc.). En effet, pour permettre une modélisation interactive de l'environnement nous pouvons faire appel à d'autres systèmes dédiés sans pour autant reprendre les travaux et tous les efforts réalisés à ce sujet.

En effet, la communication en réseau avec ces systèmes permettrait l'échange des données afin de combler ce manque. De la même façon si nous désirons faire du traitement d'images afin de recalibrer un objet virtuel sur le réel nous n'allons pas le faire avec

le système ARITI, mais une communication avec le système MCIT (Mutimédia Control Interface in Teleoperation) existant sur une station Silicon Graphics (SG) au sein de notre laboratoire pourrait très bien apporter cette solution. En effet, du moment que le transfert de l'image vidéo de ARITI vers MCIT suffit pour que ce dernier traite l'image et ensuite envoie les informations nécessaires pour recalculer l'objet virtuel sur le réel (Mallem *et al.*, 1996) et (Moreau *et al.*, 1997).

La figure 5.36 illustre cette perspective.

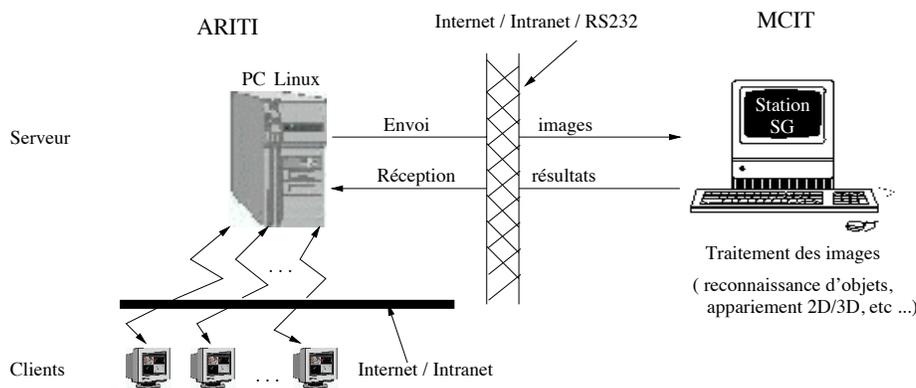


FIG. 5.36 – Illustration de la communication entre le système ARITI et MCIT pour faire du recalage d'objets virtuels sur les objets réels.

L'interopérabilité entre un système de vision artificielle (MCIT) et un système de télétravail (ARITI) permet d'étendre les possibilités de ce dernier.

Télécalibration de la caméra

Il est maintenant possible de calibrer la caméra vidéo à distance. En effet, la connaissance des coordonnées 3D de l'effecteur du robot à chaque instant (grâce au modèle géométrique direct du robot), permet à l'opérateur d'extraire les points 2D écran (à partir des images vidéo qu'il reçoit) correspondants aux points 3D de l'extrémité de l'effecteur du robot.

Cette télécalibration, peut aussi se faire automatiquement, à condition que l'extrémité de l'effecteur du robot soit suffisamment visible (contrastée par rapport à l'environnement etc.) afin que la détection du point 2D correspondant, puisse se faire par un traitement d'image. Bien entendu, cette télécalibration doit se faire par l'opérateur "administrateur", car elle nécessite la mise à jour du module de calibration de la caméra (régénérer un autre exécutable).

La télécalibration permet le changement du point de vue réel (télécommander la caméra pour faire un zoom par exemple, etc) et la mise à jour de la superposition du monde virtuel sur le réel, moyennant une intervention minimale de l'opérateur.

Téléopération de plusieurs robots en coopération

Afin de permettre à plusieurs opérateurs de contrôler plusieurs robots oeuvrant en coopération, une extension du système ARITI peut être réalisée. Cependant, il est nécessaire de s'intéresser aux problèmes de partage et de gestion des ressources communes (les objets et les outils virtuels communs, la gestion des points de vue virtuels et réels). Des questions se posent à savoir, comment doit se faire la communication entre les différentes entités virtuelles? Quelles sont les méthodes et les outils nécessaires pour permettre l'interopérabilité (Allongue et M.Soto, 1997) de ces différentes entités virtuelles?

La figure 5.37 illustre un exemple de deux opérateurs distants qui désirent réaliser une mission qui nécessite l'intervention de deux robots, chaque opérateur contrôle un robot et voit ce que fait l'autre en virtuel et en réel. Ici, les caractéristiques et les fonctionnalités de chaque robot sont gérées par un serveur dédié. L'extension du système ARITI ne servira que d'intermédiaire pour le contrôle des deux robots via Internet.

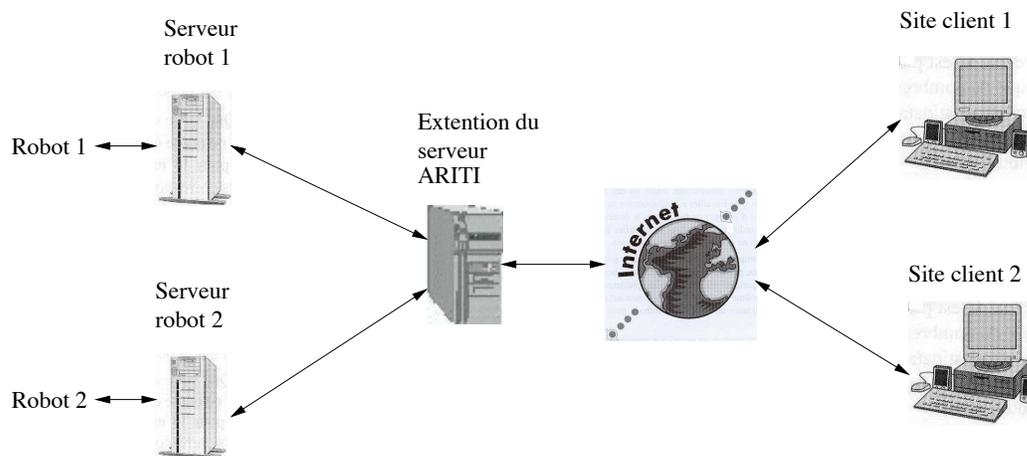


FIG. 5.37 – Illustration de la téléopération de deux robots par deux clients distants en utilisant une extension du système ARITI.

Langage de haut niveau pour la télémanipulation d'objets via Internet

Les méthodes étudiées et les outils utilisés pour réaliser le système de télétravail ARITI, peuvent très bien être utilisées et améliorées pour la conception d'un langage de haut niveau pour la télémanipulation d'objets via Internet. En effet, les méthodes de désignation et de manipulation sur écran d'objets virtuels (décrites dans le chapitre 2), ainsi que l'existence des outils virtuels paramétrables et portables (comme les guides virtuels par exemple) peuvent être utilisés pour créer un langage de télémanipulation de haut niveau.

Par exemple l'opérateur peut saisir et déposer un objet virtuel avec la souris (2D ou encore 3D ou 6D), et les robots virtuel et réel exécutent la tâche sans que l'opérateur agisse sur le robot virtuel. Ceci nécessite l'utilisation des guides virtuels complètement autonomes qui seront associés à chaque sous tâche (saisie, dépôt, etc.). Pour cela, une analyse sur les interactions de l'opérateur avec l'environnement virtuel est nécessaire (associer un sens aux différentes actions de l'opérateur vis à vis de l'environnement virtuel),

ainsi qu'une grammaire prenant en compte des outils d'assistance qui vont provoquer l'auto-commande du robot.

Cette analyse est rendu possible grâce à une connaissance plus approfondie de l'environnement. En effet, il faut en plus du modèle géométrique utilisé actuellement, prendre en compte les modèles cinématiques, dynamiques et physiques (liaisons ou dépendances entre les objets de l'environnement).

Bibliographie

- Allongue, S. et M.Soto (1997). Paradigmes sur les comportements dans le cadre de l'interopérabilité en réalité virtuelle. In *1ere conf. francophone de Modélisation et Simulation des systèmes de production et logistique MOSIM'97*.
- Andriot, C. *Automatique des Systèmes Téléopérés avec Retour d'Effort. Limitation des Performances*. Thèse de doctorat, Université Pierre et Marie Curie (1997).
- Backes, P. G., Tao, K. S., et Tharp, G. K. (1998). Mars pathfinder mission internet-based operations using wits. In *IEEE, International Conference on Robotics and Automation*, pages 284–291.
- Backes, P. G. (WWW). Pathfinder sojourner rover simulation web page. <http://mars.graham.com/wits/>.
- Bannon, L. et Schmidt, K. (1989). Cscw : Four characters in search of a context. In *In Proceedings of the European Community Conference on CSCW (EC-CSCW)*, pages 358–372, London.
- Bejczy, A. (1992). Teleoperation : The language of the human hand. In *The IEEE Int. Workshop on Robot and Human Communication, ROMAN'92*, pages 32–43, September 1-3, Tokyo, Japan.
- Bejczy, A. et Kim, W. (1990). Predictive displays and shared compliance control for time-delayed telemanipulation. In *IEEE Int. Workshop on Intelligent Robots and Systems, IROS'90*, pages 407–412, Tsuchiura, Japan, July.
- Bejczy, A., Kim, W., et Venema, S. (1990). The phantom robot: Predictive displays for teleoperation with time delay. In *IEEE Int. Conf. on Robotics and Automation*, pages 546–551.
- Binstock, A. et Rex, J. (1995). *Practical Algorithms for Programmers*. Reading, MA: Addison-Wesley.
- Blanc, C. et Schlick, C. (1996). *Ratioquadratics : an alternative model for superquadratics*.
- Bonneau, P. et Even, P. (1993). Man-machine cooperation for 3d objects pose estimation. In *IEEE SMC conf.*, volume 2, pages 294–299, Le Touquet, december.
- Brooks, T. (1990). Telerobotic response requirements. In *IEEE Int. Conf. on Systems Man and Cybernetics*, pages 113–120, Nice, France, May.
- Brooks, T. et Ince (1992). Operator vision aids for telerobotic assembly and servicing in space. In *IEEE Int. Conf. on Robotics and Automation, ICRA'92*, pages 886–891, Nice, France, May.
- Béréziat, A., Lagorce, J., et Turbé-Suetens, N. (2000). *Travail et activités à distance*. Editions d'Organisation.
- Burdea, G. (1993). *Virtual Reality systems and Applications*. Edison, NJ, USA.

