



**HAL**  
open science

## Simulation graphique d'un robot bipède dans un environnement structuré

Laure France

► **To cite this version:**

Laure France. Simulation graphique d'un robot bipède dans un environnement structuré. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1999. Français. NNT : . tel-00004831

**HAL Id: tel-00004831**

**<https://theses.hal.science/tel-00004831>**

Submitted on 18 Feb 2004

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Simulation Graphique d'un Robot Bipède dans un Environnement Structuré

---

Laure FRANCE

Thèse préparée à l'INRIA et à iMAGIS-GRAVIR/IMAG-INRIA et présentée  
pour obtenir le titre de :

**Docteur de l'Université Joseph Fourier**  
**discipline informatique**

*Arrêté ministériel du 5 juillet 1984 et du 30 mars 1992*

Soutenue le 13 Octobre 1999 devant la commission d'examen :

Bruno	ARNALDI	Président
Ronan	BOULIC	Rapporteur
Michiel	VAN DE PANNE	Rapporteur
Joëlle	COUTAZ	Examinateur
Bernard	ESPIAU	Directeur
Claude	PUECH	Directeur
Alain	GIRAULT	Invité
Jean-Dominique	GASCUEL	Invité



# Remerciements

Je tiens à remercier tout d’abord Bernard Espiau pour m’avoir accueillie dans son projet BIP et de m’avoir donné le sujet de ma thèse. Je le remercie aussi pour toutes les discussions que l’on a pu avoir à ce sujet.

Je remercie également Claude Puech d’avoir accepté de me co-encadrer et de m’avoir accueillie à iMAGIS. Je le remercie tout particulièrement pour tous les conseils avisés qu’il m’a fournis à chaque fois que j’en avais besoin. En effet, malgré son emploi du temps très chargé, il a toujours su se montrer disponible.

Je tiens à remercier les membres extérieurs de mon jury, Joëlle Coutaz et Bruno Arnaldi, d’avoir accepté de participer à ma thèse. Et tout particulièrement Ronan Boulic et Michiel van de Panne, pour avoir accepté l’ingrate tâche d’être rapporteurs de ma thèse.

Je remercie également les autres permanents des deux projets, en particulier Alain qui a eu la gentillesse de m’encadrer, tout en n’étant pas “du domaine” et Jean-Dominique pour tous les miracles sur un ordinateur qu’il a réalisés. Je tiens également à remercier Marie-Paule pour ses conseils scientifiques, George pour les conversations culturelles et enfin, mais non des moindres, François pour toutes les discussions que l’on a pu avoir ensemble, aussi bien scientifiques que culturelles, philosophiques et personnelles.

Contrairement à ce robot qui a fait l’objet de mon étude, je suis humaine et mon moral en a souvent pris un coup. Contrairement à ce robot, je suis tombée et je me suis relevée. En particulier, grâce à Céline qui était toujours présente, fidèle à son poste de “camarade de bureau”, mais aussi et surtout en tant que véritable amie. Elle m’a non seulement guidée et conseillée pour ma thèse, mais elle m’a aussi écoutée et soutenue. Sans son dynamisme et sa gentillesse, je ne serais peut-être pas parvenue à finir cette thèse. Je lui suis très reconnaissante pour tout ce qu’elle a fait pour moi et je ne saurais jamais assez l’en remercier.

Je tiens également à remercier Lydie, qui a été tout aussi présente pour moi. À la fois petite sœur, confidente, conseillère, elle s’est toujours montrée franche, sincère, tout en restant d’une extrême gentillesse. J’ai apprécié et abusé de sa bonté et de sa générosité, je n’oublierai jamais tous les moments privilégiés passés avec elle et Philippe.

Je remercie également toutes les personnes de mes deux laboratoires d’accueil. En particulier, merci à Éric, Gilles, Frédo, Nico, Eugénia, Fabrice, FFT, Joëlle, Dan, Rosine, Jean-

LucΓBilalΓAhmedΓPierre-BriceΓFranckΓRaphaëlΓSylvainΓVincent. Et le manquant ...



J'aimerais aussi remercier tous les enseignants avec lesquels j'ai effectué mes enseignementsΓen particulier Jean-Marc Vincent. Je remercie tout autant mes étudiants. Ils m'ont permis de me révélerΓà travers l'expérience d'enseignement.

J'aimerais terminer en dédiant cette thèse à toutes les personnes qui ont un rêveΓmais dont le chemin est semé d'embûchesΓet quiΓmalgré toutΓont réussi à trouver le courage au fond d'elles-mêmes de persévérerΓde continuer. J'imagine aisément ce qu'on peut vivreΓendurer. L'important est de ne pas abandonnerΓd'avoir confiance en soiΓfoi en soi. Car le but atteint est la meilleure des récompenses. Voilà l'expérience qu'offre une grande sœur à sa petite sœur et son petit frère. Maud et JulienΓc'est à vous tout particulièrement que je dédie cette thèse.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>15</b>
1.1	La synthèse d'images . . . . .	15
1.1.1	Conception assistée par ordinateur . . . . .	16
1.1.2	Divertissements . . . . .	16
1.1.3	Visualisation scientifique . . . . .	16
1.1.4	Animation et Simulation . . . . .	17
1.2	Le robot anthropomorphe . . . . .	17
1.2.1	Motivations . . . . .	18
1.2.2	Applications des robots bipèdes . . . . .	18
1.2.3	Conception du robot BIP . . . . .	19
1.3	Thème de la thèse . . . . .	19
1.4	Contenu du mémoire . . . . .	20
<b>2</b>	<b>Marche d'un robot anthropomorphe</b>	<b>21</b>
2.1	Introduction . . . . .	21
2.2	Modèle du robot . . . . .	21
2.2.1	Notation Denavit-Hartenberg . . . . .	21
2.2.2	Représentation du robot . . . . .	22
2.3	Contrôle par fonctions de tâches . . . . .	23
2.4	Étude biomécanique de la marche humaine . . . . .	25
2.4.1	Concepts de base . . . . .	25
2.4.2	Phases de la marche . . . . .	28
2.4.3	Périodes lors de la marche . . . . .	29
2.4.4	Paramètres de la marche . . . . .	31
2.5	Conclusion . . . . .	32
<b>3</b>	<b>État de l'art : Synthèse d'un mouvement de marche</b>	<b>33</b>
3.1	Introduction . . . . .	33
3.2	Méthodes cinématiques . . . . .	34
3.2.1	Définitions . . . . .	34
3.2.2	Animation par cinématique directe . . . . .	35
3.2.3	Animation par cinématique inverse . . . . .	40
3.2.4	Combinaison des deux cinématiques . . . . .	41
3.2.5	Discussion . . . . .	42
3.3	Méthodes incorporant la dynamique . . . . .	42

3.3.1	Définitions . . . . .	42
3.3.2	Animation par dynamique directe . . . . .	44
3.3.3	Animation par dynamique inverse . . . . .	46
3.3.4	Animation hybride cinématique et dynamique . . . . .	47
3.3.5	Discussion . . . . .	57
3.4	Utilisation de mouvements capturés . . . . .	57
3.4.1	Capture de mouvements . . . . .	58
3.4.2	Déformations de mouvement . . . . .	60
3.4.3	Fondu de mouvements . . . . .	63
3.5	Conclusion . . . . .	66
<b>4</b>	<b>Mouvements de référence pour le robot</b>	<b>69</b>
4.1	Introduction . . . . .	69
4.2	Adaptation de mouvements capturés pour l'obtention d'un cycle de marche	69
4.2.1	Transformation en coordonnées articulaires . . . . .	70
4.2.2	Détermination du cycle . . . . .	70
4.2.3	Longueur des segments . . . . .	71
4.2.4	Mouvement de la racine de la structure arborescente . . . . .	72
4.2.5	Discussion . . . . .	72
4.3	Synthèse de trajectoires transitoires . . . . .	73
4.3.1	Méthode . . . . .	75
4.3.2	Détermination des positions extrêmes . . . . .	75
4.3.3	Durée de transition basée sur les accélérations . . . . .	76
4.3.4	Durée de transition basée sur des contraintes géométriques . . . . .	78
4.3.5	Application au mouvement de marche . . . . .	79
4.3.6	Discussion . . . . .	82
4.4	Conclusion . . . . .	84
<b>5</b>	<b>Perception et reconnaissance de l'environnement par un robot réel</b>	<b>87</b>
5.1	Introduction . . . . .	87
5.2	Interactions avec l'environnement en robotique . . . . .	88
5.2.1	Le robot . . . . .	88
5.2.2	Perception de l'environnement . . . . .	88
5.2.3	Commande adaptative vis-à-vis de l'environnement . . . . .	89
5.2.4	Identification des besoins en capteurs . . . . .	89
5.3	Capteurs . . . . .	90
5.3.1	Rôle et fonctions . . . . .	90
5.3.2	Classification . . . . .	90
5.3.3	Discussion . . . . .	91
5.4	Capteurs proximétriques . . . . .	92
5.4.1	Physique du capteur . . . . .	92
5.4.2	Propriétés des surfaces . . . . .	93
5.4.3	Principes de détection . . . . .	94
5.4.4	Caractéristiques des capteurs optiques et acoustiques . . . . .	98
5.4.5	Discussion . . . . .	99
5.5	Conclusion . . . . .	100

<b>6</b>	<b>Modélisation de la perception</b>	<b>101</b>
6.1	Introduction . . . . .	101
6.2	Modèles de capteurs . . . . .	102
6.2.1	Outils graphiques . . . . .	102
6.2.2	Modélisation des signaux capteurs . . . . .	105
6.2.3	Simulation des capteurs . . . . .	112
6.2.4	Discussion . . . . .	118
6.3	Analyse des données capteurs . . . . .	119
6.3.1	Courbes de distances . . . . .	119
6.3.2	Profil de l'environnement . . . . .	119
6.3.3	Proposition d'algorithme de reconnaissance . . . . .	123
6.4	Validation . . . . .	130
6.4.1	Capteur optique . . . . .	134
6.4.2	Capteur à ultra-sons . . . . .	134
6.4.3	Discussion . . . . .	134
6.5	Conclusion . . . . .	134
<b>7</b>	<b>Applications</b>	<b>139</b>
7.1	Introduction . . . . .	139
7.2	Détermination du placement des capteurs . . . . .	140
7.2.1	Hypothèses de travail . . . . .	140
7.2.2	Outil d'analyse : la carte de projection . . . . .	142
7.2.3	Analyse de plusieurs placements . . . . .	143
7.2.4	Discussion . . . . .	148
7.3	Déclenchement des trajectoires transitoires . . . . .	150
7.3.1	Contrôle de haut niveau . . . . .	150
7.3.2	Simulation . . . . .	151
7.4	Conclusion . . . . .	152
<b>8</b>	<b>Conclusion</b>	<b>155</b>
8.1	Bilan . . . . .	155
8.1.1	Robot virtuel . . . . .	155
8.1.2	Perception virtuelle . . . . .	156
8.1.3	Applications . . . . .	158
8.1.4	Discussion . . . . .	158
8.2	Perspectives . . . . .	158
8.2.1	Mouvements de référence divers et variés . . . . .	158
8.2.2	Vue améliorée . . . . .	159
8.2.3	Vers l'autonomie du robot . . . . .	159
8.2.4	Exploitation du simulateur . . . . .	160
<b>A</b>	<b>Modèle d'éclairage</b>	<b>161</b>
A.1	Interaction lumière/objet en synthèse d'images . . . . .	161
A.2	Théorie : Intensité et Énergie . . . . .	162
A.2.1	Réflectivité spectrale bidirectionnelle . . . . .	164
A.2.2	Surfaces diffuses et spéculaires . . . . .	165



---

A.3	Application : Le modèle de réflexion de Phong . . . . .	165
<b>B</b>	<b>Perception et reconnaissance de formes</b>	<b>169</b>
B.1	Introduction . . . . .	169
B.1.1	Fusion multi-capteurs . . . . .	169
B.1.2	Reconstruction de formes . . . . .	171
B.1.3	Reconnaissance d'objets . . . . .	172
B.2	Conclusion . . . . .	173
<b>C</b>	<b>Tests des configurations de capteurs sur le robot</b>	<b>175</b>
C.1	Positions, orientations et portées des capteurs . . . . .	175
C.1.1	Jambe . . . . .	176
C.1.2	Cuisse . . . . .	179
C.2	Nombre de capteurs . . . . .	182
C.2.1	Configuration à 1 capteur par segment . . . . .	182
C.2.2	Configurations à 3 capteurs par segment . . . . .	182
C.2.3	Configurations à 5 capteurs par segment . . . . .	182
C.2.4	Choix . . . . .	182

# Table des figures

2.1	<i>Notation Denavit-Hartenberg.</i>	22
2.2	<i>Le squelette de notre robot, en notation Denavit-Hartenberg.</i>	23
2.3	<i>Plans de référence du corps humain dans la position anatomique standard.</i>	26
2.4	<i>Démarche vue dans les trois plans.</i>	27
2.5	<i>Projection des trois mouvements plans pour obtenir la démarche 3D.</i>	27
2.6	<i>Cycle de marche normal pour un garçon de huit ans.</i>	28
2.7	<i>Temps passé sur chaque jambe pendant un cycle de marche normale.</i>	28
2.8	<i>Nomenclature traditionnelle pour la description des huit principales périodes.</i>	29
2.9	<i>Empreintes fournissant des paramètres de distance intéressants.</i>	31
3.1	<i>Une itération par cinématique.</i>	35
3.2	<i>Diverses étapes de la démarche.</i>	36
3.3	<i>Interpolation d'une rotation.</i>	38
3.4	<i>Différentes interpolations.</i>	39
3.5	<i>Problèmes liés à la cinématique inverse.</i>	41
3.6	<i>Exemples de tâches.</i>	42
3.7	<i>Balle de masse <math>m</math> animée.</i>	44
3.8	<i>Exemple de contrainte : contrainte point-point.</i>	45
3.9	<i>Combinaison de dynamique directe et dynamique inverse.</i>	46
3.10	<i>Contraintes spatio-temporelles.</i>	48
3.11	<i>Contraintes spatio-temporelles : définition de postures.</i>	48
3.12	<i>Contrôleur avec boucle de rétro-action.</i>	51
3.13	<i>Algorithme pour l'animation par contrôleur.</i>	51
3.14	<i>Exemple de graphe de contrôle de poses pour la marche humaine.</i>	52
3.15	<i>Contrôle du cycle limite.</i>	57
3.16	<i>Processus de capture de mouvements.</i>	59
3.17	<i>Système de capture de mouvements.</i>	60
3.18	<i>Courbes des mouvements original et déformé.</i>	61
3.19	<i>Courbe de positions définie dans l'espace de positions.</i>	64
3.20	<i>Fondu d'un mouvement de marche et d'un mouvement de smash.</i>	65
4.1	<i>Calcul des angles par la méthode des tangentes.</i>	71
4.2	<i>Cycle de marche pour le robot bipède.</i>	72
4.3	<i>Postures du robot bipède.</i>	74
4.4	<i>Mouvement du pied libre pendant la trajectoire transitoire de départ.</i>	80
4.5	<i>Mouvement transitoire de départ du robot.</i>	81

4.6	<i>Mouvement du pied libre pendant la trajectoire transitoire d'arrêt.</i>	83
4.7	<i>Mouvement transitoire d'arrêt du robot.</i>	84
5.1	<i>Fonctions de base d'un robot.</i>	88
5.2	<i>Schéma de fonctionnement d'un capteur.</i>	89
5.3	<i>Types de base de surfaces.</i>	93
5.4	<i>Modèle par micro-facette de la surface de réflexion.</i>	93
5.5	<i>Réflexion comme fonction de la rugosité de la surface.</i>	94
5.6	<i>Réflexion d'onde simple.</i>	94
5.7	<i>Réflexion multiple d'onde.</i>	95
5.8	<i>Principe par mesure du temps de vol.</i>	95
5.9	<i>Mesure de la différence de phase entre le signal émis et celui reçu.</i>	96
5.10	<i>Mesure par triangulation.</i>	97
5.11	<i>Forme typique de rayon ultra-son à 50 kHz.</i>	98
5.12	<i>Lobes principaux d'émissions pour des capteurs à ultra-sons.</i>	98
6.1	<i>Caméra perspective.</i>	102
6.2	<i>Définition du rayon pour l'action de sélection.</i>	104
6.3	<i>Types de perception.</i>	105
6.4	<i>Calcul de la distance à l'obstacle.</i>	106
6.5	<i>Calcul de l'enveloppe pour définir la zone de détection du capteur 3D.</i>	107
6.6	<i>Atténuation du signal en fonction de la distance.</i>	108
6.7	<i>Importance de l'incidence dans la réflexion du rayon par l'obstacle.</i>	109
6.8	<i>Modèle d'éclairage.</i>	110
6.9	<i>Couleurs du bruit.</i>	112
6.10	<i>Éléments de l'interface, pour définir les capteurs à placer sur le robot.</i>	114
6.11	<i>Visualisation du rayon modélisant l'onde émise par un capteur 1D.</i>	115
6.12	<i>Visualisation de l'image de la scène vue par un capteur 3D.</i>	115
6.13	<i>Le signal du capteur de réflectance est fonction de l'angle d'incidence.</i>	117
6.14	<i>Exemples de courbes de distances idéales.</i>	120
6.15	<i>Autres exemples de courbes de distances idéales.</i>	121
6.16	<i>Profil parfait calculé à partir de la courbe de distances de la Figure 6.14 (b).</i>	123
6.17	<i>Objets à reconnaître.</i>	124
6.18	<i>Objets non reconnus.</i>	125
6.19	<i>Quelques exemples de profils possibles d'obstacles détectés.</i>	127
6.20	<i>Automate reconnaisseur des cinq types d'objets désirés.</i>	128
6.21	<i>Reconnaissance au fur et à mesure d'un obstacle.</i>	128
6.22	<i>Le robot en présence d'un grand trou.</i>	130
6.23	<i>Profil détecté lors de la simulation.</i>	131
6.24	<i>Courbe de distances lorsque le robot est face à un grand trou.</i>	131
6.25	<i>Capteur optique KM1/AKM (Matsushita).</i>	132
6.26	<i>Capteur à ultra-sons SENSOPART.</i>	132
6.27	<i>Cône de détection du capteur à ultra-sons réel SENSOPART.</i>	133
6.28	<i>Orthèse du genou utilisée pour les tests.</i>	133
6.29	<i>Enregistrements des sorties capteurs optiques.</i>	135
6.30	<i>Enregistrements des sorties capteurs à ultra-sons.</i>	136

6.31	<i>Enregistrements des sorties capteurs à ultra-sons (2).</i>	137
7.1	<i>Espace à détecter, situé devant le robot et de dimensions celles du robot.</i>	142
7.2	<i>Carte de projection.</i>	143
7.3	<i>Première scène modélisée : salle à manger.</i>	144
7.4	<i>Deuxième scène modélisée : partie de bâtiment.</i>	144
7.5	<i>Meilleure configuration canonique de capteurs sur le robot.</i>	148
7.6	<i>Meilleure configuration finale de capteurs sur le robot.</i>	149
7.7	<i>Algorithme de contrôle du robot.</i>	151
7.8	<i>Simulation complète du robot dans l'environnement.</i>	153
8.1	<i>Photographies du robot anthropomorphe réel.</i>	156
A.1	<i>Conventions utilisées en réflexion.</i>	163
A.2	<i>Intensité mesurée.</i>	163
A.3	<i>Réciprocité en lancer de rayons.</i>	164
A.4	<i>Types de surfaces.</i>	165
A.5	<i>Modèle de réflexion de Phong.</i>	166
A.6	<i>Intensité lumineuse en un point.</i>	167
B.1	<i>Reconstruction de formes à partir de différents points de vue.</i>	171
B.2	<i>Reconstruction de formes à partir de l'ensemble des rayons lancés.</i>	172
B.3	<i>Reconstruction de formes à partir de sections planes.</i>	173
C.1	<i>Capteur au milieu du support, à <math>90^\circ</math>, et une portée de 1.5m.</i>	177
C.2	<i>Capteur au milieu du support, à <math>90^\circ</math>, et une portée de 2m.</i>	177
C.3	<i>Capteur au milieu du support, à <math>80^\circ</math>, et une portée de 2m.</i>	177
C.4	<i>Capteur au milieu du support, à <math>90^\circ</math>, et une portée de 2m.</i>	178
C.5	<i>Capteur au milieu du support, à <math>90^\circ</math>, et une portée de 3m.</i>	178
C.6	<i>Capteur au milieu du support, à <math>80^\circ</math>, et une portée de 3m.</i>	178
C.7	<i>Capteur au milieu du support, à <math>90^\circ</math>, et une portée de 2m.</i>	180
C.8	<i>Capteur au milieu du support, à <math>75^\circ</math>, et une portée de 2m.</i>	180
C.9	<i>Capteur au milieu du support, à <math>60^\circ</math>, et une portée de 2m.</i>	180
C.10	<i>Capteur au milieu du support, à <math>90^\circ</math>, et une portée de 2m.</i>	181
C.11	<i>Capteur au milieu du support, à <math>75^\circ</math>, et une portée de 2m.</i>	181
C.12	<i>Capteur au milieu du support, à <math>60^\circ</math>, et une portée de 3m.</i>	181
C.13	<i>Configuration à 1 capteur par support.</i>	183
C.14	<i>Configuration à 3 capteurs par support.</i>	183
C.15	<i>Configuration à 5 capteurs par support.</i>	183
C.16	<i>Carte de projection de la configuration de la Figure C.13.</i>	184
C.17	<i>Carte de projection de la configuration de la Figure C.14 (<math>20^\circ</math>).</i>	185
C.18	<i>Carte de projection de la configuration de la Figure C.14 (<math>30^\circ</math>).</i>	185
C.19	<i>Carte de projection de la configuration de la Figure C.14 (<math>40^\circ</math>).</i>	185
C.20	<i>Carte de projection de la configuration de la Figure C.15 (<math>20^\circ</math>).</i>	186
C.21	<i>Carte de projection de la configuration de la Figure C.15 (<math>30^\circ</math>).</i>	186
C.22	<i>Carte de projection de la configuration de la Figure C.15 (<math>40^\circ</math>).</i>	186



*Je viens de rencontrer un marionnettiste ...  
Il est tout bouleversé!  
Hier soir, il avait laissé sa marionnette côté cour.  
Il l'a retrouvée ce matin côté jardin.  
Ça signifie quoi ?  
Ça signifie que dans la nuit,  
La marionnette a traversé la scène toute seule!  
Elle a dû faire ses premiers pas sans l'aide de personne!  
Sans quelqu'un derrière elle pour la manipuler!  
C'est émouvant, non ?  
Ça ne fait pas l'affaire d'un marionnettiste, hé!  
Si sa marionnette se met à marcher toute seule ...  
C'est son boulot qui fout le camp!*

Raymond DevosΓ  
"Le montreur de marionnettes"



# Chapitre 1

## Introduction

Les travaux menés dans cette thèse visent à simuler le mouvement d'un robot bipède dans un environnement structuréΓaménagé pour les humains. Au-delà de la dynamique de la marche et de la résolution de l'équilibreΓun problème majeur est de permettre au robot de percevoir cet environnementΓnotamment les obstaclesΓde manière efficaceΓrapide et sûre.

La simulation informatique offre cet avantage de pouvoir attaquer la résolution d'un problème avant même d'avoir construit le mécanisme qui fait l'objet de l'étudeΓet même avant de fixer tous les choix technologiques. Cela doit permettreΓau moment de la réalisation concrèteΓd'obtenir rapidement un robot fonctionnelΓet surtout de ne pas faire courir de risque grave à un matériel coûteux.

AinsiΓces travaux se situent au carrefour de la synthèse d'images et de la robotiqueΓle premier domaine apportant un cadre à la simulation d'environnement et à la simulation de la marcheΓet le second constituant le contexte réel à la source des spécifications et à l'issue de la validation. Nous allons donc introduire ces deux domainesΓavant de détailler les objectifs de la thèse et l'organisation de ce mémoire.

### 1.1 La synthèse d'images

La synthèse d'images est devenue une technologie de choix pour communiquer des idéesΓdes données et des tendances dans des domaines variés tels que le commerceΓles sciencesΓl'ingénierie et l'éducation. La création d'une communication graphique qui soit efficaceΓrepose principalement sur la modélisation des objets dont nous souhaitons produire les images. Grâce à la synthèse d'imagesΓnous pouvons créer des *réalités artificielles* ou *virtuelles*Γque nous pouvons explorer. Il nous est alors possible d'examiner et de contrôler les objets et les phénomènes de manière naturelle et intuitiveΓen exploitant notre habileté très développée pour la reconnaissance de formes visuelles.

La synthèse d'images offre le moyen le plus important pour produire des imagesΓdepuis l'invention de la photographie et de la télévision. Elle permet en effet de représenter les objets réelsΓmais également des objets virtuelsΓqui peuvent ou non exister dans la réalité.



Ces images offrent un moyen correct pour communiquer de l'information. Il est par ailleurs possible de créer des séquences d'images de synthèse quiΓaffichées à l'écran à une fréquence de plus de 15 images par secondeΓconstitue l'analogie d'un film. L'avantage d'un film est de pouvoir montrer les changements au cours du temps de manière plus graphique et explicite que ne peut le faire une séquence de diapositives. AinsiΓun film en images de synthèse traduit mieux un mouvement ou un changement de forme que des images simples. Pour toutes ces raisonsΓla synthèse d'images est actuellement un outil très intéressant et largement utilisé pour diverses fonctionsΓen particulier pour la conception assistée par ordinateurΓles divertissementsΓla visualisation scientifiqueΓl'animation et la simulationΓque nous exposons maintenant.

### 1.1.1 Conception assistée par ordinateur

Dans de nombreuses tâches de conceptionΓde mise en œuvre et de constructionΓl'information que donnent les images est pratiquement indispensable. En conception assistée par ordinateurΓla synthèse d'images interactive est utilisée pour concevoir des composants et des systèmes mécaniquesΓélectriquesΓélectromécaniques et électroniquesΓmais aussi des structures telles que des immeublesΓdes voituresΓdes avionsΓdes bateauxΓdes systèmes optiquesΓetc. Ce type de représentations permet notamment de tester les propriétés des objets modélisésΓcomme les propriétés structurellesΓélectriques ou thermiques. Une fois les objets conçusΓdes programmes peuvent réaliser des post-traitements pour définir des contrôles numériquesΓpar exemple pour découper ou percer des tôles afin de réaliser des portières de voitures.

### 1.1.2 Divertissements

La synthèse d'images se retrouve aussi dans de nombreux divertissements. Elle permet notamment de rendre les effets spéciaux de plus en plus élégants dans les filmsΓtels que pour "Star Wars"Γ"Abyss"Γ"Jurassic Park"Γ"Terminator 2"Γ"Titanic" et "Matrix"Γet même dans les publicités téléviséesΓcomme pour Coca-Cola ou Perrier. Elle est aussi largement utilisée dans les jeux vidéo. Une autre voie en plein développement concerne les films d'animation. Des systèmes produisent des images intermédiaires par interpolation de deux images spécifiques. En outreΓles personnages sont de plus en plus modélisés par ordinateurΓvia des descriptions de formes 3DΓet dont les mouvements sont contrôlés par des commandes via l'ordinateurΓplutôt que par des dessins à la main réalisés par des graphistes. Certains films d'animation sont même entièrement réalisés avec des images de synthèseΓcomme "Toy Story"Γ"Bugs Life" ou encore "Antz".