- Burdea, G. et Coiffet, P. (1994). *Virtual Reality Technology*. John Wiley et Sons, Inc Eds, New York.
- Burdea, G., Zhuang, G., Roskos, J., et E. Silver, K. L. (1992). A portable dextrous master with force feedback. In *Presence*, volume 1, No. 1, pages 18–28.
- Cammoun, Detriche, Lauture, et Lesigne (1994). Telerobotics in the service of disabled persons. In *ORIA '94 de la téléprésence vers la réalité virtuelle, 5ème colloque international et convention d'affaires*, pages 249–254, Marseilles, Decembre.
- Caudell, T. P. (1994). Introduction to augmented reality. In *SPIE Proceedings: Telemanipulator and Telepresence Technologies*, volume 2351, pages 66–70.
- Causse, O. et Crowley, J. (1993). A man-machine interface for a mobile robot. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'93*, pages 1487–1494, Yokohama, Japan, July 26-30.
- Chekhar, Y. *Saisie et Traitement d'images télémétriques. Application à la Téléopération*. Doctorat de robotique, Université d'Evry Val d'Essonne (1994).
- Clement, G., Fournier, Gravez, P., et Morillon, J. (1988). Computer aided teleoperation: From arm to vehicle control. In *IEEE Int. Conf. on Robotics and Automation, ICRA '88*, volume 1, pages 590–592,, Philadelphia, PA, April 24-29.
- Coiffet, P. et Gravez, P. (1991). *Human-Robot Cooperation: Toward an Advanced Teleoperation Mode*. Tzafestas editor, Marcel Dekker Inc.
- Conway, L., Volz, R., et Walker, M. (1990). Teleautonomous systems: Projection and coordinating intelligent action at a distance. In *IEEE Tran. on Robotics and Automation*, volume 6, pages 146–158, April.
- Crespin, R. (1999). Surfaces implicites. <http://dept-info.labri.u-bordeaux.fr/~crespin/Implicite/index.html>.
- DeLarminat, P. (1993). *Automatique – commande des systèmes linéaires*. Edition Hermès, Paris.
- Devy, M., Garric, V., et Orteu, J. (1997). Camera calibration from multiple views of a 2d object, using a global non linear minimization method. In *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'97*, volume 3, pages 1583–1589, Grenoble, septembre.
- Feiner, S., MacIntyre, B., et al (1993). Windows on the world: 2d windows for 3d augmented reality. In *Proceedings of ACM Symposium on User Interface Software and Technology*, pages 145–155, Atlanta, GA, Association for Computing Machinery.
- Ferrell, W. R. (1965). Remote manipulation with transmission delay. In *IEEE Transactions on Human Factors in Electronics*, volume 6, pages 24–32, september.
- Ferrell, W. R. (1966). Delayed force feedback. In *IEEE Transactions on Human Factors in Electronics*, pages 445–449, octobre.
- Fiorini, A., Bejczy, A. K., et Schenker, S. (1992). Integrated interface for advanced teleoperation. In *IEEE SMC Conf*, pages 18–21, October.
- Foley, van Dam, Feiner, et Hughes (1990). *Computer Graphics - principles and practice*. Addison-Wesley publishing company.
- Fraisse, P. *Contribution à la Commande Robuste Position/Force des Robots Manipulateurs à Architecture Complexe Application à un Robot à Deux Bras*. Thèse de doctorat, l'Université Montpellier II (17 Février, 1994).
- Freedman, P. (1993). Robotics in the 1990's: An overview of current trends. *IEEE Canadian Review*.

- Fuchs, P. (1996). *Les interfaces de la réalité virtuelle*. Collection -Interfaces- Les journées de montpellier, Montpellier.
- Galerie, S. *Architecture ouverte de la commande adaptée à la robotique de coopération h/m. Application au domaine médical*. Thèse de doctorat, Université Paris XII (1989).
- Goldberg, K., Mascha, M., Getner, S., et Rothenberg, N. (1995). Desktop teleoperation via the world wide web. In *IEEE, International Conference on Robotics and Automation*, Nagoya, Japan.
- Grace, Vulkovich, et Chun (1993). Asix degree of freedom micromanipulator for ophthalmic surgery. In *IEEE Int. Conference on Robotics and Automaion*, pages 630–635, Atlanta, USA.
- Graves, S. et Volz, R. (1995). Action selection in teleautonomous systems. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*, volume 3, pages 14–19, Pittsburgh, PA, August 5-9.
- Greenberg, S. (1991). Computer-supported cooperative work and groupware: an introduction to the special issues. In *Int. Journal on Man-Machine Studies*, pages 133–141.
- Gzib, C. (www). <http://www.info-zip.org/pub/infozip/zlib/>.
- Hasegawa, T. (1991). A model-based tele-robot system with manipulation skills. In *ISART*, pages 449–506, Tokyo, Japan, March 91.
- Hayati, S. et Venkataraman, S. (1989). Design and implementation of a robot control system with traded shared control capability. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 1310–1315, May 14-19, Scottsdale, Arizona.
- Hickey, S., Manninen, T., et Pulli, P. P. (July. 23-26. 2000). Telereality - the next step for telepresence. In *World Multiconference on Systemics, Cybernetics and Informatics SCI 2000*, volume 3. Virtual Engineering and Emergent Computing, pages 65–70, Orlando, Florida, USA.
- Hunter, Jones, et Sagar (1994). A teleoperated microsurgical robot and associated virtual environment for eye surgery. In *PRESENCE*, volume 2 No. 4, pages 265–280.
- Joly, L. et Andriot, C. (1995). Imposing motion constraints to a force reflecting telerobot through real-time simulation of virtual mechanism. In *IEEE Int. Conf. on Robotics and Automation, ICRA '95*, pages 357–362, Nagoya, Japan.
- Kheddar, A. *Téléopération basée sur le concept du robo caché*. Doctorat de spécialité robotique, Université Pierre et Marie Curie (19 Decembre 1997).
- Kheddar, A., Atlani, D., Iles, A., et Blazevic, P. (1996). New trends in legged robots teleoperation. In *IEEE Int. Workshop on Robot and Human Communication, RO-MAN'96*, pages 232–237, Nov. 11-14, Tsukuba, Japan.
- Kim, W. S. (1993). Graphical operator interface for space telerobotics. In *IEEE Int. Conf. on Robotics and Automation*, pages 761–768, Atlanta, Georgia.
- Kim, W., Hannaford, B., et Bejczy, A. (1992). Force-reflection and shared compliant control in operating telemanipulators with time delay. In *IEEE Trans. on Robotics and Automation*, volume 8, No. 2, pages 176–185, April.
- Koren, Y. (1985). *Robotics for engineers*. McGraw-Hill, New York, USA.
- Kosuge, K., Itoh, T., Fukuda, T., et Otsuba, M. (1995). Tele-manipulation system based on task-oriented virtual tool. In *IEEE Int. Conf. on Robotic and Automation, ICRA '95*, pages 351–356, Nagoya, Japan.
- Kosuge, K., Murayama, H., et Takeo, T. (1996). Bilateral feedback control of telemanipulators via computer network. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'96*, volume 3, pages 1380–1385, Osaka, Japan.

- Lawrence, D. (1993). Stability and transparency in bilateral teleoperation. In *The IEEE Transactions on Robotics and Automation*, volume 9, No. 5, pages 624–637, October.
- Lederman, S. et Klatzky, R. (1994). The intelligent hand: An experimental approach to human object recognition and implications for robotics and ai. *AI Magazine*.
- Leung, G. et Francis, B. (1992). Channel”. In *30th Annual Allerton Conf. on Communication, Control and Computing*, pages 629–701, Monticello, IL, Sept. 30 - Oct. 2.
- Loukil, A. *Interface Homme-machine de Contrôle Commande en Robotique téléopérée*. Doctorat de robotique, Université d’Evry Val d’Essonne (1993).
- Mallem, M., Chavand, F., et Colle, E. (1992). Computer-assisted visual perception in teleoperated robotics. In *Robotica Journal*, volume 10, pages 93–103, Cambridge Univ. Press, England.
- Mallem, M., Rougeaux, et Mellanger, H. (1993). A trajectory generation module for 2d and 1/2 environment. In *IEEE Computers in design, manufacturing and production, 7th Annual European Computer conference*, pages 24–27, May.
- Mallem, M., M. Shaheen, X., Dourille, et Chavand, F. (1996). A matching method between an image and its 3d-model using a geometric constraint approach based on contact. In *CESA’96 IMACS Multiconf rence*, pages 565–569.
- Mallem, M., Shaheen, M., et Chavand, F. (1999). Automatic camera calibration based on robot calibration. In *the 16th IEEE Instrumentation and Measurement Technology Conference*, Venice, Italy, 24-26 mai.
- Michelman, P. et Allen, P. (1994). Shared autonomy in a robot hand teleoperation system. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems IROS’94*, volume 1, pages 253–259, Munich, Germany, Sept. 12-16.
- Milgram, P. et Kishino, F. (Dec. 1994). A taxonomy of mixed reality visual displays. In *IEICE Trans. on Information Systems, special issue on Networked Reality*, volume E77-D, N^o. 12, pages 1321–1329.
- Milgram, P., Rastogi, A., et grodski, J. (July. 5-7. 1995). Telerobotic control using augmented reality. In *Proceedings 4th IEEE International Workshop on Robot and Human Communication (RO-MAN’95)*, Tokyo.
- Moreau, G., Mallem, M., Chavand, F., et N’Zi, E. (1997). Two 3d recovering methods for robot control. In *IFAC’97, SYROCO*, pages 531–537.
- Nelson, M. et Gailly, J.-L. (1996). *The Data Compression Book*. New York, NY: M&T Books.
- Niemeyer, G. et Slotine, J. (1991). Stable adaptive teleoperation. In *The IEEE Journal of Oceanic Engineering*, volume 16, No. 1, pages 152–162, January.
- Nilles, J. (1998). *Managing telework: Strategies for managing the virtual Workforce*. John Wiley & Sons.
- Noyes, M. et Sheridan, T. (1984). A novel predictor for telemanipulation through a time delay. In *Annu. Conf. Manual Control*, Moffett Field, CA, NASA Ames Research Center.
- N’zi, C. *Mod lisation et reconstruction 3D interactive d’environnement: application   la t l op ration*. Doctorat de sp cialit  robotique, Universit  Evry Val d’Essonne, France (7 Decembre 1995b).
- N’zi, C., Chavand, F., Mallem, M., et Triboulet, J. (1994). Methods for updating the environment’s geometric database in telerobotics. In *IMACS SPRANN’94, Signal Processing Robotics And Neural Networks*, page 628.

- N'zi, C., Mallem, M., et Chavand, F. (1995a). Method for updating the environment 3d geometrical database using single camera view. In *IFIP WG 7.6 Working Conference*, pages 301–307.
- Otmane, S., Mallem, M., Mavel, S., et Chavand, F. (8 et 9 Avril 1999). Un système de réalité augmentée pour des applications télérobotique via interne. In *JJCR'11, Journée des Jeunes Chercheurs en Robotique*, pages 57–62, EPFL Lauzane.
- Otmane, S., Mallem, M., et Chavand, F. (3 et 4 février 2000a.). Les guides virtuels actifs pour l'assistance à la téléopération via internet. In *JJCR'12, Journée des Jeunes Chercheurs en Robotique*, pages 118–124, Bourges, France.
- Otmane, S., Mallem, M., Kheddar, A., et Chavand, F. (April 16-20, 2000b). Ariti: an augmented reality interface for teleoperation on the internet. In *Advanced Simulation Technologies Conference (ASTC2000) - High Performance Computing*, pages 254–261, Wyndham City Center Hotel, Washington, D.C., USA.
- Otmane, S., Mallem, M., Kheddar, A., et Chavand, F. (April 16-20, 2000c). Active virtual guide as an apparatus for augmented reality based telemanipulation system on the internet. In *IEEE Computer Society - 33rd Annual Simulation Symposium ANSS2000*, pages 185–191, Wyndham City Center Hotel, Washington, D.C., USA.
- Otmane, S., Colle, E., Mallem, M., et Hoppenot, P. (July 23-26, 2000d). Disabled people assistance by a semiautonomous robotic system: Use of virtual reality to enhance human performances. In *World Multiconference on Systemics, Cybernetics and Informatics (SCI2000)*, volume 3 - Virtual Engineering and Emergent Computing, pages 684–689, Orlando, Florida, USA.
- Oyama, E., Tsunemoto, N., Tachi, S., et Inoue, Y. (1992). Remote manipulation using virtual environment. In *2nd Int. Symp. on Measurement and Control in Robotics, ISMCR'92*, pages 311–318, Tsukuba Science City, Japan, Nov. 15-19.
- Pampagnin, L. *Reconnaissance d'objets tridimensionnels en perception monoculaire et multisensorielle - Application à la robotique spatiales*. Thèse de doctorat, Université Paul Sabatier, Toulouse (31 Octobre 1990).
- Peroche, B., Ghazanfarpour, D., Argence, J., et Michelucci, D. (1988). *La synthèse d'image*. Hermes.
- Pira, R. (1998). New geometric solids in ray tracing. http://www.engr.mun.ca/~rahim/graphics/superquad_noframe.html.
- Qiang, S. *Stratégies de localisation et identification d'objet à partir de quelques mesures tridimensionnelles*. Thèse de doctorat, Institut National Polytechnique de Lorraine (10 Juillet 1989).
- Rastogi, A., Milgram, P., Drasic, D., et Grodski, J. J. (1996). *SPIE Volume 2653: Stereoscopic Displays and Virtual Reality Systems III*. Mark T. Boalas and Scott S. Fisher and John O. Merritt, San Jose, California, USA.
- Rosenberg, L. (1992). The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments. In *Technical Report, No. AL-TR-1992-XXX, USAF Armstrong Laboratory, WPAFB OH*.
- Rosenberg, L. (1993). The use of virtual fixtures to enhance telemanipulation with time delay. In *Proceedings, ASME Winter Annual Meeting on Haptic Interfaces for Virtual environment and Teleoperator Systems*, New Orleans, Louisiana.
- Saucy, P. et Mondala, F. (Oct. 12-17, 1998). Khepentheweb: One year of access to a mobile robot on the internet. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'98*, pages 23–30, Victoria, B.C. Canada.

- Sayers, C. et Paul, R. (1994). An operator interface for teleprogramming employing synthetic fixtures. In *Presence*, volume 3, No. 4.
- Sayers, C., Lai, A., et Paul, R. (May, 1995). Visual imagery for subsea teleprogramming. In *IEEE Robotics and Automation Conference*.
- Schal, T. et Zeller, B. (1990). A methodological approach to computer-supported cooperative work. In *Fifth European Conference on Cognitive Ergonomics*, pages 3–6.
- Schulz, D., Burgard, W., et Cremers, A. (Oct. 12-17, 1998). Predictive simulation of autonomous robots for tele-operation systems using the world wide web. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'98*, pages 31–36, Victoria, B.C. Canada.
- Shaheen, M. *Reconnaissance d'objets polyédriques à partir d'une image vidéo pour la téléopération*. Doctorat de robotique, Université d'Evry Val d'Essonne (9 avril 1999).
- Sheridan, T. B. (1992). *Telerobotics, Automation and Human Supervisory Control*. The MIT Press, Cambridge, USA.
- Sheridan, T. (1993). Space teleoperation through time delay: Review and prognosis. *The IEEE Trans. on Robotics and Automation*, 9:592–606.
- Simmons, R. (Oct. 12-17, 1998). Xavier: An autonomous mobile robot on the web. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'98*, pages 43–48, Victoria, B.C. Canada.
- Spenny, C. et Schneider, D. (1997). Object resolved teleoperation. In *IEEE Int. Conf. on Robotics and Automation, ICRA'97*, Albuquerque, NM, USA, April 20-25.
- Spong, M. (1993). Communication delay and control in telerobotics. In *Journal of the Robotics Society of Japan*, volume 11, No. 6, pages 803–810.
- Stein, M. (Oct. 12-17, 1998). Painting on the world wide web: The pumapaint project. In *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'98*, pages 37–42, Victoria, B.C. Canada.
- Tarn, T. J., Xi, N., Guo, C., et Bejczy, A. (1995). Function-based control sharing for robotic systems. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*, volume 3, pages 1–6, Pittsburgh, PA, August 5-9.
- Taylor, K. et Trevelyan, J. (Oct. 1995). Australia's telerobot on the web. In *26th International Symposium On Industrial Robots*, Singapore.
- Thibout, C. *Réalité Virtuelle et Langages Graphiques: une application pour la téléopération*. Doctorat de mention informatique, Université de Rennes 1 (07 novembre 1996).
- Triboulet, J. *Caractérisation d'un système multicapteur pour la modélisation d'environnement: application à la téléopération*. Doctorat de spécialité robotique, Université d'Evry Val d'Essonne, France (1996).
- Uenohara, M. et Kanade, T. (1995). Vision-based object registration for real-time image overlay. In *Computer Vision, Virtual Reality and Robotics in Medicine: CVRMed '95*, pages 14–22, N. Ayache. Berlin, Springer-Verlag.
- Urban, E. C. (1995). The information warrior. In *IEEE Spectrum*, volume 11, pages 66–70.
- Vertut, J. et Coiffet, P. (1984). *Les Robots: Téléopération*. Edition Hermes, Vol 3A et 3B, Paris, France.
- Wang, C., Cannon, D. J., et MA, H. (1996). A human-machine system interacting virtual tools with a robotic collision avoidance concept using conglomerates of spheres. In *Journal in Teleoperation*.

Zak, H. et Das, H. (1995). Towards a training methodology for skilled teleoperation. In *IEEE Trans. on Systems Man and Cybernetics*, volume 25, No. 2, pages 313–327, February.

Table des figures

1.1	Illustration de l'architecture générale d'un système de Téléopération .	11
1.2	Exemples de système : A gauche, le système de téléopération du CEA/CEREM; A droite, le système de téléprésence avec un robot mobile du groupe de recherche en téléprésence "California Corporation Telepresence Research"	11
1.3	Classement de la TAO, TDO et TSA selon le degré d'autonomie du dispositif esclave.	13
1.4	Exemple du système Poste de Travail Mobile (PTM) du CEA piloté par un contrôleur TAO 2000	15
1.5	Exemples de robots: A gauche, l'exploration de Mars par "sojourner", à droite , la maintenance de satellites	15
1.6	Exemple d'utilisation d'un robot embarqué sur un véhicule sous marin JASON ((Sayers <i>et al.</i> , 1995))	16
1.7	Illustration de la téléchirurgie	16
1.8	Rehaussement d'images vidéo dégradées par superposition d'une image synthétique "fil de fer" à une image caméra	21
1.9	Utilisation de la prédiction graphique pour la téléopération spatiale. A gauche prédiction et confirmation de la tâche, à droite exécution . . .	23
1.10	Architecture du système ARTEMIS	24
1.11	schéma général du système de téléopération proposé par Kheddar . .	25
2.1	les mécanismes virtuels pour la télémanipulation: A gauche, la structure avec les coordonnées cylindriques (x, r, θ_1) et l'impédance mécanique attachée au bout de l'outil $(\theta_2, \theta_3, \theta_4)$ qui permet le suivi de surface. A droite, un mécanisme virtuel associé à la tâche d'assemblage de la boîte saisie par le robot.	32
2.2	Téléopération basée sur le concept de mécanismes virtuels (d'après Joly et Andriot 95	33
2.3	Exemple d'outils virtuels: en haut, des champs de force répulsifs pour l'évitement de zones dangereuses; en bas, des mécanismes virtuels à placer sur l'effecteur du robot pour guider ses mouvements (ici la rotule pour le blocage des translations, la glissière pour le suivi de droite et de pivot)	34
2.4	La métaphore de la boîte à outils virtuels	34