### 1.1.3 Visualisation scientifique

La *visualisation scientifique* est devenue un domaine important à la fin des années 80Γlorsque les scientifiques et les ingénieurs ont réalisé qu'ils ne pourraient plus interpréter les énormes quantités de données produites rapidement par les ordinateurs puissantsΓsans faire la synthèse de ces données et mettre en évidence les tendances et les phénomènes au moyen de diverses représentations graphiques. AinsiΓles animations de comportements évolutifs d'objets réels ou simulés par synthèse d'imagesΓdeviennent de plus en plus

populaires dans le domaine de la visualisation scientifique. Nous pouvons nous en servir pour visualiser des entités mathématiques abstraites et des modèles mathématiques de phénomènesΓtels que les fluidesΓla relativitéΓles réactions chimiques et nucléairesΓles systèmes physiologiques et les fonctions organiquesΓles déformations de structures mécaniquesΓetc.

### 1.1.4 Animation et Simulation

*Animer* signifie littéralement “amener à la vie”. Bien que l’animation soit souvent considérée comme synonyme de mouvement en synthèse d’imagesΓelle recouvre en fait tous les changements qui ont un effet visuel. Cela comprend la position qui varie au cours du tempsΓmais aussi la formeΓla couleurΓla transparenceΓla structure et la texture d’un objetΓainsi que les changements d’éclairageΓde techniques de renduΓet les changements de positionΓd’orientation et de focale de la caméra. AinsiΓgrâce à l’animationΓles objets peuvent évoluer par rapport à un observateur immobile. Les objets peuvent aussi rester immobilesΓet c’est l’observateur qui tourne autour d’euxΓfait un zoom d’une partie de la scèneΓcomme s’il regardait au travers d’un viseur d’une caméra vidéo.

Une animation peut être directement produite à partir de modèles ou d’ensembles d’équations qui spécifient le comportement dynamique des objets. Ce type d’animation est qualifié de *simulation*Γet est souvent utilisé pour la visualisation scientifique. Elle est aussi particulièrement utilisée dans le cas des simulateurs : simulateurs de volΓde conduiteΓd’opérations chirurgicalesΓetc. Tous ces simulateurs exploitent le mouvement interactif. Par exempleΓpour les simulateurs de volΓdes environnements sont représentés en se basant sur des lieux réelsΓtels que les aéroports internationaux. Une séquence d’animation est produite en temps réelΓet elle est contrôlée à partir des dispositifs électromécaniques maniés par le “pilote”. L’exigence du temps réel pour réaliser l’interaction avec l’utilisateur contraint généralement l’animation à n’être constituée que de polygones nuancés par un rendu de Gouraud et plaqués de texturesΓau lieu d’objets plus complexes et de rendu plus réaliste.

## 1.2 Le robot anthropomorphe

L’objectif du projet BIP<sup>1</sup> consiste à construire intégralement un robot réel anthropomorpheΓcapable de marcher comme les humains. Il ne s’agit pas de reproduire la morphologie complète des jambes humainesΓqui est une machine extraordinaire comprenant quelques 55 osΓ90 muscles et 16 nerfs. Le but est plutôt d’approcher un comportement anthropomorphique à l’aide d’une version simplifiée des jambes. Le premier prototype est une machine à 17 degrés de libertéΓdotée de deux jambes et d’un pelvisΓet équipée de capteurs internes et externes.

---

<sup>1</sup>BIP est un projet INRIA.

### 1.2.1 Motivations

La mobilité au moyen de pattes possède certains avantages majeurs sur d'autres types de mobilités tels que le mouvement généré par des roues. Par exemple un robot marcheur ne nécessite pas un contact permanent avec le sol et peut donc fonctionner sur un terrain non préparé comportant des ondulations des obstacles à enjamber et des trous. Ainsi de tels robots bipèdes ou multipodes ont une plus faible emprise avec le sol et leur adaptabilité est alors meilleure.

Cependant la technologie multipode doit laisser place en général à la bipédie si nous voulons conserver de bonnes possibilités de déplacement et d'action sans pour autant modifier les environnements de travail du robot dans le cas où ceux-ci étaient conçus au préalable pour l'humain. Ainsi l'intérêt est grandissant pour ces systèmes dits humanoïdes destinés à accompagner l'être humain dans certaines de ses activités personnelles ou professionnelles.

### 1.2.2 Applications des robots bipèdes

Le premier type d'application se situe dans la maintenance d'environnements à risque tels que les centrales nucléaires vieillissantes et les usines chimiques. Ces environnements posent des problèmes de sécurité en raison du risque potentiel lié à un accident. La maintenance de tels environnements est par conséquent extrêmement importante pour permettre un fonctionnement correct et éviter des accidents catastrophiques. Cependant ces environnements étaient initialement conçus pour être entretenus par des travailleurs humains et comportaient des couloirs étroits, tournants, des escaliers, des passerelles etc. C'est précisément un milieu où les robots bipèdes peuvent faire prévaloir leurs avantages, les véhicules à roues et les robots multipodes ne pouvant avoir un libre accès.

Un deuxième type d'application concerne la robotique d'assistance. En effet une certaine forme d'assistance à domicile de personnes à mobilité réduite pour des tâches très routinières pourrait être assurée par des robots bipèdes puisqu'ils ont la faculté de pouvoir se déplacer sans adaptation particulière de l'environnement humain.

Un troisième type d'application consiste à créer une synergie entre la robotique et la biomécanique concernant l'étude de la démarche humaine. La locomotion humaine bien qu'elle ait fait l'objet d'études approfondies n'est pas très bien comprise en particulier en ce qui concerne les traitements à appliquer à des patients en rééducation et à ceux souffrant de maladies comme la paralysie. Un problème majeur est d'effectuer des mesures in-vivo du muscle et des forces conjuguées. Ainsi un simulacre robotique peut être très utile en constituant un excellent banc d'essais. Il permet d'une part de réaliser des mesures précises de diverses quantités dans des conditions faciles et d'autre part de tester des théories variées relatives à la locomotion humaine aboutissant à des progrès dans la rééducation et également dans l'amélioration de la conception de chaussures, d'orthèses et de prothèses. D'autre part les modèles de mouvements humains obtenus peuvent s'avérer particulièrement utiles en biomécanique pour réaliser des prédictions post-opératoires de chirurgies du squelette ou des muscles.

### 1.2.3 Conception du robot BIP

La difficulté de conception d'un robot marcheur bipède résulte des contraintes spécifiques au principe même de la marche bipède. En effet, le système doit comporter un nombre élevé d'articulations dans un volume réduit. De plus, ce système doit en permanence être en équilibre, soit statique, soit dynamique. Des couples instantanés élevés sont alors nécessaires, et doivent être, de plus, correctement répartis sur l'ensemble des articulations. Aussi se pose le problème d'une gestion globale et dynamique de la puissance nécessaire. Le phénomène de la marche est un processus très complexe, et pourtant, nous l'effectuons avec une aisance et une grâce étonnante, et de plus inconsciemment, après un long apprentissage pendant l'enfance. C'est seulement lorsque l'on se lance dans la construction d'un robot bipède, que l'on prend conscience de la capacité d'adaptation et d'harmonie dont la nature nous a doté.

Un robot bipède peut être considéré comme une chaîne articulée arborescente de corps rigides dans l'espace tridimensionnel. La marche est caractérisée par différentes phases, elles-mêmes fonctions d'événements affectant le centre de gravité et les pieds à cause des contacts au sol. Le modèle mécanique d'un tel système comporte trois aspects : une dynamique de corps rigides, définie par des équations physiques ; un ensemble de contraintes inégalités, qui est variable selon les phases, et qui détermine ainsi un mode de fonctionnement, en traduisant l'existence de contacts sans pénétration et sans glissement à respecter ; un choix de loi d'impact modélisant les transitions entre les modes de fonctionnement. Aussi, ce système dynamique hybride apparaît-il comme complexe, et son étude est encore peu explorée.

## 1.3 Thème de la thèse

Notre but est de créer un simulateur du robot bipède. D'une part, le robot réel n'est pas encore complètement construit, et plusieurs variantes pourront être testées sur le simulateur, sans pour autant toutes les construire. D'autre part, il s'agit d'un système extrêmement coûteux, et il est par conséquent hors de question de tester directement sur le robot réel le contrôle calculé pour réaliser un mouvement donné. Ainsi, le simulateur doit permettre aux roboticiens de tester leurs lois de contrôle du mouvement du robot sur le robot virtuel, afin de les valider. Par conséquent, ce simulateur doit être le plus fidèle possible à la réalité.

Nous nous intéressons donc à la simulation d'un robot bipède anthropomorphe, qui doit évoluer dans un environnement inconnu. Pour réaliser les tâches qui lui seront assignées, le robot doit être autonome. Tout d'abord, il lui est nécessaire de se déplacer. Ce seront les contrôles développés par les roboticiens qui calculeront le mouvement que doit réaliser le robot.

Cependant, ce mouvement n'est pas indépendant de l'environnement du robot. Nous avons vu que des applications possibles pour le robot réel concernent des environnements conçus pour des êtres humains, tels que les centrales nucléaires ou les habitations. En

conséquence l'environnement peut être très varié et comporter par exemple des couloirs des obstacles et des escaliers. Or en général l'environnement lui est inconnu au moins initialement. Pour acquérir une certaine connaissance de cet environnement le robot doit être capable de percevoir les objets qui l'entourent et de les reconnaître. C'est à cette condition que le robot sera capable d'évoluer dans un environnement en toute sécurité sans risque de collision ou de chute dommageable en adaptant son comportement de manière appropriée.

## 1.4 Contenu du mémoire

Nous présentons dans le chapitre 2 le modèle de notre robot ainsi que le type de contrôle qui lui sera appliqué. Actuellement aucun contrôle n'a été développé par les roboticiens pour faire marcher ce robot virtuel. Comme ce contrôle nécessite une certaine connaissance du mouvement à réaliser par le robot nous choisissons de rechercher cette connaissance à fournir au robot. Dans ce but nous présentons ce qu'est la marche pour les humains à défaut de robot bipède du point de vue de la biomécanique. Dans le chapitre 3 nous passons en revue les méthodes de synthèse d'images qui permettent de calculer le mouvement de personnages articulés anthropomorphes et nous adaptons l'une de ces techniques dans le chapitre 4 pour calculer le mouvement géométrique de notre robot. Nous complétons ce mouvement de nature cyclique en calculant les mouvements transitoires de départ et d'arrêt.

Concernant la perception et la reconnaissance de l'environnement nous présentons dans le chapitre 5 ce qui est réalisé en robotique dans le cas de robots réels. Dans le chapitre 6 nous modélisons des capteurs proximétriques afin d'obtenir la distance aux objets locaux au robot. Nous réalisons aussi des outils afin de permettre l'analyse de ces données et d'obtenir ainsi des informations plus pertinentes sur l'environnement notamment pour reconnaître le type d'objets en présence. Puis nous appliquons dans le chapitre 7 les outils précédemment développés dans un premier temps pour étudier le placement des capteurs sur le robot afin de déterminer une configuration optimale et dans un second temps pour réaliser une simulation complète du robot évoluant dans un environnement et qui change de mode de locomotion ou s'arrête selon le type d'information délivrée par les capteurs. Enfin nous concluons et proposons des perspectives à cette étude.

# Chapitre 2

## Marche d'un robot anthropomorphe

### 2.1 Introduction

Notre but est de modéliser et de permettre le contrôle d'un robot bipède anthropomorphe. La morphologie de ce robot est très proche de celle d'un humain en ce qui concerne les degrés de liberté, les limites articulaires, la distribution de la masse etc. De plus, nous souhaitons voir le robot marcher comme les humains. Nous espérons ainsi qu'il sera mieux adapté aux environnements conçus pour les humains pour lesquels il est destiné.

Le contrôle d'un robot consiste généralement à calculer le mouvement d'une représentation simplifiée du modèle du robot. Cette représentation est une chaîne articulée nommée *squelette*. Une notation couramment employée en robotique pour décrire une chaîne articulée de manière simple est la notation Denavit-Hartenberg [DH55]. Nous présentons cette notation dans un premier temps. Elle nous permet ensuite de définir le modèle mathématique de notre robot. Puis, nous présentons une méthode de contrôle du robot par fonction de tâches, méthode qui sera employée par les roboticiens [Ali99]. Nous souhaitons ainsi connaître ce qu'est le contrôle de la marche du robot. Enfin, nous abordons la marche humaine d'un point de vue biomécanique afin de savoir en quoi consiste exactement ce mouvement qui nous est familier.

### 2.2 Modèle du robot

#### 2.2.1 Notation Denavit-Hartenberg

Le premier problème dans le contrôle d'une structure articulée est sa représentation mathématique. Pour représenter une structure articulée, nous nous référons à la notation Denavit-Hartenberg (notation DH) [DH55]. Elle décrit la cinématique de chaque lien, c'est-à-dire le segment entre deux articulations, par rapport à ses voisins, en attachant un repère à chacun des liens. Quatre paramètres sont utilisés pour définir une matrice de transformation linéaire entre des repères adjacents, attachés à chaque articulation. Ce sont la longueur du lien  $a_i$ , la distance entre les liens  $d_i$ , la rotation entre les liens  $\alpha_i$  et l'angle entre les liens  $\theta_i$  (cf. Figure 2.1) :

- $a_i$  est la distance de  $z_i$  à  $z_{i+1}$  mesurée sur l'axe  $x_i$
- $\alpha_i$  est l'angle entre  $z_i$  et  $z_{i+1}$  mesuré par rapport à l'axe  $x_i$
- $d_i$  est la distance entre les axes  $x_{i-1}$  et  $x_i$  mesurée sur l'axe  $z_i$
- $\theta_i$  est l'angle entre les axes  $x_{i-1}$  et  $x_i$  mesuré par rapport à l'axe  $z_i$ .

L'axe  $z_i$  du repère  $i$  est sur l'axe de l'articulation. L'axe  $x_i$  est normal à cet axe et pointe vers l'articulation  $i+1$ . L'origine du repère  $i$  se trouve à l'intersection de  $z_i$  avec la normale commune à  $z_i$  et  $z_{i+1}$ .