2.5	le système de téléprésence utilisé pour tester les performances de l'opérateur lors des tâches d'insertion avec ou sans l'aide des guides virtuels	35
2.6	Exemples de guides virtuels projetés dans la zone de manipulation pour assister l'opérateur dans des tâches d'insertion d'après Rosenberg	35
2.7	Le système de téléprogrammation utilisé par Sayers: A gauche, le site maître; A droite le site esclave.	36
2.8	exemple de guides virtuels d'assistance à l'opérateur proposés par Sayers en 1994: Première ligne, guide virtuel point-point pour déplacer la pointe de l'effecteur jusqu'à un point dans l'espace et ensuite effectuer des rotations; Deuxième ligne, pour le suivi de zones circulaires; Troisième ligne, pour effectuer un contact surface-surface.	37
2.9	Guides virtuels dédiés exclusivement à l'assistance de l'opérateur. Une projection du repère du pouce sur le sol permet un guidage plus précis pour saisir ou manipuler les objets virtuels.	37
2.10	Exemple d'application des guides virtuels. Ajustement d'un préhenseur limité en capacité d'ouverture monté sur un robot plan de type SCARA. Cette figure illustre aussi les transformations entre l'action de l'opérateur et le robot intermédiaire de transformation.	38
2.11	Classification de la métaphore des guides virtuels.	38
2.12	Exemple d'une parabole générée par la famille des coniques	40
2.13	Exemple d'hyperboles et d'ellipse générées par la famille des coniques	41
2.14	Exemple de superellipses, de haut en bas et de gauche à droite avec $p=0.38, 0.78, 1, 1.78, 2, 3$ et 6	42
2.15	ellipsoïdes avec $\varepsilon_1 = \varepsilon_2 = 1, a_1 = a_2 = a_3$ et $\varepsilon_1 = \varepsilon_2 = 1, a_1 > a_2 > a_3$	42
2.16	ellipsoïdes avec $\varepsilon_1 = 1, \varepsilon_2 = 2; \varepsilon_1 = 0.1, \varepsilon_2 = 1; \varepsilon_1 = \varepsilon_2 = 0.35$	43
2.17	ellipsoïdes avec $\varepsilon_1 = \varepsilon_2 = 3; \varepsilon_1 = \varepsilon_2 = 2; \varepsilon_1 = 3, \varepsilon_2 = 2$	43
2.18	supertoroïdes avec $\varepsilon_1 = \varepsilon_2 = 1, \varepsilon_1 = \varepsilon_2 = 0.5$	43
2.19	supertoroïdes avec $\varepsilon_1 = 3, \varepsilon_2 = 1; \varepsilon_1 = 1, \varepsilon_2 = 3; \varepsilon_1 = \varepsilon_2 = 2$	44
2.20	Cône, Cylindre, tube, cube, disque et plan	47
2.21	Représentation simplifiée de la projection perspective, f étant la focale.	47
2.22	Exemple de sélection d'un objet virtuel sur écran	48
2.23	Exemple d'un cube avant déformation	49
2.24	Résultat de la déformation sur le cube	49
2.25	Plan, Cube ouvert et Cylindre: En haut, avant déformation; En bas après déformation avec $p = 0.99$ et $f = 0.9$	50
2.26	Collision entre l'effecteur du robot et un objet guide sous forme d'une demi-sphère	51
2.27	La géométrie des guides: de gauche à droite: Cube, (cylindre ou cône ou tube), (cône ou tube droits), plan, disque.	52
2.28	Zone de collision de seuil "d" autour d'un segment [AB]	53
2.29	Existence de trous dans les guides virtuels	53
2.30	Modèle utilisé à l'écran (en haut) et modèle utilisé pour les calculs (en bas)	54

2.31	Exemple de sphère de sécurité	54
2.32	Exemple de “guide virtuel simple”, pour assister l’opérateur et/ou le robot à atteindre une cible	56
2.33	Exemple de “guide virtuel composé”, pour assister l’opérateur et/ou l’effecteur du robot à suivre une trajectoire et manipuler un objet	56
2.34	Exemple de “guide virtuel composé”, pour assister l’opérateur et/ou un robot mobile à traverser un porte	57
2.35	Exemple de “guide virtuel répulsif”, à gauche, une barrière et à droite, un volume englobant répulsif	57
2.36	Exemple de “guide virtuel attractif”, pour atteindre un objet de l’environnement avec un outil porté par le robot	58
2.37	Représentation des différents types de données générés lors de la création d’un guide virtuel	59
2.38	Schéma général de la structure des propriétés d’un guide virtuel	60
3.1	Représentation informatique BREP d’un polyèdre	66
3.2	Représentation CSG simple	66
3.3	Exemple de représentation interne d’un triangle dans l’espace	68
3.4	Modèles géométriques de caméra	69
3.5	Les composantes du robot à 4 ddl utilisé pour la calibration	73
3.6	Les différents repères du système de mesures	73
3.7	Illustration du matériel utilisé pour la réalisation du système ARITI	77
3.8	Architecture simplifiée du système ARITI	78
3.9	Le continuum Réalité-Virtualité d’après Milgram 1994 et Hickey 2000	79
3.10	Représentation des données réelles et virtuelles dans un environnement mixte	79
3.11	Les trois types d’assistance pendant le télétravail	80
3.12	Illustration de l’assistance à la perception de l’environnement	80
3.13	Illustration de l’assistance à la commande	81
3.14	Organigramme d’assistance à la supervision	81
3.15	Schéma simplifié de la téléopération sans assistance à la commande	82
3.16	Schéma simplifié de la téléopération avec assistance à la commande	82
3.17	Illustration de la tâche “désignation d’une cible à l’écran”	84
3.18	Schéma simplifié de l’exécution de tâches en mode Téléprogrammation.	84
3.19	Illustration à la Télésupervision.	85
3.20	Exemple de préparation et d’exécution d’une mission, (A) : préparation et exécution par un seul opérateur, (B) : Télétravail coopératif avec 4 opérateurs.	86
3.21	Architecture générale de communication avec le système ARITI: En pointillés, une possibilité d’accéder directement au serveur ARITI sans passer par le serveur Web	87
3.22	Protocole de demande d’un numéro d’identification	87
3.23	Protocole de communication client/serveur image	89

3.24	Schéma de fonctionnement simplifié du serveur image	89
3.25	Schéma détaillé du fonctionnement du serveur image	90
3.26	Protocole de communication client/serveur commande. Le caractère “S” désigne la demande de contrôle du robot	91
3.27	Protocole de communication entre le serveur de commande et les autres clients superviseurs : le caractère “F” désigne la demande de l’état final du robot	92
3.28	Schéma détaillé du fonctionnement du serveur de commande	94
4.1	Image du robot et de la zone de manipulation.	98
4.2	Image de l’armoire de commande (a) et de la caméra montée sur une tourelle (b)	98
4.3	Caractéristiques d’un objet à manipuler et de son support	99
4.4	Illustration des deux types de clients “Administrateur” et “Utilisateur” du système ARITI	100
4.5	l’IHM destinée au client administrateur, avec en haut à gauche, une image vidéo de résolution 288×384 , en bas à droite la partie réservée à l’administrateur pour créer des guides virtuels.	101
4.6	l’IHM destinée au client utilisateur, avec en haut à gauche, une image vidéo de résolution 192×256 , et l’utilisateur ne peut pas créer de guides virtuels.	102
4.7	(A) : Tableau de bord de l’interface de ARITI. (B) : mode téléopération, (C) : mode téléprogrammation de tâche, (D) : contrôle du robot réel et le mode télé supervision ou télécoopération	103
4.8	L’interface du modeleur 3D pour la création et la manipulation de guides virtuels.	104
4.9	Générateur de guides virtuels: Exemple d’un plan généré à partir de deux paramètres, la longueur et la largeur.	105
4.10	Guide virtuel utilisé pour atteindre une cible: ici la cible est un point 3D se trouvant sur la périphérie de l’objet cylindre	105
4.11	Propriétés du guide virtuel d’assistance à la téléopération pour atteindre une cible. La cible dans ce cas est l’objet cylindre numéro 3 (figure 4.1)	106
4.12	Illustration du principe de suivi de cible mobile par l’effecteur du robot.	108
4.13	Exemple de suivi de cible mobile par le robot. La cible est atteinte par le robot (A), a effectué une translation par rapport à l’axe X (B), ensuite une rotation par rapport à l’axe Z (C) et enfin une translation par rapport à l’axe Z (D)	109
4.14	Le guide virtuel utilisé pour le suivi de contour. Le point “E” désigne l’extrémité de l’effecteur du robot et le point “A” appartient au contour du disque de passage de “E”.	109
4.15	Propriétés du guide virtuel utilisé pour le suivi de contour. Dans ce cas ce guide permet de suivre le contour du cylindre numéro 1	110

4.16	Le guide virtuel utilisé pour la saisie d'un objet cylindre. Le point "E" désigne l'extrémité de l'effecteur du robot et le point "A" désigne le centre du disque passant par le point "E".	111
4.17	Les propriétés des guides virtuels utilisés pour la saisie de l'objet cylindre numéro 3.	112
4.18	Le guide virtuel utilisé pour le dépôt d'un objet cylindre. (A) : Positionnement des différents guides, avec le point "E" qui désigne l'extrémité de l'effecteur du robot et le point "A" le centre du disque passant par le point "E"; (B) : Exemple d'influence des guides sur les trajectoires réalisées par les robots virtuel et réel.	114
4.19	Les propriétés des guides virtuels utilisés pour le dépôt de l'objet cylindre numéro 3 sur le support numéro 1.	115
4.20	Mise à jour de l'architecture du système ARITI pour le contrôle d'un robot mobile	117
4.21	Les images des deux robots mobiles utilisés. (A) : le robot utilisé avec le système ARITI, (B) : le robot destiné au projet d'assistance aux personnes handicapées	117
4.22	L'IHM utilisée pour le contrôle et la supervision du robot mobile. . .	118
4.23	Illustration de la téléopération d'un robot mobile avec assistance de deux guides virtuels répulsifs sous forme de plans. (A) : Le robot mobile subit l'influence des deux guides, (B) : le robot traverse la porte. . . .	119
4.24	Propriétés des deux guides virtuels utilisés pour traverser une porte par le robot mobile.	120
4.25	Simulation du parcours d'une B-Spline par le robot mobile virtuel. (A) : l'état initial, (B) : pendant le parcours, (C) : le robot traverse la porte, (D) : l'état final	121
4.26	Propriétés du guide virtuel généré par une B-Spline	121
4.27	Illustration du télé-enseignement	123
5.1	Vue rapprochée de l'effecteur du robot et des trois cylindres en polystyrène.	126
5.2	Illustration du découpage de l'environnement virtuel en trois zones . .	128
5.3	Illustration du signal visuel utilisé pour atteindre le cylindre numéro 1	129
5.4	Gestion des collisions pendant que l'opérateur essaie d'atteindre le cylindre numéro 1. (A) : détection des collisions, (B) : le nombre total de collisions réalisé par chaque opérateur.	129
5.5	Les imprécisions obtenues lorsque la cible est atteinte. (A) : la moyenne des dix tests pour chaque opérateur suivant X, Y et Z, (B) : la moyenne sur les dix opérateurs suivant X, Y et Z.	130
5.6	Le temps moyen pour atteindre la cible (cylindre numéro 1) à chaque essai.	130
5.7	Utilisation d'un guide virtuel pour atteindre le cylindre numéro 1 . .	131

5.8	Les imprécisions obtenues lorsque la cible est atteinte en utilisant un guide virtuel repulsif. (A): la moyenne des dix tests pour chaque opérateur suivant X , Y et Z , (B): la moyenne sur les dix opérateurs suivant X , Y et Z	131
5.9	Le temps moyen pour atteindre la cible (cylindre numéro 1) à chaque essai en utilisant un guide virtuel répulsif	132
5.10	Les imprécisions obtenues suivant X , Y et Z , lorsque la cible est atteinte en utilisant un guide virtuel attractif	133
5.11	Le temps moyen pour atteindre la cible (cylindre numéro 1) à chaque essai en utilisant un guide virtuel attractif	134
5.12	Utilisation d'un guide virtuel pour saisir et sortir le cylindre numéro 1 depuis son support métallique.	134
5.13	Le temps moyen pour saisir le cylindre numéro 1 à chaque essai en utilisant un guide virtuel répulsif ou attractif. En pointillé sont représentées les moyennes pour les dix opérateurs.	135
5.14	Utilisation d'un guide virtuel pour déposer le cylindre numéro 1 sur le support métallique numéro 3.	135
5.15	Le temps moyen pour déposer le cylindre numéro 1 sur le support numéro 3 à chaque essai en utilisant un guide virtuel répulsif ou attractif. En pointillé sont représentées les moyennes sur les dix opérateurs	136
5.16	Evaluation des algorithmes de compression suivant la dynamique de l'image	137
5.17	Les temps de compression lorsqu'il y a peu de mouvements dans l'image.	138
5.18	Les temps de compression lorsqu'il y a beaucoup de mouvements dans l'image.	138
5.19	Organigramme de choix automatique de méthodes de compression d'image. l'image est envoyée aux clients avec le type de compression utilisé.	139
5.20	Evaluation du temps de capture d'images vidéo T_i avec dix images successives.	140
5.21	Evaluation du temps d'envoi T_e suivant la dynamique de l'image. Le temps représenté est une moyenne sur dix images envoyées.	141
5.22	Evaluation des temps serveur et réponse du client lorsqu'il y a peu de mouvements dans l'image. (A): Pour des images brutes, (B): Pour des images compressées avec Gzip.	142
5.23	Evaluation des temps serveur et réponse du client lorsqu'il y a peu de mouvements dans l'image. (A): Pour des images prétraitées ensuite compressées en Gzip, (B): Pour des images prétraitées en compressées avec Huffman.	142
5.24	Evaluation des temps serveur et réponse du client lorsqu'il y a beaucoup de mouvements dans l'image. (A): Pour des images compressées avec Gzip, (B): Pour des images prétraitées et ensuite compressées avec Gzip.	143
5.25	Evaluation des temps serveur et réponse du client à moyenne distance. L'adresse Internet du site client est "Toulouse-17-133.abo.wanadoo.fr".	144

5.26	Evaluation des temps serveur et réponse du client à moyenne distance. L'adresse Internet du site client est "pc-41-13.inrets.fr".	144
5.27	Evaluation des temps serveur et réponse du client à moyenne distance. L'adresse du site client est "dyn-213-36-102-192.ppp.libertysurf.fr".	144
5.28	Evaluation des temps serveur et réponse du client à moyenne distance. L'adresse Internet du site client est "pc.area.ba.cnr.it".	145
5.29	Evaluation des temps serveur et réponse du client à longue distance. L'adresse du site client est "1Cust161.tnt1.san-francisco3.ca.da.uu.net"	146
5.30	Evaluation des temps serveur et réponse du client à longue distance. L'adresse Internet du site client est "internet-pub2.biblio.polymtl.ca"	146
5.31	Les temps moyens, serveur et reprise du serveur, pour une connexion de 1 à 5 clients lorsqu'il y a peu de mouvements dans l'image. (A) : Pour des images brutes, (B) : Pour des images compressées avec Gzip.	148
5.32	Les temps moyens, serveur et reprise du serveur, pour une connexion de 1 à 5 clients lorsqu'il y a peu de mouvements dans l'image. (A) : Pour des images prétraitées ensuite compressées en Gzip, (B) : Pour des images prétraitées et compressées avec Huffman.	148
5.33	Les temps serveur, et reprise du serveur, pour une connexion de 1 à 5 clients lorsqu'il y a beaucoup de mouvements dans l'image. (A) : Pour des images compressées avec Gzip, (B) : Pour des images prétraitées et ensuite compressées avec Gzip.	149
5.34	Temps de mise à jour des images réelles et des environnements virtuels en fonction du nombre d'opérateur utilisant le système ARITI en connexion locale.	150
5.35	Illustration des transferts d'informations multimodales via Internet entre le site maître (client) et le site esclave (serveur).	156
5.36	Illustration de la communication entre le système ARITI et MCIT pour faire du recalage d'objets virtuels sur les objets réels.	157
5.37	Illustration de la téléopération de deux robots par deux clients distants en utilisant une extension du système ARITI.	158
A.38	Illustration de l'arbre binaire de Huffman pour la compression d'images	i
B.39	Tâche de téléprogrammation "Atteinte du cible"	v
B.40	Tâche de téléprogrammation "Atteinte une cible mobile"	vi
B.41	Tâche de téléprogrammation "Suivre le contour du cylindre numéro 1"	vi
B.42	Tâche de téléprogrammation "Saisir ou Déposer un objet"	vii
B.43	Tâche de téléprogrammation "Tour de Hanoi"	viii

Liste des tableaux

1.1	Développement du télétravail en Europe, 1998-1999. d'après (Béréziat <i>et al.</i> , 2000)	7
1.2	Etude marketing des applications de RV	18
2.1	Les paramètres des guides virtuels pour la détection des collisions. . . .	51
2.2	Les équations des guides virtuels pour la détection des collisions	52
2.3	Les différents cas d'utilisation des guides virtuels associés à chaque acteur.	61
3.1	Les paramètres obtenus lors de calibration de la tige du robot	75
4.1	Exemple d'organisation d'une équipe pour la réalisation d'une mission de téléopération en coopération	116
5.1	Vitesses de déplacement et de rotation des composantes du robot réel .	127
5.2	Les pas de déplacement et de rotation du robot virtuel	127
5.3	Quelques résultats de connexion à moyenne distance	145
5.4	Quelques résultats de connexion à longue distance	147
5.5	La moyenne des temps pour la mise à jour des images réelles et des environnements virtuels lors d'un télétravail coopératif avec plusieurs opérateurs. Dans ce cas, les opérateurs se trouvent dans le même laboratoire.	150
5.6	La moyenne des temps pour la mise à jour des images réelles et des environnements virtuels lors d'un travail coopératif avec deux opérateurs distants. Dans ce cas, les opérateurs se trouvent en Espagne.	151
5.7	La moyenne des temps pour la mise à jour des images réelles et des environnements virtuels lors d'un travail coopératif avec deux opérateurs distants. Dans ce cas, le premier se trouve en Italie et le deuxième au Canada.	151
C.8	Quelques adresses Internet des sites clients en Europe ayant utilisé ARITI	ix
C.9	Quelques adresses Internet des sites clients dans le monde ayant utilisé ARITI	x
C.10	suite: Quelques adresses Internet des sites clients dans le monde ayant utilisé ARITI	xi

Annexes

A Les méthodes de compression Gzip et Huffman

A.1 Codage de Huffman

Le codage de Huffman est préparé par un algorithme spécial que nous avons mis en œuvre pour avoir une bonne compression d'images dans certains cas (peu de mouvements dans l'image). Chaque élément de l'image (par exemple: un octet) est remplacé, suivant un alphabet spécifique, par une série de 0 ou de 1.

Pour compresser, l'algorithme de Huffman applique deux passes :

- Première passe: Permet de mesurer le poids de chaque élément dans l'image. Ce poids correspond au nombre de fois qu'un élément est présent dans le fichier à compresser. Puis, grâce aux poids obtenu, on en déduit un arbre binaire de Huffman où chaque fin de branche correspond à un élément (Voir FIGURE A.38).
- Deuxième pass: Permet de remplacer chaque élément du fichier par son alphabet de Huffman correspondant.