Il est courant de définir  $(a_i, \alpha_i)$  comme étant les paramètres du lien et  $(d_i, \theta_i)$  comme étant les paramètres de l'articulation. Cette notation est utilisée pour décrire les structures articulées dont les articulations ont uniquement 1 degré de liberté en rotation. Pour représenter des articulations à 3 degrés de liberté, trois articulations à 1 degré de liberté sont localisées en un même point de l'espace.

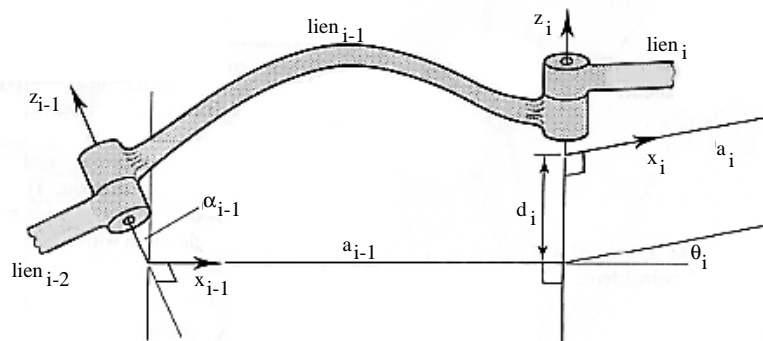


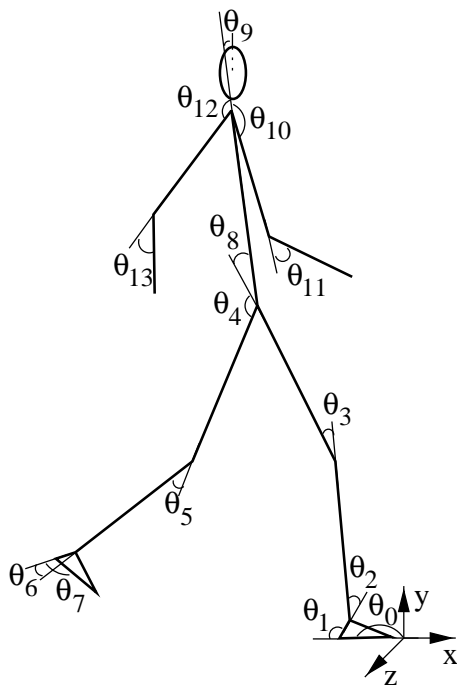
FIG. 2.1: Notation Denavit-Hartenberg.

### 2.2.2 Représentation du robot

Le squelette du robot est représenté sous la forme d'une hiérarchie de segments rigides connectés à des articulations. Ces articulations obligent les extrémités des segments qu'elles relient à rester confondues tout le temps et contraignent les mouvements relatifs de ces segments. Ceci est réalisé implicitement avec la notation Denavit-Hartenberg par le fait que chaque segment est exprimé dans un repère local. Il est caractérisé par sa position et son orientation par rapport au repère local de son segment-père dans la hiérarchie. Le mouvement relatif du segment par rapport à son père comprend au maximum 3 degrés de liberté de rotation et les 3 degrés de liberté de translation étant exclus puisque les extrémités des deux segments doivent rester confondues.

Notre robot virtuel est un modèle simplifié du robot bipède comportant 17 degrés de liberté et dont les segments sont coplanaires<sup>1</sup> (cf. Figure 2.2). Le robot bipède est décrit

<sup>1</sup>Cette représentation correspond aux données de mouvement humain enregistrées et utilisées pour animer le robot (cf. Chapitre 4).



$i$	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
0	0.	0.	0.	-0.22
1	0.118	0.	0.	1.96
2	0.064	0.	0.	-0.37
3	0.311	0.	0.	-0.13
4	0.374	0.	0.3	-2.44
5	0.374	0.	0.	0.27
6	0.311	0.	0.	-0.01
7	0.064	0.	0.	-1.8
8	0.374	0.	0.15	0.22
9	0.378	0.	0.	0.04
10	0.373	0.	-0.15	-2.43
11	0.238	0.	0.	-0.29
12	0.373	0.	0.15	-3.2
13	0.238	0.	0.	-0.92

Paramètres DH associés.

FIG. 2.2: Le squelette de notre robot, en notation Denavit-Hartenberg.

comme une structure articulaire définie en termes de coordonnées articulaires (14 angles) dans la notation Denavit-Hartenberg. L'orteil du pied droit représente la position dans le monde ou coordonnées 3D de l'origine de la structure ce qui rajoute 3 degrés de liberté. Nous associons au robot un *vecteur d'état*  $X = (q, \dot{q})$  où  $q$  est le vecteur de positions articulaires du robot et  $\dot{q}$  le vecteur de vitesses articulaires. Le vecteur d'état contient donc 28 composantes. Un *mouvement* est alors défini comme une séquence de vecteurs d'état dans le temps.

Il est à noter qu'en animation il est courant de choisir comme origine de la structure articulée représentant le personnage virtuel le bassin de celui-ci. Toutefois en robotique pour des raisons tendant à interdire la pénétration des pieds dans le sol il semble plus intéressant de choisir le pied comme origine afin de mieux évaluer et contrôler le mouvement du robot à ce niveau-là. Aussi conformément à l'emploi robotique nous avons choisi le pied comme origine de notre robot virtuel.

## 2.3 Contrôle par fonctions de tâches

Nous souhaitons permettre aux roboticiens d'appliquer leur contrôle sur le robot bipède simulé. Nous nous intéressons donc à savoir comment ce contrôle est réalisé. Une approche courante et susceptible d'être employée à cette fin est l'approche par fonction de tâches. Cette approche par fonction de tâches est une approche globale qui traite à la fois les capteurs de bas niveau et le contrôle de haut niveau. Une fonction de tâches est définie par l'utilisateur et représente complètement la tâche que le robot doit accomplir [SBE91].



En robotique le comportement dynamique d'un robot rigide est donné par l'équation d'état du système :

$$\Gamma = M(q)\ddot{q} + N(q, \dot{q}) + C^T \lambda \quad (2.1)$$

où  $M$  est la matrice d'inertie  $N$  rassemble les termes de forces centrifuge et de Coriolis de gravité et articulaires  $C$  représente les contraintes unilatérales pour traiter les interactions locales entre pieds et sol  $\lambda$  sont les multiplicateurs de Lagrange correspondant aux forces de contact et  $\Gamma$  est la matrice des couples des actionneurs articulaires.

[SBE91] définissent une fonction de tâches  $e(q, t)$  qui peut appartenir à deux classes différentes. Soit elle permet d'effectuer du suivi de trajectoires sans capteur et alors elle est définie comme étant l'erreur entre la trajectoire courante  $q$  et le mouvement de référence  $q^r(t)$  :

$$e(q, t) = q - q^r(t) \quad (2.2)$$

Soit elle considère l'interaction du robot avec l'environnement en prenant en compte les informations fournies par les capteurs. Elle est alors définie par l'équation :

$$e(q, t) = J_1^+ e_1 + (I - J_1^+ J_1) \frac{\partial h_s}{\partial q} \quad (2.3)$$

avec  $J_1$  le jacobien  $J^+$  la pseudo-inverse  $e_1$  la tâche principale par exemple relative aux capteurs et  $\frac{\partial h_s}{\partial q}$  la tâche secondaire par exemple concernant le contrôle du mouvement du robot.

Nous montrons comment la fonction de tâches permet de réaliser le contrôle du robot. Tout d'abord considérant une fonction de tâches quelconque  $e(q, t)$  nous la dérivons et nous obtenons :

$$\dot{e} = \frac{\partial e}{\partial t} + J\dot{q}$$

où  $J$  est la matrice Jacobienne définie par  $J = \frac{\partial e}{\partial q}$ .

En dérivant la fonction de tâche  $e(q, t)$  une seconde fois nous obtenons :

$$\ddot{e} = \frac{d}{dt} \frac{\partial e}{\partial t} + \dot{J}\dot{q} + J\ddot{q} \quad (2.4)$$

$$= f(q, \dot{q}, t) + J\ddot{q} \quad \text{où } f(q, \dot{q}, t) = \frac{d}{dt} \frac{\partial e}{\partial t} + \dot{J}\dot{q} \quad (2.5)$$

$$(2.5) \quad - \quad \ddot{q} = J^{-1}\ddot{e} - J^{-1}f(q, \dot{q}, t) \quad (2.6)$$

En remplaçant  $\ddot{q}$  dans l'équation (2.1) par les termes de l'équation (2.6) nous obtenons :

$$\Gamma = M(q)J^{-1}\ddot{e} - M(q)J^{-1}f(q, \dot{q}, t) + N(q, \dot{q}) + C^T \lambda \quad (2.7)$$

Ensuite en utilisant un contrôleur Proportionnel-Dérivé sur l'erreur dans l'équation (2.7) nous calculons la loi de contrôle suivante :

$$\Gamma = -\mu M(q)J^{-1}G[\nu De + \dot{e}] - M(q)J^{-1}f(q, \dot{q}, t) + N(q, \dot{q}) + C^T \lambda \quad (2.8)$$

où  $G$  et  $D$  sont des matrices définies positives et  $\mu$  et  $\nu$  deux scalaires positifs à choisir par l'utilisateur.

Nous pouvons remarquer de par la définition de la fonction de tâches (cf. équation (2.2)) que cette approche nécessite une trajectoire de référence  $q^r(t)$ . Dans le paragraphe suivant nous présentons la marche humaine d'un point de vue biomécanique de manière à déterminer ce qu'est la marche afin de savoir comment obtenir une trajectoire de référence pour la fonction de tâches.

## 2.4 Étude biomécanique de la marche humaine

Malgré tous les travaux qui lui ont été consacrés la marche suscite encore une multitude de recherches. Il s'agit d'une acquisition motrice précoce. Progressivement affinée chez l'enfant elle devient une fonction complexe intégrée à de multiples niveaux du système nerveux central à tel point qu'il n'est pas possible de lui reconnaître un centre spécial. Elle met en jeu une multitude d'automatismes qui permettent la reproduction cyclique de la séquence active des membres inférieurs tout en autorisant l'adaptivité nécessaire aux variations du relief aux changements d'allure ou aux modifications de trajectoire. Il est d'ailleurs intéressant de noter que l'homme par son mode de locomotion bipède possède des membres inférieurs très sophistiqués : 29 degrés de liberté et 48 muscles pour chaque membre.

La liste des systèmes spécifiques qui permettent le parfait déroulement du programme de marche chez l'humain est longue : moteur pelvien pour la marche en terrain horizontal et poussée du triceps sural et des extenseurs de hanche pour les changements de pente et les reliefs accidentés ; organe de contact au sol spécifiquement adapté grâce à ses grandes capacités de déformations dues à ses nombreux degrés de liberté à sa semelle plantaire épaisse au jeu vertical et transversal des métatarsiens ; capteur d'accélération angulaire et linéaire assurant le niveau vestibulaire de la stabilisation dynamique et permettant les contrôles de trajectoire en l'absence du contrôle visuel ; etc.

Le but de cette étude est d'essayer de comprendre la marche humaine qui est une activité d'une extrême complexité. Il suffit de regarder un jeune enfant s'essayer à cette activité nouvelle et l'on se rend compte des difficultés que doit surmonter le marcheur en puissance. Activité apprise la marche est également intégrée au niveau non volontaire.

### 2.4.1 Concepts de base

Pour étudier le mouvement humain qui est réalisé dans un espace à trois dimensions on détermine trois plans de référence du corps humain qui sont le *plan sagittal* le *plan coronal* ou *frontal* et le *plan transversal* (cf. Figure 2.3). La marche est habituellement

étudiée dans le plan sagittal car c'est dans ce plan que le mouvement de marche est le plus important. Ainsi les chercheurs ont tendance à ignorer les deux autres plans lors de l'étude de la marche. Pourtant ces autres plans peuvent fournir des informations non négligeables. En effet l'analyse de la démarche humaine devrait vraiment être tridimensionnelle de manière à améliorer le modèle de démarche (cf. Figure 2.4). Les trois projections séparées pourraient être combinées en une image composée et les paramètres exprimés dans le repère du corps plutôt que dans celui du laboratoire (cf. Figure 2.5).

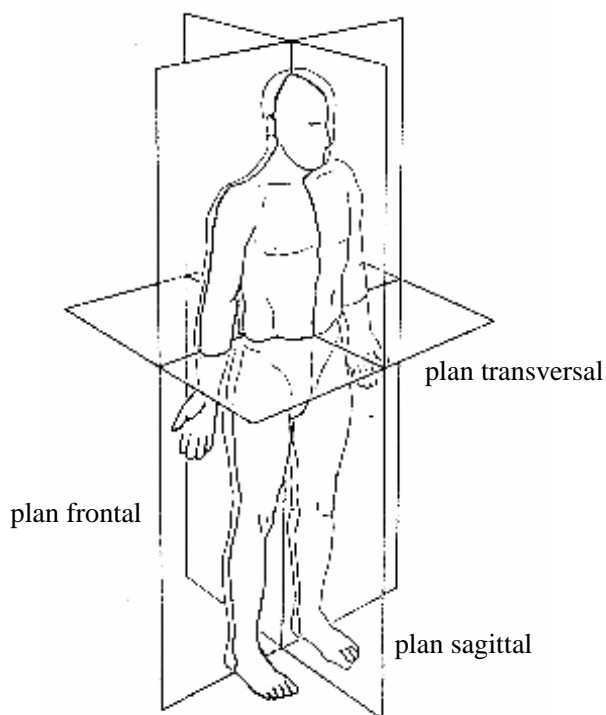


FIG. 2.3: Plans de référence du corps humain dans la position anatomique standard ([VDO92]).

Lorsque l'on désire étudier la marche du point de vue biomécanique on se représente un modèle cyclique du mouvement qui est répété sans arrêt pas après pas. Ainsi les descriptions de marches sont normalement condensées en un simple *cycle* en faisant l'hypothèse que les cycles suivants sont tous les mêmes. Bien que cette hypothèse ne soit pas tout à fait correcte c'est une approximation raisonnable. La Figure 2.6 illustre un cycle de marche pour un enfant de huit ans. Habituellement on note par convention que le cycle débute lorsque le pied de référence le pied droit pour nous fait contact avec le sol.