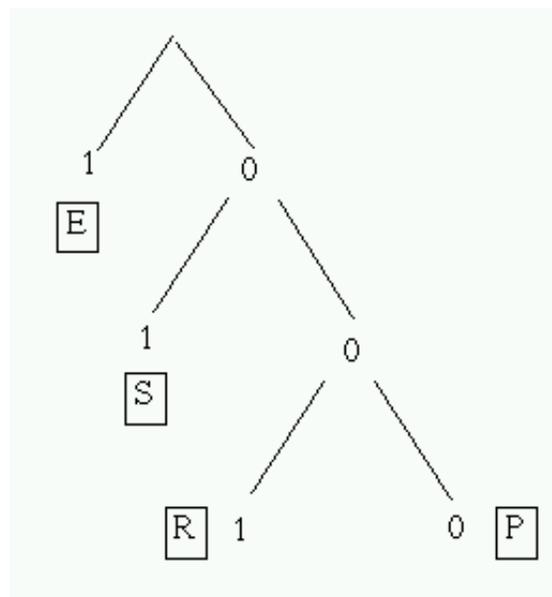


FIG. A.38 – Illustration de l'arbre binaire de Huffman pour la compression d'images

Pour le cas du serveur image, nous avons choisi un cas particulier de l'arbre de Huffman: les branches de gauche (qui ont une valeur de 1) n'ont pas de branches filles. L'avantage avec cet arbre, c'est que son codage reste simple: on a juste à donner le nombre d'éléments suivi des éléments puis du codage.

A.1.1 Avantage de la compression d'Huffman

- Rapide (GZip demande plus de temps).
- Bonne compression dans le cas où seulement quelques éléments ont des poids très grands.
- Compression sans perte.

Par conséquent cette compression peut être très mauvaise s'il y a beaucoup de mouvements dans l'image.

A.1.2 Exemple de compression

soit par exemple le mot 'PRESSEEPRESSEE':

- Poids de 'P': 2
- Poids de 'R': 2
- Poids de 'E': 6
- Poids de 'S': 4

Grâce à l'arbre créé, la compression donne:

000.001.1.01.01.1.1.000.001.1.01.01.1.1

Soit 26 bits, donc 4 octets (sans compter l'entête) au lieu de 14 octets.

Le codage serait par exemple:

14d, 04d, E, S, R, P, 00000110b, 10111000b,00110101b,11000000b

au lieu de:

14d, 'P', 'R', 'E', 'S', 'S', 'E', 'E', 'P', 'R', 'E', 'S', 'S', 'E', 'E'

soit 10 octets au lieu de 15.

A.2 Codage de Gzip

Pour compresser en Gzip, on utilise la librairie '**Zlib**'. Cette compression utilise une méthode de dégonflement' qui correspond à un codage de Huffman spécifique et un compactage LZ77.

Dans la méthode de dégonflement', on ajoute 2 règles supplémentaires dans l'algorithme de Huffman classique:

- Les éléments qui ont des codes plus courts sont placés à la gauche de ceux de plus longs codes.
- Le positionnement d'éléments ayant des codes de même longueur, sont toujours placés vers la gauche.

La compression de LZ77 consiste à retrouver des mots qui sont répétés. Par exemple, si on a:

'Blah blah blah blah blah!'

on pourra le coder par

'Blahb[$D = 5, L = 18$]!'.

[$D = 5$] veut dire que le début de la chaîne a une distance de 5, ce qui donne le mot 'lah b'

[$L = 18$] veut dire que le mot sera répété jusqu'à obtenir une longueur de 18 caractères. La compression Gzip utilise la compression LZ77 en premier, puis la compression de Huffman.

A.2.1 Avantage de la compression GZip

- Rapide si le niveau de compression est bas.
- Bonne compression en général (allant de 5:1 à 2:1).
- Compression sans perte.
- La librairie existe C/C++ et en Java.

Par contre, la compression Gzip devient trop lente pour un niveau de compression trop haut.

B Schémas de fonctionnement des tâches de téléprogrammation

B.1 Atteindre un point cible

La figure ci-dessous (B.39) montre le schéma de fonctionnement de la tâche consistant à atteindre une cible dont l'opérateur fournit les coordonnées 3D.

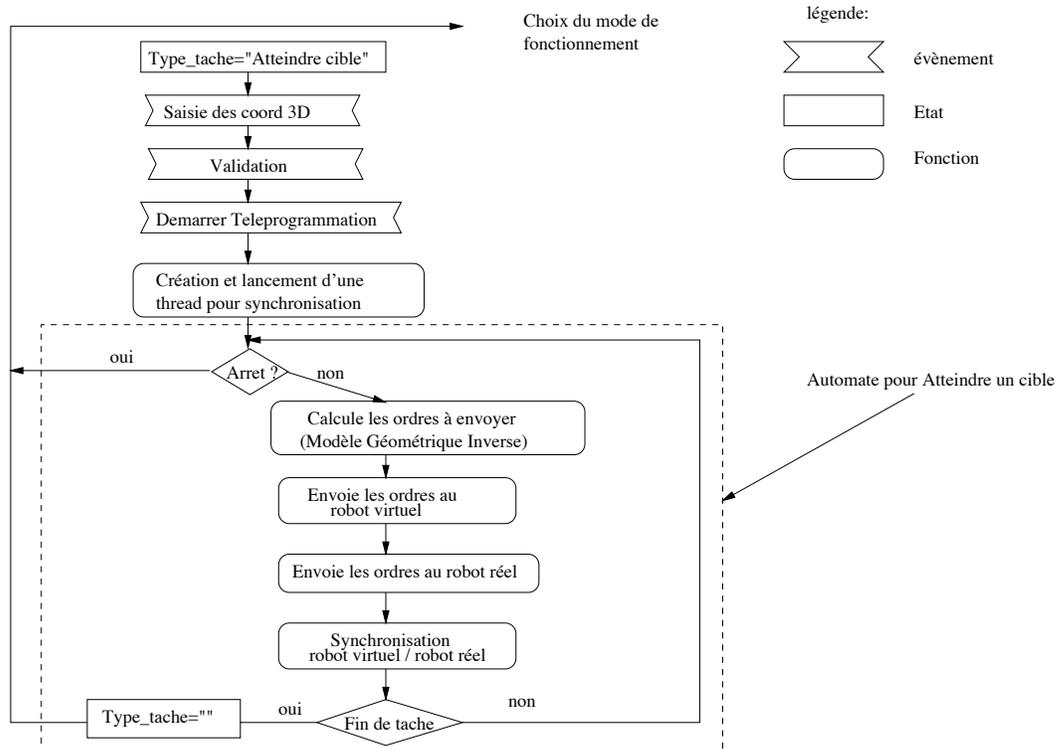


FIG. B.39 – Tâche de téléprogrammation “Atteinte du cible”

B.2 Atteindre une cible mobile

La figure B.40 montre le schéma de fonctionnement de la tâche consistant à atteindre une cible mobile.

Cette tâche utilise donc l'automate précédent pour atteindre une cible.

B.3 Suivre le contour du cylindre 1

La figure B.41 montre le schéma de fonctionnement de la tâche consistant à suivre le contour du cylindre 1. Elle utilise aussi l'automate pour atteindre une cible.

B.4 Prendre/déposer un objet

La figure B.42 montre le schéma de fonctionnement de la tâche consistant à prendre ou à déposer une cylindre.

A tout instant, on connaît l'état mémoire du système grâce à deux tableaux: le premier

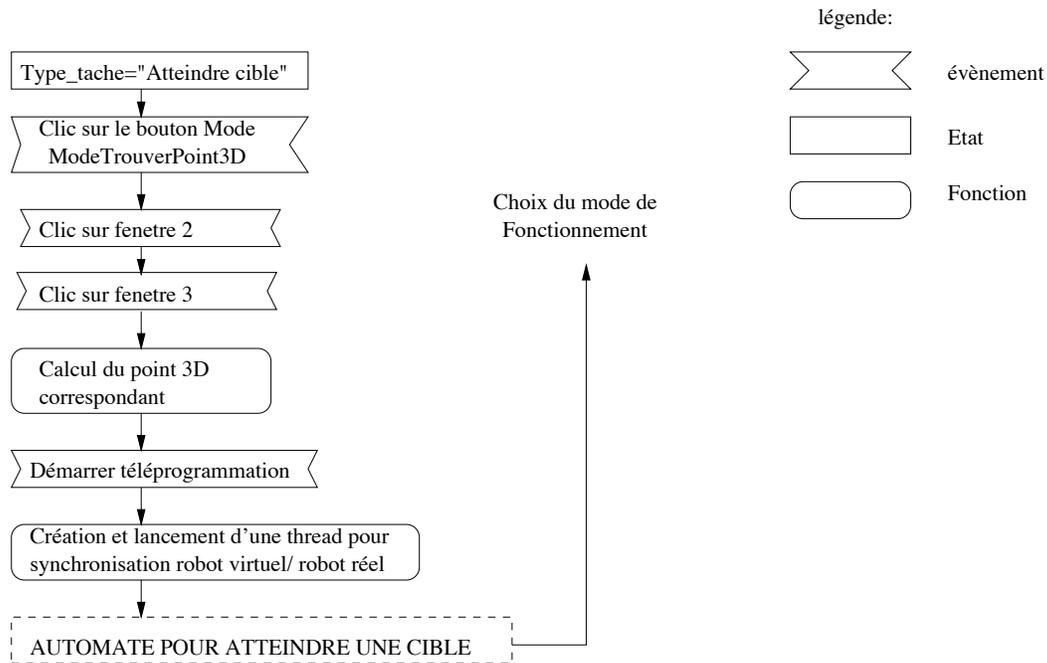


FIG. B.40 – Tâche de téléprogrammation “Atteinte une cible mobile”