La nomenclature de la Figure 2.6 se réfère au côté droit du corps mais la même terminologie est à appliquer au côté gauche qui correspond pour une personne normale à la moitié de cycle suivant (ou précédent) du côté droit. En effet dans la marche normale il y a une symétrie entre les côtés gauche et droit.

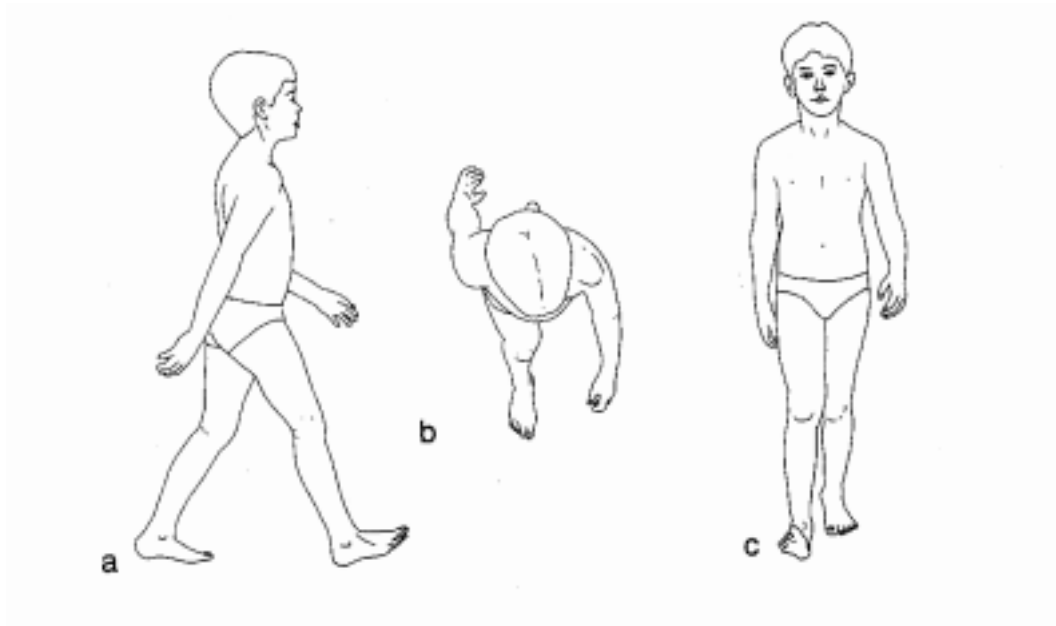


FIG. 2.4: Démarche vue dans les trois plans : a) sagittal, b) transversal, c) frontal ([VDO92]).

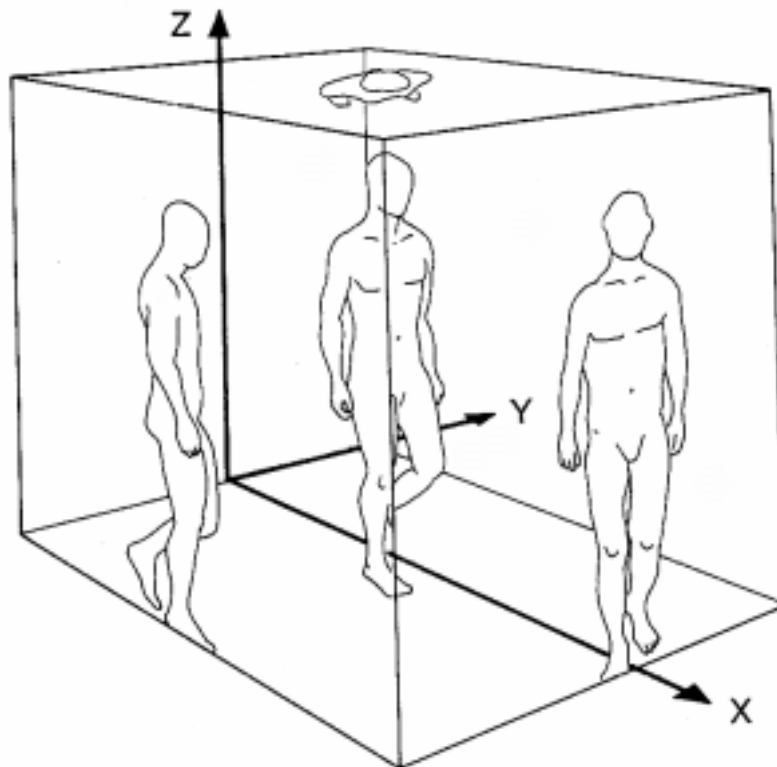


FIG. 2.5: Projection des trois mouvements plans pour obtenir la démarche 3D ([VDO92]).

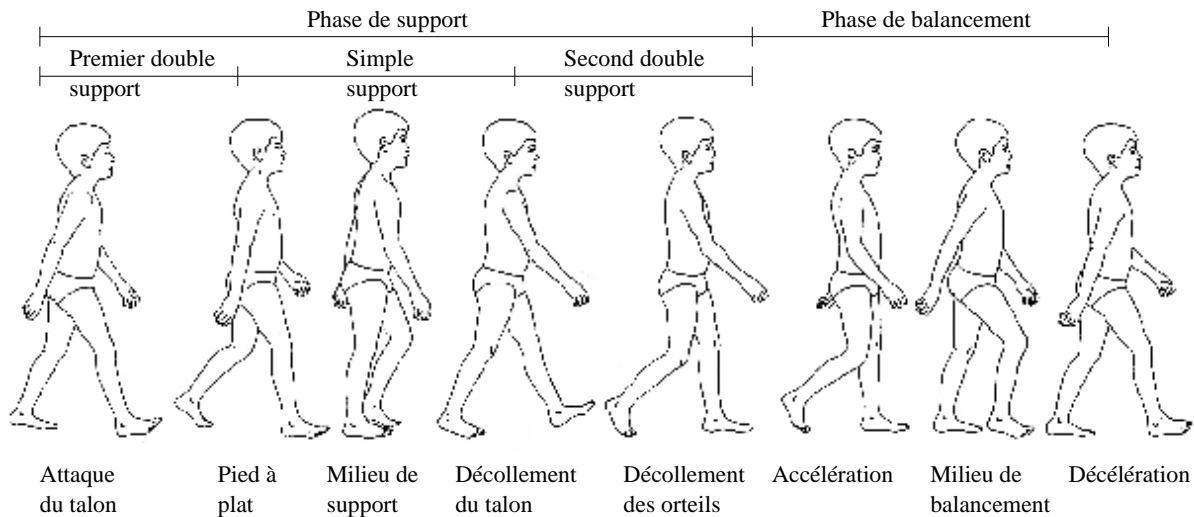


FIG. 2.6: Cycle de marche normal pour un garçon de huit ans ([VDO92]).

### 2.4.2 Phases de la marche

Le cycle de marche se décompose en deux phases principales pour chaque pied (cf. Figure 2.6) : pendant la *phase de support* le pied est posé au sol alors que pendant la *phase de balancement* ou *de vol* ce même pied n'est plus en contact avec le sol et la jambe est balancée en vue de la préparation du prochain contact avec le sol. La phase de support peut elle-même être subdivisée en trois phases :

- Premier double support : les deux pieds sont en contact avec le sol.
- Simple support : un des pieds est en contact avec le sol tandis que l'autre pied est balancé.
- Second double support : les deux pieds sont à nouveau en contact avec le sol.

Maintenant si l'on considère les phases d'un cycle de marche pour un pied donné la phase de support dure 62% de la durée du cycle complet et la phase de balancement 38% (cf. Figure 2.7).

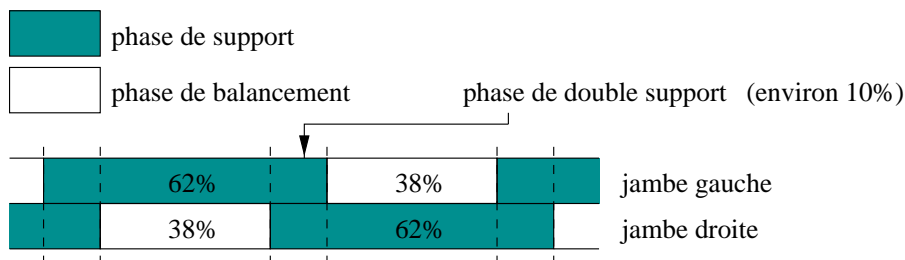


FIG. 2.7: Temps passé sur chaque jambe pendant un cycle de marche normale ([VDO92]).

### 2.4.3 Périodes lors de la marche

Traditionnellement le cycle de marche est divisé en huit périodes, cinq pendant la phase de support et trois pendant celle du vol. Ces périodes portent le nom des événements liés au mouvement du pied, lesquels débutent ces périodes (cf. Figure 2.8).

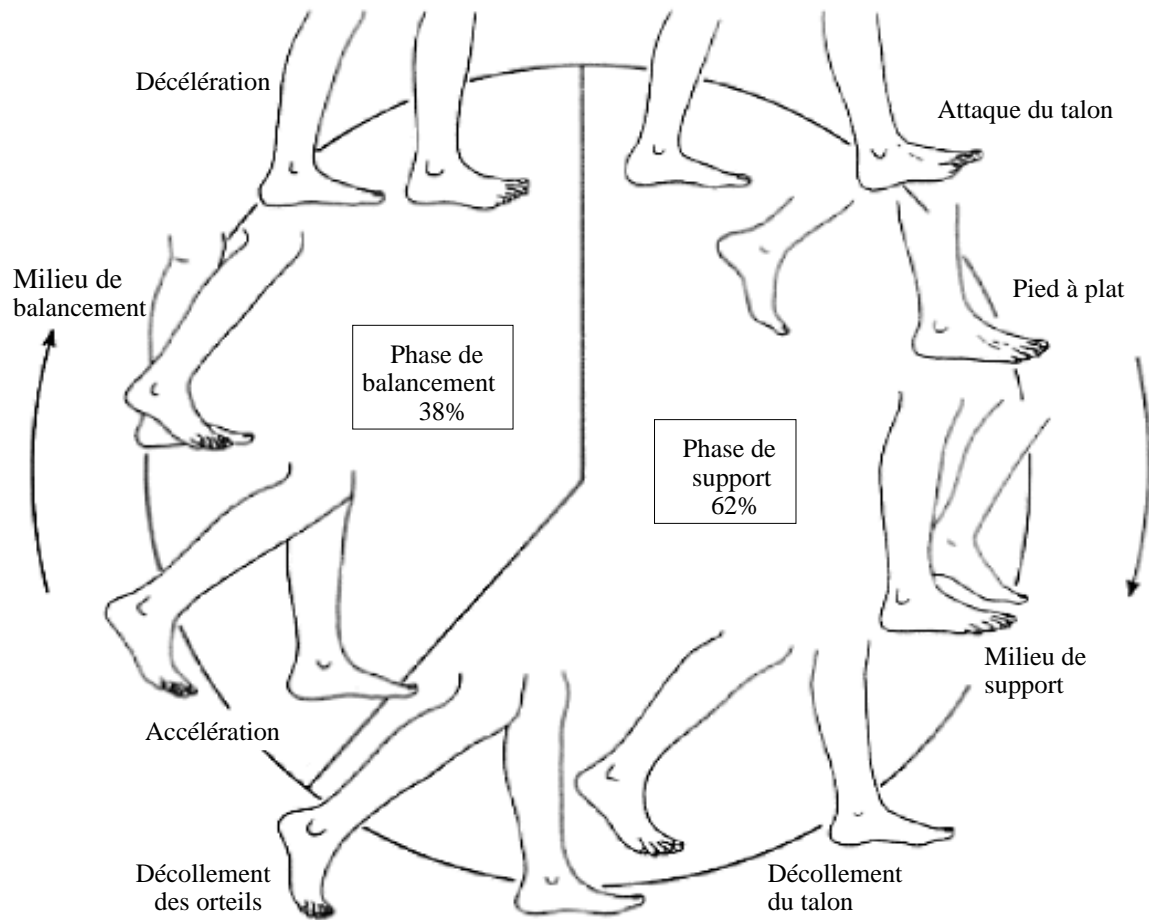


FIG. 2.8: Nomenclature traditionnelle pour la description des huit principales périodes, qui soulignent la nature cyclique de la marche humaine ([VDO92]).

#### a) Périodes de la phase de support

- L'*attaque du talon* initialise le cycle de marche et représente le moment où le centre de gravité du corps est au plus bas. Au cours de cette action très brève, le membre inférieur est allongé au maximum et le bassin est en adduction horizontale du côté de l'attaque par rapport au membre portant. Le genou est en extension complète ou quasi complète. La cheville se trouve en position neutre. Ces déplacements articulaires ont pour but de donner au membre inférieur sa longueur maximale.
- Le *pied à plat au sol* apparaît lorsque la surface plantaire du pied touche le sol. Le contact ayant été pris par le talon, puis par la semelle plantaire tout entière.

le marcheur se trouve en équilibre monopodal. La longueur maximale du membre inférieur atteinte au cours de la phase précédente serait maintenant désavantageuse car elle soumettrait le centre de gravité à une accélération verticale trop importante. Pour cette raison le genou sera légèrement fléchi de  $15^\circ$  à  $25^\circ$  suivant le marcheur et la rapidité du déplacement. En effet si le marcheur normal bloque le genou en extension quasi totale lors d'une marche très lente l'accélération de l'allure détermine une flexion du genou de plus en plus importante.