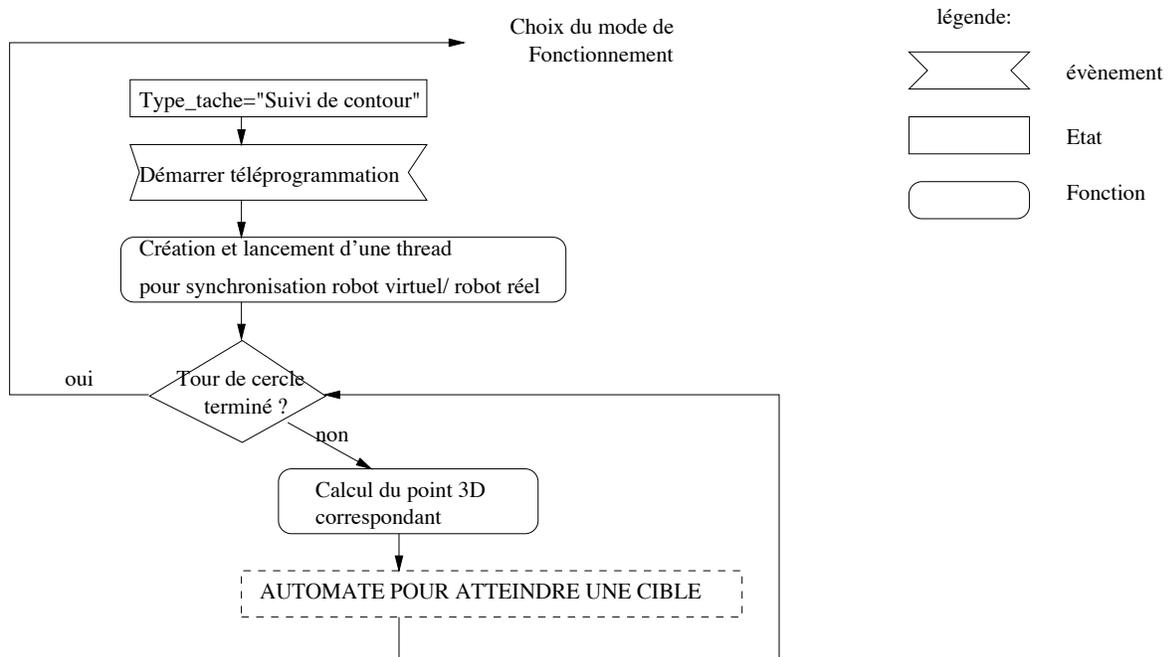


FIG. B.41 – Tâche de téléprogrammation “Suivre le contour du cylindre numéro 1”

tableau contient le numéro du support de chaque cylindre et le deuxième tableau contient la position du cylindre sur son crochet.

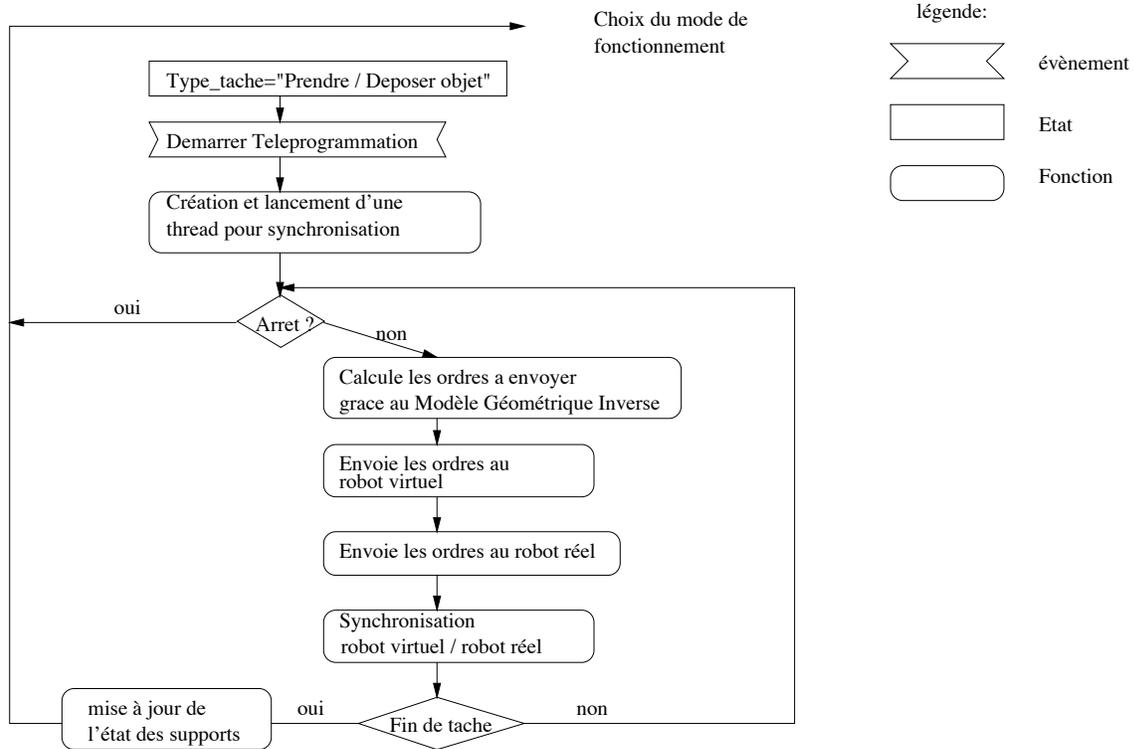


FIG. B.42 – Tâche de téléprogrammation “Saisir ou Déposer un objet”

B.5 Tour de Hanoï

La figure B.43 montre le schéma de fonctionnement de la tour de Hanoï. Ce schéma utilise l’automate permettant d’atteindre une cible.

L’algorithme de la tour de Hanoï correspond à une succession de tâches de prise et de dépôt d’objets. On peut donc créer une tâche complexe à partir d’un ensemble de tâches plus simples.

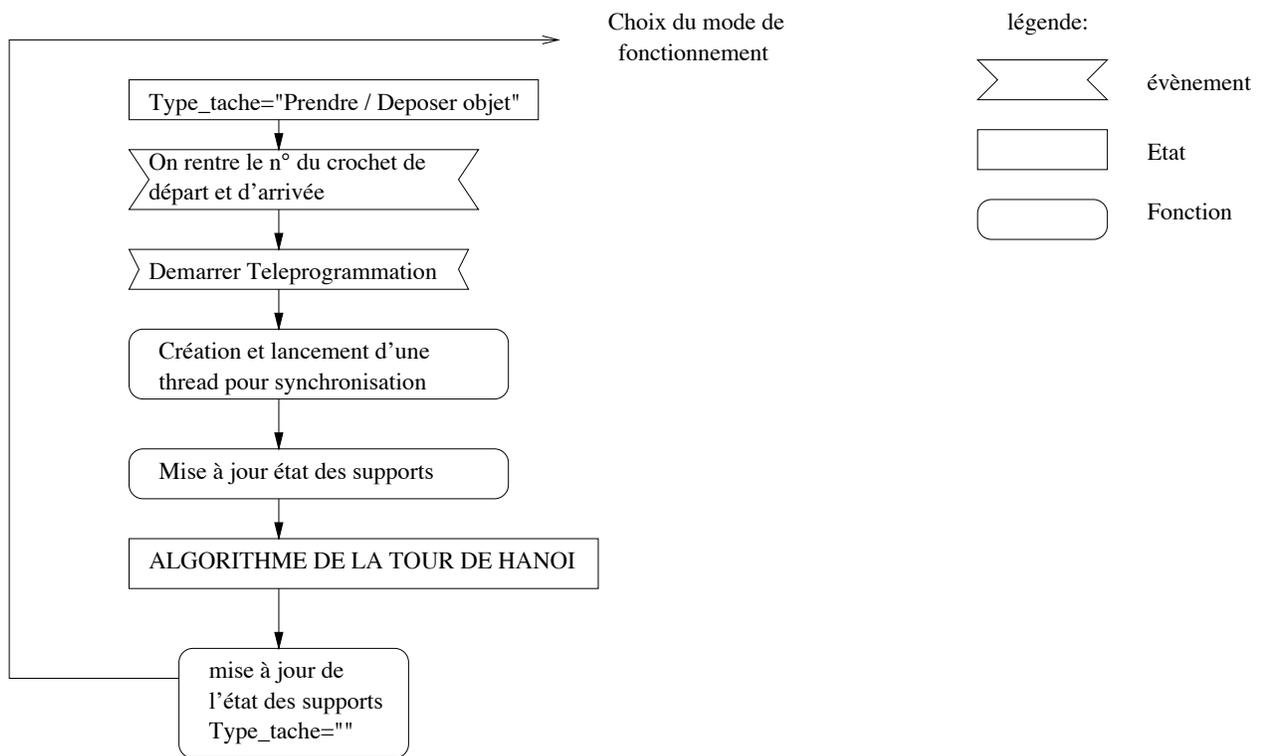


FIG. B.43 – Tâche de téléprogrammation “Tour de Hanoi”

C Les sites clients ayant utilisé ARITI

Nous présentons dans cette annexe, les adresses et les noms Internet des sites clients dans le monde ayant utilisés le système ARITI entre juillet et fin septembre 2000.