- Le *milieu de support* a lieu lorsque le pied de balancement (contrelatéral) dépasse le pied de support et que le centre de gravité du corps est dans sa position la plus haute.
- Le *décollement du talon* se produit lorsque le talon perd contact avec le sol et la poussée est engendrée par les muscles “triceps surae” ce qui entraîne le fléchissement plantaire de la cheville. Cette étape se caractérise par une activité intense des muscles dits “fléchisseurs plantaires” agissant sur la cheville mais dont l'activité n'a pas pour but de provoquer ce dernier mouvement. En effet la cheville reste neutre ou bien le mouvement de dorsiflexion s'accroît légèrement non par un déplacement du pied vers la face antérieure de la jambe mais par un déplacement angulaire antérieur du segment jambier. Le pied reste l'élément squelettique stable à partir duquel les autres segments se meuvent. Lors du décollement du talon les orteils restent fermement en contact avec le sol la cheville se trouve placée en position haute. Ceci risquerait de soumettre le centre de gravité à une accélération verticale trop importante aussi le genou se plie-t-il au même instant.
- Le *décollement des orteils* termine la phase de support alors que le pied quitte le sol.

## b) Périodes de la phase de vol

- L'*accélération* commence dès que le pied quitte le sol et que le sujet active les muscles fléchisseurs de la hanche pour accélérer la jambe vers l'avant. À ce stade le membre inférieur atteint son minimum de longueur les articulations de hanche et de genou se fléchissent l'articulation de la cheville se mobilisant pour porter le pied en dorsiflexion. Le poids du corps du marcheur est maintenant entièrement supporté par le membre inférieur opposé.
- Le *milieu de balancement* a lieu lorsque le pied passe juste sous le corps coïncidant avec l'évènement milieu de support de l'autre pied.
- La *décélération* décrit l'action des muscles qui ralentissent la jambe et stabilise le pied par anticipation du prochain contact du talon avec le sol. Le squelette se place dans la position de plus grande longueur possible afin d'atteindre le sol en avant du corps. On assiste alors de la phase d'accélération à un raccourcissement maximal destiné à éviter le sol en fléchissant la hanche le genou et la cheville. On retrouve maintenant le marcheur dans la position suivante : le bassin est en rotation antérieure maximale du côté où va se produire l'attaque du talon la hanche est en flexion le genou est en extension totale ou quasi totale le pied est en dorsiflexion le talon dirigé vers le sol la plante du pied éloignée de la surface de contact.

### 2.4.4 Paramètres de la marche

La décomposition du mouvement en périodes permet de décrire les paramètres de mouvement sur des intervalles réduits. En outre, lorsque l'on rapporte la durée de chacun de ces intervalles à la durée totale de l'enjambée, on constate que les valeurs des rapports sont presque inchangées, quelle que soit la vitesse de marche. Cela révèle un fait important : indépendamment de sa fréquence, le patron de base conserve à peu près la même structure temporelle.

On peut aussi déterminer d'autres paramètres, tels que diverses mesures de la marche. La longueur de la foulée donne la distance parcourue par une personne pendant une foulée, ou cycle, et peut être mesurée comme étant la longueur entre les talons, depuis un contact du talon à un autre, du même côté. Deux longueurs de pas, gauche et droit, font une longueur de foulée. Pour des personnes normales, dont la marche n'est pas altérée par une maladie, les deux longueurs de pas sont approximativement égales. Un autre paramètre utile est la largeur du pas, qui décrit la distance médiolatérale entre les pieds et a une valeur de quelques centimètres (cf. Figure 2.9). Enfin, l'angle du pied par rapport à la ligne de progression peut aussi fournir une information intéressante sur le degré de rotation externe ou interne de l'extrémité la plus basse durant la phase de support.

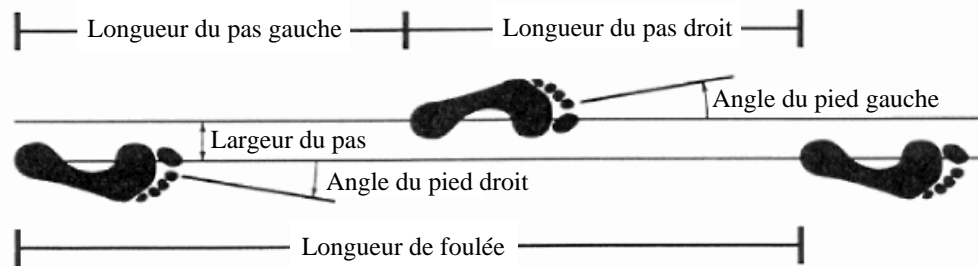


FIG. 2.9: Empreintes fournissant des paramètres de distance intéressants ([VDO92]).

Un autre type d'information pouvant aider l'étude du mouvement de marche est fourni par les données expérimentales d'activités musculaires enregistrées [Win90, VDO92]. Ces données permettent de savoir à chaque instant, quels sont les muscles actifs et les mesures physiques associées. Toutefois, pour déterminer la correspondance entre activité musculaire et action physique résultante, il est nécessaire de développer un modèle biomécanique de corps humain très complexe. Il faut notamment prendre en considération les formes exactes des os, des muscles et des tendons, ainsi que les liens/attaches entre eux, et émuler les comportements dynamiques de cette structure squelette-muscles de manière réaliste. Cette dernière tâche est particulièrement délicate. En synthèse d'images, il est courant de ne considérer que des modèles plus ou moins approximatifs [KS97, SPCM97, ZCK97].

Ainsi, il existe des paramètres nombreux et variés pour étudier la marche humaine. Cependant, tous font prendre conscience qu'il n'est pas aisé de décrire ce mouvement qui nous est si naturel, en termes simples. Bon nombre de subtilités, de détails viennent agrémenter le mouvement de base. Toutefois, ils ne sont pas négligeables, puisque c'est



sur eux que repose le réalisme visuel de la démarche. Comme ont pu le remarquer à leurs dépens les chercheurs en animation de personnages et autres animateurs. Par ailleurs la Figure 2.9 indique dans quelle mesure la marche en 2D (plan sagittal) est une approximation de la marche réelle.

## 2.5 Conclusion

Nous avons représenté le robot au moyen de la notation Denavit-Hartenberg. Ce robot est composé de 17 degrés de liberté. Pour contrôler le mouvement de notre robot une approche donnant de bons résultats pratiques est l'approche par fonction de tâches. Cependant cette approche requiert une trajectoire de référence pour calculer le mouvement dynamique du robot. Nous nous sommes donc intéressés à acquérir ces trajectoires de référence. D'une part elle nous permettront d'animer le robot de manière géométrique et d'avoir déjà une idée des difficultés et des problèmes liés à son évolution dans un environnement inconnu. D'autre part elles pourront être utilisées par les roboticiens pour développer leurs lois de contrôle.

L'étude biomécanique de la marche humaine nous a permis de nous rendre compte qu'il n'est pas évident de déterminer rapidement une telle marche de référence. De nombreux paramètres permettent d'étudier la marche mais aucun à notre connaissance n'est capable de nous fournir ce dont nous avons besoin à savoir l'évolution des angles des articulations au cours d'un cycle de marche. Parce que le corps humain est une chaîne biodynamique articulée les différents segments du corps s'influencent mutuellement engendrant ainsi un mouvement complexe des jambes lors de la marche. C'est pourquoi notre attention s'est portée sur les méthodes de calcul de mouvements proposées en synthèse d'images. Nous présentons ces méthodes dans le chapitre suivant avec pour objectif d'obtenir un mouvement de référence qui soit physiquement valide et réaliste et ainsi adapté au futur sujet d'étude le robot réel.

## Chapitre 3

# État de l'art : Synthèse d'un mouvement de marche

### 3.1 Introduction

Nous nous intéressons au calcul d'une trajectoire de référence pour notre robot bipède. Nous recherchons donc des méthodes habituellement employées en synthèse d'images pour calculer le mouvement de marche d'un personnage de synthèse simulant un humain ce qui se rapproche le plus de notre robot anthropomorphe.

La marche d'un personnage articulé représentatif d'un être humain découle de mécanismes complexes. Diverses approches tentent de simuler ce comportement. Pour cela le réalisme visuel de la simulation est recherché au niveau du mouvement et non pas au niveau de la plastique du personnage. En effet on s'intéresse davantage au mouvement principal du personnage son apparence pouvant être considérée comme un habillage qui peut être ajouté par la suite soit en recouvrant le squelette d'une peau modélisée par surfaces déformables [CHP89GV91Tur95ST95TSC96Opa96Sto99] soit directement avec des vêtements [HBG92CYMTT92VCMT95VMTJT96BW98].

L'une des difficultés de la simulation d'un tel personnage réside dans le fait que notre œil est très habitué à la marche humaine et que les imperfections comme par exemple une mauvaise coordination des membres ou une mauvaise synchronisation des mouvements du personnage choquent immédiatement. Le réalisme visuel peut être assuré par un certain réalisme physique : conservation du centre de gravité mouvement avec une continuité  $C^2$  forces raisonnables etc. En modélisant le corps humain par un squelette plus ou moins fidèle à la réalité les méthodes cinématiques et les méthodes basées sur l'interpolation que nous développons dans le paragraphe 3.2 tentent de calculer l'évolution des degrés de liberté des articulations et leurs butées angulaires au cours du mouvement et ce de manière correcte afin d'éviter la génération de mouvements irréalistes et incohérents.

Une animation plus complexe est réalisée par des méthodes dynamiques que nous développons dans le paragraphe 3.3. Ces méthodes considèrent que le squelette représentant le personnage subit des forces externes telles que la gravité ainsi que des forces

internes Γ celles exercées par les muscles notamment. On peut s'assurer que les mouvements sont compatibles avec les données biomécaniques de l'être humain et les forces musculaires que l'être humain peut développer Γ en ajoutant des contraintes physiques à la simulation.

Cependant Γ un des buts en synthèse d'images est non seulement l'animation d'un personnage Γ mais aussi son contrôle. En effet Γ on peut vouloir imposer une trajectoire Γ un mouvement au personnage Γ ou encore ajouter des obstacles dans l'environnement où évolue le personnage Γ ce qui doit avoir pour conséquence la modification du chemin et donc du mouvement du personnage Γ afin d'éviter les obstacles. Il est alors nécessaire de paramétrer la marche. Les paramètres de contrôle doivent être aussi intuitifs que possible. Des méthodes d'optimisation Γ ou celles basées sur des contrôleurs Γ développées dans le paragraphe 3.3 Γ ou encore celles basées sur la capture de mouvement Γ développées dans le paragraphe 3.4 Γ tentent de répondre à cette attente.

## 3.2 Méthodes cinématiques

### 3.2.1 Définitions

La cinématique étudie les variations au cours du temps des relations géométriques des objets. Il existe deux catégories d'animations cinématiques de structures articulées Γ la cinématique directe et la cinématique inverse.

#### a) Cinématique directe

La *cinématique directe* consiste à calculer le mouvement dans l'espace de la structure articulée Γ à partir des lois d'évolution des degrés de liberté spécifiées par l'utilisateur :

$$X(t) = f(q(t)) \quad (3.1)$$

où  $X(t)$  est le vecteur de coordonnées cartésiennes de la structure articulée à un instant  $t$  donné Γ qui est fonction du vecteur d'état de coordonnées articulaires  $q(t)$ . Le mouvement de l'effecteur terminal (dans le cas d'un personnage Γ ce sont les pieds et les mains) est déterminé indirectement par accumulation de toutes les transformations en amont de l'effecteur terminal Γ comme l'arbre de la structure articulée est parcouru en descendant.

#### b) Cinématique inverse

La *cinématique inverse* consiste à calculer les lois d'évolution des degrés de liberté à partir des mouvements dans l'espace cartésien :

$$q(t) = f^{-1}(X(t)) \quad (3.2)$$

Par commodité Γ l'animateur définit la position des effecteurs terminaux seulement Γ et la cinématique inverse résout la position et l'orientation de toutes les articulations dans la hiérarchie de segments menant jusqu'à l'effecteur terminal (cf. Figure 3.1). Le mouvement est alors appelé mouvement orienté vers un but.

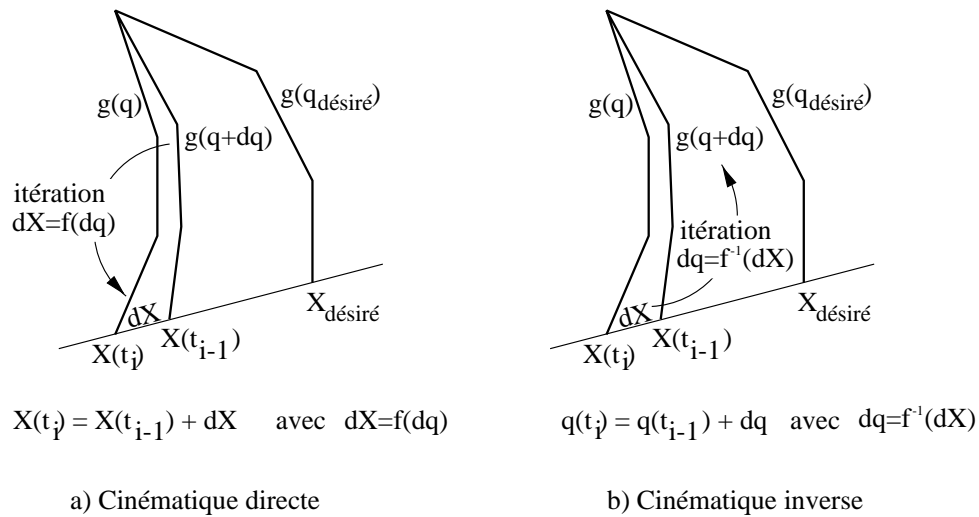


FIG. 3.1: Une itération par cinématique.

### 3.2.2 Animation par cinématique directe

Animer un personnage articulé avec pour seul moyen la cinématique directe est très difficile. En effet l'évolution des paramètres angulaires qui engendre un mouvement de marche n'est pas intuitive et il est difficile de s'en faire une idée. Il faut être au moins un animateur confirmé pour réaliser cette tâche ardue. De plus les techniques cinématiques deviennent compliquées à utiliser dès lors que la complexité de la structure articulée augmente. À chaque fois qu'une articulation est ajoutée dans la hiérarchie le nombre de degrés de liberté de la figure est incrémenté de 1 au moins de même pour la dimension du vecteur d'état  $q$ . Une première méthode consiste à hiérarchiser le problème une seconde à utiliser des méthodes d'interpolation.