C.1 En Europe

Adresse du site distant	Nom du site distant
- France -	- France -
213.36.112.69	dyn-213-36-112-69.ppp.libertysurf.fr
195.132.110.53	r110m53.cybercable.tm.fr
131.254.40.59	castor.irisa.fr
131.254.40.52	semnon.irisa.fr
164.138.17.133	Toulouse-17-133.abo.wanadoo.fr
137.121.41.13	pc-41-13.inrets.fr
— Espagne —	— Espagne —
62.36.142.66	usuario1-36-142-66.dialup.uni2.es
62.36.131.230	usuario1-36-131-230.dialup.uni2.es
62.36.133.138	usuario1-36-133-138.dialup.uni2.es
— Italie —	— Italie —
194.119.205.166	ba.cnr.it
139.191.160.68	pert28.jrc.it
- Royaume Uni-	- Royaume Uni -
62.137.116.152	modem-152.pigeon.dialup.pol.co.uk
212.111.142.8	user142008.dial.netline.net.uk
- Grèce -	- Grèce -
143.233.3.18	iris.iit.demokritos.GR
195.242.129.18	borealis.compulink.gr
- Finlande -	- Finlande -
212.38.227.18	free-1-18.dyn.nic.fi

TAB. C.8 – *Quelques adresses Internet des sites clients en Europe ayant utilisé ARITI*

C.2 Dans le monde

Adresse du site distant	Nom du site distant
- Argentine -	- Argentine -
200.45.52.199	host052199.arnet.net.ar
200.47.57.181	line181.comsat.net.ar
200.47.57.133	line133.comsat.net.ar
200.42.105.46	a200042105046.rev.prima.com.ar
200.32.56.184	rqta-1-184.trcnet.com.ar
200.42.136.184	a200042136184.rev.prima.com.ar
209.13.191.122	AVE2ppp-890.uc.infovia.com.ar
200.51.38.140	modem140-as12.capfed1.sinectis.com.ar
216.244.209.48	modem48-as4.capfed2.sinectis.com.ar
200.10.117.168	ppp-117-168.movi.com.ar
216.244.207.59	modem59-tc11.capfed1.sinectis.com.ar
200.42.146.89	a200042146089.rev.prima.com.ar
216.244.208.8	modem8-as3.capfed2.sinectis.com.ar
200.47.72.219	line219.comsat.net.ar
200.45.52.199	host052199.arnet.net.ar
200.42.105.46	a200042105046.rev.prima.com.ar
200.16.135.225	h200016135225.ssd.net.ar
- Brésil -	- Brésil -
200.135.24.31	maverick.furb.rct-sc.br
- Australie -	- Australie -
203.12.148.119	as1-p119.ncle.hunterlink.net.au
- Mexique -	- Mexique -
148.233.186.36	du-148-233-186-36.prodigy.net.mx
- Canada -	- Canada -
132.207.88.24	internet-pub2.biblio.polymtl.ca

TAB. C.9 – Quelques adresses Internet des sites clients dans le monde ayant utilisé ARITI

Adresse du site distant	Nom du site distant
- .net -	- .net -
38.30.134.121	ip121.baltimore20.md.pub-ip.psi.net
63.38.44.161	1Cust161.tnt1.san-francisco3.ca.da.uu.net
209.244.215.139	dialup-209.244.215.139.Washington2.Level3.net
208.5.13.64	cols20851364.cols.net
208.37.59.59	w059.z208037059.lax-ca.dsl.cnc.net
216.230.1.204	ip204.vcu4.richmond.i-c.net
62.158.248.216	p3E9EF8D8.dip.t-dialin.net
63.225.232.20	pppdslj20.slkc.uswest.net
24.131.171.151	el08-24-131-171-151.ce.mediaone.net
208.37.14.250	w250.z208037014.sjc-ca.dsl.cnc.net
38.31.170.115	ip115.salt-lake-city12.ut.pub-ip.psi.net
141.155.104.21	adsl-141-155-104-21.bellatlantic.net
- .edu -	- .edu -
141.211.31.173	141-211-31-173.bus.umich.edu
132.194.22.50	nc2608-50.cudenver.edu
35.9.38.30	walther.egr.msu.edu
128.2.179.206	MARYSMACHINE.SPEECH.CS.CMU.EDU
- .com -	- .com -
199.174.226.92	user-33qtois.dialup.mindspring.com
199.174.227.69	user-33qtoq5.dialup.mindspring.com
4.54.14.160	PPPa35-ResaleGastonia1-2R7135.saturn.bbn.com
24.93.38.185	cs9338-185.austin.rr.com
204.210.133.137	roc-204-210-133-137.rochester.rr.com
24.93.38.185	338-185.austin.rr.com
63.64.166.69	minnow019.mapletronics.com
209.79.29.29	209-79-29-29.max-tnt-01.simi.ca.us.cnmnetwork.com
207.226.216.177	stpm3-4-177.olg.com
- autres -	- autres -
130.240.35.111	xi111.sm.luth.se
130.240.35.94	xi094.sm.luth.se
193.10.62.197	unnamed.kmh.se
193.231.207.101	ppp96.dnttm.ro
193.193.217.132	murka.kot.poltava.ua

TAB. C.10 – suite : Quelques adresses Internet des sites clients dans le monde ayant utilisé ARITI

D Les différents systèmes utilisant Internet comme moyen de contrôle

Nous présentons ici une liste des différents systèmes utilisant Internet comme moyen de contrôle. Cette liste est hébergée par la NASA à l'adresse :

http://ranier.oact.hq.nasa.gov/telerobotics_page/realrobots.html



ARITI- Augmented Reality Interface for Telerobotic applications via Internet - control a 4-DOF robot using a remote computer

Australian Telerobot - remotely operate a 6-DOF ASEA manipulator located at the University of Western Australia. Believed to be the first remotely operated industrial robot on the Web!

Bradford Robotic Telescope - robotic operation of an automated telescope

CSC Telerobot - remotely operate a 6-DOF manipulator located at the Carnegie Science Center in Pittsburgh.

Drinking Maiden Exhibition - telerobotic examination of The Drinking Maiden by Ernst Wenck

Eyebot Project - control a manipulator holding a webcam - and read about their great no-budget implementation!

iINTERFACE - (formerly known as Doppelganger) an attempt to create a telerobotic machine that is controllable through the web which will eventually be displayed in an art context

Interactive Model Railroad - remote operation of a real train set in Germany! Is it a robot? It depends on your definition...

Interfacing Reality - remotely fly a robotic blimp (currently offline)

Jason - use a web-connected mobile robot to solve an online mystery. Part of the Jason Project

Khep On The Web - a Khepera robot controlled from your web browser, with streaming video

Legal Tender - a telerobotic laboratory to investigate counterfeit currency

Meachanical Gaze - allows multiple remote users to actively control up to six degrees of freedom (DOF) of a robot arm with an attached camera to explore a collection of physical museum exhibits

Mercury project - robotic tele-excavation of "Southwest Nevada" over the Internet (currently inactive, but the historical logs are great!)

Net-Robot - a net-controllable manipulator in Germany, which you can watch solve the Towers-Of-Hanoi problem

PumaPaint - use a Puma robot to paint with brushes, paint and easel - create real art!

Rhino - let your robotic tourguide show you the Deutsches Museum Bonn (only active for short periods)

Robotic Garden - robotic plant tending and maintenance over the Internet. If you water the robot, will it grow up to be a big robot?

RoboToy - control a 5-DOF manipulator at the University of Wollongong to pick up pieces of Australian styrofoam! In a very neat twist, you can control either the real robot, or a JAVA simulation!

Schoolnet - Robotics Centre WebCam with web-driven remote pan and tilt pointing

UC Santa Barbara Remotely Operated Telescope - the Remote Access Astronomy Project invites you to access their Remotely Operated Telescope and submit requests

University of Ballarat Telerobot - internet operation of a robot manipulator

V-Car - remotely drive R/C cars via the web, with a very interesting interface

Xavier - an Internet-guided mobile robot currently wandering around Carnegie Mellon University

Résumé :

La téléprésence, ou la présence virtuelle, devient de plus en plus courante grâce à l'évolution technologique, impliquant une ouverture du potentiel du télétravail. Ce dernier est fortement lié à la virtualité et à la téléprésence, donc aussi à la robotique et à l'informatique, autant qu'aux communications. Cette thèse étudie les méthodes et les outils nécessaires à la réalisation d'un système de télétravail via Internet.

Dans un premier temps, nous présentons les tendances technologiques qui ont contribué à l'évolution du télétravail. Ensuite, nous nous intéressons à la téléopération et la télérobotique, nous verrons comment les techniques de la réalité virtuelle et augmentée ont contribué à percer certains verrous, liés généralement à la distance qui sépare les deux sites maître et esclave. Dans un deuxième temps, nous nous intéressons à l'assistance au télétravail via Internet. Nous étudions les méthodes et les outils nécessaires à la réalisation des guides virtuels portables et paramétrables en proposant ainsi le formalisme implanté.

Dans un troisième temps, nous étudions les méthodes utiles pour un contrôle en réalité augmentée (modélisation d'environnement et calibration de la caméra et du robot). Nous présentons la méthode retenue pour le télétravail via Internet. Ensuite, nous présentons notre système expérimental de télétravail baptisé ARITI (Augmented Reality Interface for Telerobotic applications via Internet). Dans un quatrième temps, nous présentons deux applications, une pour la télémanipulation et l'autre, pour la téléopération d'un robot mobile.

Enfin, dans un cinquième temps, Nous évaluons quelques tâches de téléopération ainsi que les méthodes de compression d'image vidéo utilisées. Nous montrons d'une part, la stabilité du serveur du système ARITI vis à vis des différentes connexions, locale, moyenne et grande distance. D'autre part, comment le retour prédictif distribué facilite à plusieurs opérateurs de télétravailler en coopération.

Mots-clé : télétravail, travail coopératif, télérobotique, réalité virtuelle, réalité augmentée, Internet

Abstract:

Telepresence or virtual presence has become more and more commonplace thanks to evolution of technology which implies an opening to the telework capacities. Telework is essentially linked to virtuality and telepresence and consequently linked to robotics and computing as well as communications. This work deals with the methods and the tools necessary to the achievement of a telework system via Internet.

We have first presented, the technologies which has contributed to the advance of the telework. We will then focus on teleoperation and telerobotics and we will realize how virtual reality technologies contribute to cope with some problems due to the distance between the master and the slave sites. Secondly, our interest will focus on telework assistance via Internet. The methods and tools necessary to the achievement of transportable and configurable virtual fixtures are studied and a formalism is proposed.

Thirdly, the useful methods for augmented reality control are studied (environment modeling, camera and robot calibration), then the selected method for the telework via Internet is presented. Then, the description of the experimental telework system namely ARITI (Augmented Reality Interface for Telerobotic applications via Internet) is given. Fourthly, Two applications are presented, one is about telemanipulation tasks, the other one is about teleoperation of the mobile robot.

Finally, we will estimate the efficiency of some teleoperation tasks and video images compression methods. On the one hand, we will see the efficiency of the ARITI system with multiple connections, locale, medium and high distance. On the other hand, we will see how the distributed predictive display makes it easy for different operators to work together in cooperative mode.

Keywords: telework, cooperative work, telerobotic, virtual reality, augmented reality, Internet