#### a) Hiérarchisation du problème

[SIE53] tentent d'expliquer la nature cinématique sous-jacente de la locomotion bipède. Pour cela ils définissent un ensemble de déterminants de démarches qui sont appliqués l'un après l'autre sur un squelette simple (cf. Figure 3.2). Ainsi le mouvement est construit au fur et à mesure que la complexité s'accroît.

#### Déterminants de la démarche

- a) La *démarche du compas* : les jambes restent tendues évoluant dans des plans parallèles et le bassin bouge en une série d'arcs dont le rayon est déterminé par la longueur de la jambe.
- b) La *rotation du bassin* : en permettant au bassin de tourner autour d'un axe vertical en son centre cela permet à la longueur du pas d'être augmentée et à l'arc d'être aplati.
- c) L'*inclinaison du bassin* : si le bassin peut se pencher et tourner l'arc de la trajectoire peut encore s'aplatir. En pratique la hanche du côté du balancement dans la marche

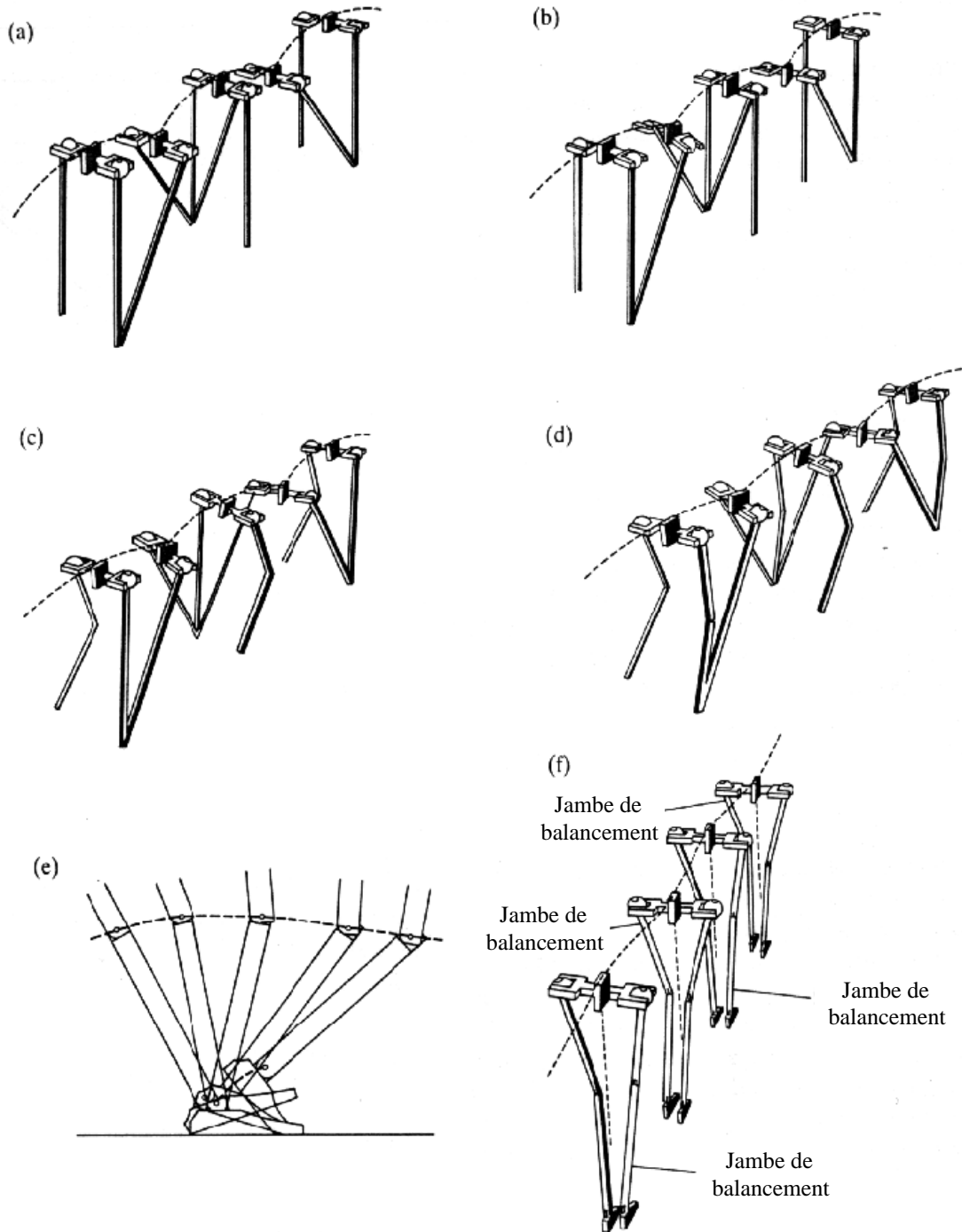


FIG. 3.2: *Diverses étapes de la démarche [IRT81] : (a) Démarche d'un compas. (b) Rotation du bassin. (c) Inclinaison du bassin. (d) Flexion de la jambe d'appui. (e) Flexion plantaire de l'angle d'appui. (f) Déplacement latéral du bassin.*

tombe en-dessous de la hanche de support. L'abaissement se produit immédiatement après la phase de double support et du décollement de l'orteil du côté du balancement. L'inclinaison du bassin alors introduite entraîne nécessairement une flexion du genou du côté du balancement.

- d) La *flexion de la jambe d'appui* : à ce moment-là l'arc de la trajectoire du bassin est encore plus aplati.
- e) La *flexion plantaire de l'angle d'appui* : la transition entre la phase de double support et la phase de balancement est plus régulière sans à-coups si l'angle de la jambe d'appui diminue juste avant le décollage de l'orteil c'est-à-dire si le pied fléchit par rapport au tibia.
- f) Le *déplacement latéral du bassin* : la marche normale entraîne le déplacement du bassin d'un côté à un autre comme le poids est transféré d'une jambe à l'autre.

### *Discussion*

Le mouvement est ainsi décomposé hiérarchisé ce qui rend la nature du mouvement plus compréhensible et sa modélisation semble plus simple à réaliser. Une animation intuitive du personnage consiste alors premièrement en une rotation de la hanche deuxièmement en une rotation du genou troisièmement en une rotation de la cheville. Le résultat de cette animation simplifiée apparaît raisonnable tant qu'aucun sol n'est affiché mais de sévères restrictions apparaissent quand un sol est ajouté à la scène. En effet il y a de fortes chances pour que le pied pénètre dans le sol car aucune contrainte de contact entre le pied et le sol n'est prise en compte.

Une alternative est alors d'animer le personnage articulé à partir de son pied de support qui aura au préalable été positionné correctement sur le sol. La racine de la hiérarchie devient ce pied de support à la place du bassin. Cependant durant la marche le pied de support change passant du pied droit au pied gauche et ainsi de suite. Ainsi le mouvement de la racine n'est continu que par intervalle et qu'au premier ordre c'est-à-dire que pour les positions. Or on a besoin d'avoir une continuité au moins d'ordre 1 c'est-à-dire au niveau des vitesses pour assurer la fluidité du mouvement.

### **b) Méthodes par interpolation**

Les systèmes d'images clés (*keyframes*) tirent leur nom du système de production hiérarchique traditionnel développé par Walt Disney [WW92]. Dans ces systèmes les animateurs les plus "habiles" conçoivent une séquence particulière en dessinant des images clés qui constituent la base de l'animation. L'habileté particulière de ces animateurs repose sur la façon convaincante d'animer un personnage donné. La production de la séquence complète est laissée à des artistes "secondaires" qui se servent des images clés pour produire les images intermédiaires. Il est possible de spécifier à nouveau des séquences ou de les ajuster. Émuler cette technique par ordinateur consiste à remplacer les artistes secondaires par l'interpolation qui produit les images intermédiaires de manière automatique. Il est cependant nécessaire d'avoir une idée du sens de ce que l'on souhaite

interpoler. Par exemple le résultat est différent selon que l'interpolation d'un segment est réalisée sur l'angle avec l'horizontale ou sur l'une de ses extrémités (cf. Figure 3.3).

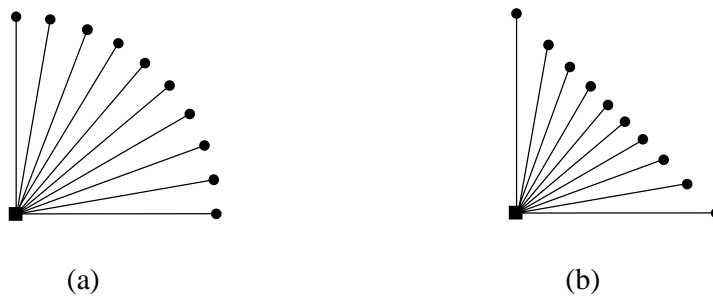


FIG. 3.3: *Interpolation d'une rotation : (a) l'interpolation d'angle produit un mouvement différent de celui obtenu par interpolation des extrémités (b).*

### *Interpolation d'images clés*

À l'origine les positions intermédiaires sont obtenues en interpolant les images clés elles-mêmes. L'interpolation linéaire produit des effets indésirables tels que le manque de régularité dans le mouvement la présence de discontinuités dans la vitesse du mouvement et de distorsions dans les rotations. Les méthodes d'interpolation par splines sont plus appropriées. Les splines sont construites en général par morceaux avec des polynômes cubiques et sont donc continues sur les intervalles avec une continuité  $C^1$  ou  $C^2$  aux coutures. Deux types de splines sont particulièrement employés : les splines d'interpolation avec une continuité  $C^1$  aux nœuds et les splines d'approximation avec une continuité  $C^2$  aux nœuds. Pour l'animation en général les splines les plus intéressantes sont les splines d'interpolation telles que les cardinal splines et les Catmull-Rom splines.

### *Interpolation de paramètres*

La technique d'interpolation a rapidement été généralisée pour interpoler n'importe quel paramètre affectant le mouvement fournissant ainsi un contrôle de plus haut niveau que les images clés traditionnelles. Cette méthode est appelée *animation par images clés paramétrées*. Dans un modèle paramétré l'animateur crée des positions clés en spécifiant l'ensemble des valeurs des paramètres. Les paramètres sont ensuite interpolés par des splines généralement et les images sont finalement construites individuellement à partir des paramètres interpolés.

L'interpolation peut porter sur différents paramètres ce peut être par exemple :

- *l'interpolation de coordonnées cartésiennes* : dans [BC93] des points de contrôle de la spline déterminent le mouvement en position et vitesse des articulations durant la marche. Les valeurs intermédiaires de ces paramètres sont alors calculées du pas précédent au pas courant par interpolation cubique. Ainsi les caractéristiques du mouvement d'un objet sont explicitement spécifiées par les splines (cf. Figure 3.4 (a)).



a) Interpolation des coordonnées cartésiennes. b) Interpolation des coordonnées articulaires.

FIG. 3.4: *Différentes interpolations.*

- l'*interpolation de coordonnées articulaires* : dans [Zel82ΓZel83]Γ la démarche du squelette synthétique est basée sur une connaissance biomécanique de la locomotion. Une posture clé du personnage est définie par des coordonnées articulaires  $q$ . Ces postures sont interpolées linéairement pour produire les valeurs angulaires intermédiaires (cf. Figure 3.4 (b)).
- l'*interpolation de vitesses* : dans [BMTT90]Γ le cycle de marche du personnage est paramétréΓen particulier par la modulation de la fréquence de la marche qui permet de contrôler la vitesse du personnage. Cette vitesse est interpolée linéairement pour atteindre la fréquence désirée à partir de la fréquence courante.

### *Discussion*

La méthode d'interpolation de paramètres permet de produire de meilleures images que l'interpolation d'images clésΓcar elle tient compte non seulement des positionsΓmais aussi des rotations ou des vitesses des objets par exemple. Les images clés sont facilement utilisables pour des mouvements de corps rigides simplesΓmais deviennent vite ingérables pour des objets plus complexes.

De plusΓune animation précise et réaliste de personnages articulés exige un grand nombre de positionsΓnécessitant une mise au point attentive et un temps importantΓpour ne pas engendrer involontairement des mouvements interdits. Ceci résulte du fait qu'aucune aide au réalisme assurant la conformité du mouvement aux lois physiques n'est fournie. Aucun traitement des collisions et interpénétrations n'est effectué.

ToutefoisΓpour limiter la non-pénétration dans le solΓune méthode consiste à changer la racine de la structure articulée hiérarchisée lorsque le pied de support change [BC89]. À chaque contact d'un pied au solΓce nouveau pied de support devient la racineΓet sa position est fixée dans les coordonnées du monde. Cette méthode est utilisée pour simuler un pendule inverséΓde manière à calculer un mouvement réaliste de la jambe d'appui.



D'autre part, dans le cas de l'animation par positions clés paramétrées, la paramétrisation du système nécessite un soin tout particulier car des paramétrisations inappropriées et/ou des interpolations naïves peuvent aboutir à des mouvements de qualité médiocre. Il est donc nécessaire de réaliser tout un jeu d'expérimentations pour obtenir le résultat désiré.

### 3.2.3 Animation par cinématique inverse

L'insuffisance de la cinématique directe vient principalement de la difficulté à spécifier un ensemble de lois de variation des articulations au cours du temps. De plus, aucune contrainte ne peut être ajoutée au mouvement. Ainsi, il est quasiment impossible de vérifier que les pieds ne pénètrent pas dans le sol pendant les phases de support. Cette limitation peut être supprimée grâce à la cinématique inverse. Le but de cette technique est de spécifier le mouvement désiré sous forme d'un ensemble de contraintes de manière à faire poursuivre une trajectoire par chaque degré de liberté. Les valeurs des coordonnées articulaires de chaque articulation sont calculées à partir des coordonnées cartésiennes de segments particuliers (mains, pieds ...).

#### a) Principe

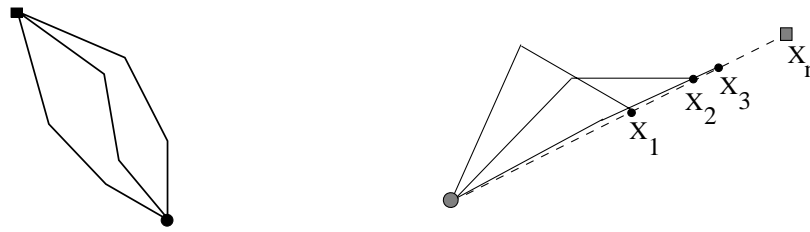
Le mouvement  $\Delta X$  du segment terminal d'une chaîne est exprimé dans les coordonnées cartésiennes ou coordonnées du monde. Le système doit calculer la variation  $\Delta q$  du vecteur d'état du personnage c'est-à-dire les orientations des segments intermédiaires pour satisfaire les contraintes de position liées à  $X + \Delta X$ . La relation entre la tâche principale  $\Delta X$  et les déplacements angulaires  $\Delta q$  est de la forme :

$$\Delta X = J \Delta q \quad (3.3)$$

où  $J$  est la matrice Jacobienne du système avec  $J_{ij} = \frac{dX_i}{dq_j}$  ce qui indique un comportement cinématique approximé localement par un modèle linéaire. Souvent,  $J$  n'est pas directement inversible à cause des dimensions généralement différentes de  $X$  et de  $q$ . Un premier problème vient du fait que les actions traduisant une intention de déplacement ou de modification n'appartiennent pas à l'espace articulaire. Il existe alors une infinité de solutions dans l'espace articulaire pour obtenir une même configuration dans l'espace cartésien (cf. Figure 3.5 (a)). Un second problème peut survenir : il s'agit du cas où le mouvement spécifié est impossible à réaliser (cf. Figure 3.5 (b)).

#### b) Recherche de solutions

Le problème de recherche de  $q$  pour une position  $X$  donnée est en général sous-défini et le système est dit redondant. Tout un sous-espace  $q_X$  défini par :  $q(q_0, \dots, q_{n-1}) \in q_X$  si  $f(q) = X$  fournit une solution pour  $X$ . On tente alors de réduire le sous-espace de solutions possibles en ajoutant des contraintes au système. Chaque contrainte forme un sous-espace et l'intersection de ces sous-espaces de contraintes avec  $q_X$  donne un ensemble de solutions possibles. La différence entre le nombre de degrés de liberté et le nombre



a) Plusieurs configurations de la chaîne pour atteindre le même objectif. b) Configuration non réalisable, car l'extrémité de la chaîne ne peut atteindre l'objectif  $X_n$ .

FIG. 3.5: Problèmes liés à la cinématique inverse.

de contraintes indépendantes est appelée degré de redondance. Par exemple la solution proposée dans [GM85] est :

$$\Delta q = J^+ \Delta X + \alpha (\mathbb{I} - J^+ J) \Delta z \quad (3.4)$$

où  $J^+$  est la pseudo-inverse de la matrice Jacobienne  $J$   $\alpha$  est une constante de pénalité  $\mathbb{I}$  est la matrice d'identité et  $\Delta z$  appelée tâche secondaire est une contrainte à minimiser. Cette tâche secondaire est forcée à rester dans l'espace nul de la tâche principale par la projection sur l'espace  $\text{Ker}(J)$ . Ainsi le second terme de l'équation (3.4) n'affecte pas la réalisation de la tâche principale et ce quelque soit la tâche secondaire  $\Delta z$ . En général cette tâche secondaire est utilisée pour prendre en compte les limites angulaires des articulations ou pour minimiser un certain critère d'énergie (cf. Figure 3.6). Elle peut être aussi utilisée à d'autres fins par exemple pour maintenir l'équilibre du personnage comme c'est le cas dans [BMT96]. D'autres méthodes de calcul plus efficaces permettent d'améliorer les résultats obtenus. Notamment un mouvement géométriquement possible peut être couplé à un algorithme d'optimisation ce qui engendre un résultat meilleur que la projection sur l'espace possible de mouvement (cf. paragraphe 3.3.4).

### 3.2.4 Combinaison des deux cinématiques

La plupart des approches cinématiques utilisées pour la génération de locomotion d'humains synthétiques reposent sur une connaissance biomécanique et combinent la cinématique directe et la cinématique inverse pour calculer les mouvements. [BMTT90 BT92] utilisent une approche cinématique directe standard pour générer des positions clés à interpoler à partir d'informations biomécaniques. Ils utilisent ensuite des algorithmes de cinématique inverse pour maintenir des contraintes supplémentaires sur la position du pied. Un processus de correction de la jambe est alors appliqué pour modifier les positions intermédiaires invalides. Par exemple si le pied pénètre dans le sol un algorithme de cinématique inverse va modifier sa position modélisant ainsi le contact au sol.

De manière analogue [vdP97] définit le mouvement d'un bipède à partir d'empreintes de pas et d'informations temporelles. Ces critères permettent de synthétiser le mouvement du centre de masse par une spline. Ensuite à chaque instant de l'animation les positions des jambes sont calculées par cinématique inverse en fonction de la position courante du centre de masse et des pieds.

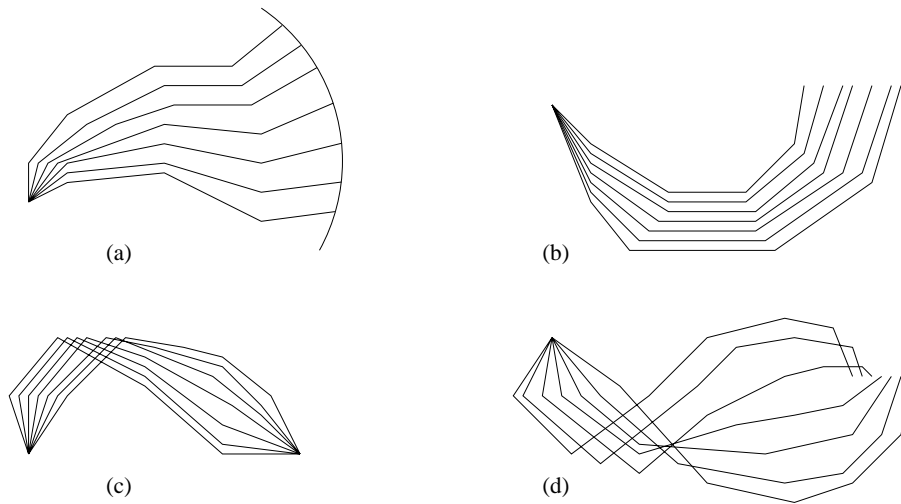


FIG. 3.6: Exemples de tâches [BMT96] : (a) et (b) Tâche principale seulement (norme minimale). (c) L'espace nul est illustré par un vecteur nul de la tâche principale. (d) La même tâche que (b) mais avec une certaine tâche secondaire.

### 3.2.5 Discussion

Les techniques de cinématique reposent sur une certaine compréhension des mécanismes de base du mouvement de marche. Au moins d'un point de vue purement descriptif comparées à la conception directe de positions clés. La qualité du mouvement repose alors sur la qualité du modèle plutôt que sur l'habileté de l'animateur.

Les faiblesses de la cinématique viennent du fait que ne sont prises en compte ni les forces ni la dynamique de la structure. La plupart des inconvénients de l'animation de personnages par cinématique proviennent du processus d'interpolation géométrique utilisé pour la génération des positions intermédiaires. Les trajectoires angulaires sont en effet calculées indépendamment pour chaque articulation alors que les mouvements de celles-ci sont fortement couplés dans la réalité. De plus l'interpolation peut filtrer la dynamique intrinsèque de la locomotion conduisant à une perte de réalisme. Aussi n'y a-t-il pas de moyen aisé pour générer un mouvement de marche sur des terrains accidentés à partir de ces approches. L'adaptation de la marche à ce type de terrain doit alors être explicitement décrite lors de la conception du modèle.

## 3.3 Méthodes incorporant la dynamique

### 3.3.1 Définitions

La simulation dynamique permet de modéliser des objets en tant que masses qui bougent sous l'influence de forces et de couples comme c'est expliqué dans [Wil91, Gre91]. Le mouvement est prédit par les équations dynamiques de mouvement. Ces équations expriment l'influence des forces et des couples sur les mouvements des masses. Les équations

dynamiques du mouvement d'un corps sont :

$$\sum \vec{F} = m\ddot{x} \quad (3.5)$$

$$\sum \vec{C} = I\dot{\omega} + \omega \wedge I\omega \quad (3.6)$$

où  $\sum \vec{F}$  représente les forces  $\sum \vec{C}$  les couples  $m$  la masse  $I$  la matrice d'inertie  $\ddot{x}$  l'accélération cartésienne  $\omega$  la vitesse angulaire et  $\dot{\omega}$  l'accélération angulaire.

Tout comme la cinématique il existe deux catégories d'animation dynamique : la dynamique directe et la dynamique inverse.

### a) Dynamique directe

La *dynamique directe* consiste à résoudre les équations (3.5) et (3.6) pour trouver les accélérations à partir des forces et des couples. Ensuite une intégration est réalisée afin de trouver les nouvelles vitesses et positions. Par exemple on peut utiliser la méthode d'intégration d'Euler :

$$\begin{aligned} f'(t+dt) &= f'(t) + f''(t)dt \\ f(t+dt) &= f(t) + f'(t)dt \end{aligned}$$

Une méthode indépendante du pas de temps et aussi du sens du temps en avant ou en arrière et par conséquent plus intéressante consiste à utiliser l'intégration suivante :

$$\begin{aligned} f'(t+dt) &= f'(t) + f''(t)dt \\ f(t+dt) &= f(t) + \frac{1}{2}(f'(t) + f'(t+dt))dt \end{aligned}$$

Appliquée au cas du mouvement cela donne :

$$\dot{x}(t+dt) = \dot{x}(t)dt + \ddot{x}(t)dt \quad (3.7)$$

$$x(t+dt) = x(t) + \frac{1}{2}(\dot{x}(t) + \dot{x}(t+dt))dt \quad (3.8)$$

$$\omega(t+dt) = \omega(t) + I^{-1}(C(t) - \omega(t) \wedge I\omega(t))dt \quad (3.9)$$

$$R(t+dt) = \left( \mathbb{I} + \frac{1}{2}(\tilde{\omega}(t) + \tilde{\omega}(t+dt))dt \right) R(t) \quad (3.10)$$

où  $R$  est l'orientation et  $\tilde{\omega}$  la matrice adjointe représentant le produit vectoriel avec le vecteur  $\omega$ <sup>1</sup>. On remarque que les équations (3.7) et (3.9) sont données par la première méthode d'intégration et non pas la seconde car les accélérations  $\ddot{x}(t+dt)$  et  $\dot{\omega}(t+dt)$  ne sont pas encore connues. Finalement les nouvelles positions obtenues donnent une description cinématique du mouvement pour animer les objets. Des mouvements complexes et réalistes peuvent être générés avec la spécification de la part de l'utilisateur de l'état initial du système  $(x, \dot{x}, R, \omega)$ .

---

<sup>1</sup> soit  $\omega = \begin{pmatrix} p \\ q \\ r \end{pmatrix}$  alors  $\tilde{\omega} = \begin{pmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{pmatrix}$  et  $\omega \wedge a = \tilde{\omega}.a$

### b) Dynamique inverse

La *dynamique inverse* consiste à résoudre les équations (3.5) et (3.6) pour trouver les forces et les couples à partir des accélérations. Ces accélérations sont calculées au préalable à partir des positions et des vitesses qui définissent le mouvement du corps (cf. Figure 3.7).

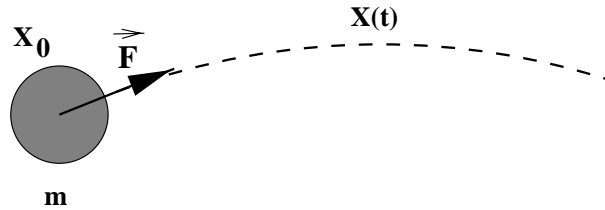


FIG. 3.7: Balle de masse  $m$  animée : a) par dynamique directe : une fois la position initiale  $X_0$  et la force  $F$  spécifiées, la trajectoire  $X(t)$  de la balle est entièrement définie ; b) par dynamique inverse : une fois le mouvement  $X(t)$  spécifié, on peut calculer la force permettant d'engendrer ce mouvement physiquement.

### 3.3.2 Animation par dynamique directe

#### a) Simulation de chaînes articulées

Pour calculer le mouvement d'une structure articulée une méthode récursive dont une description détaillée peut être trouvée dans [AG85] est habituellement employée. La méthode consiste à calculer les équations de la dynamique (3.5) et (3.6) pour chaque segment de la structure hiérarchique ainsi qu'une relation de dépendance entre l'accélération de chaque segment fils et de son père. En remontant dans la hiérarchie de segments les relations de dépendance des accélérations père-fils on calcule l'accélération de la racine en translation et en rotation. Puis on redescend pour obtenir celle de chacun des fils. On calcule ensuite par un schéma d'intégration les nouvelles positions et vitesses à partir des nouvelles accélérations et des positions et vitesses à l'instant précédent.

Cette résolution basée sur un double parcours de la hiérarchie n'est pas exempte de problèmes. Tout d'abord les objets doivent nécessairement être des chaînes ouvertes c'est-à-dire sous forme d'arborescence sans boucle fermée. Or pendant le double support le bipède est une chaîne fermée. De plus elle est instable numériquement à cause de l'accumulation des différences pendant la remontée des informations vers la racine puis de la sommation des corrections qui sont redescendues aux feuilles. Enfin cette approche par dynamique n'offre aucun contrôle sur l'animation et surtout aucune aide à l'utilisateur quant aux forces et couples à appliquer sur chaque segment pour réaliser un mouvement désiré.

#### b) Simulation par contraintes

Des approches permettent de résoudre les contraintes pour des animations de structures complexes. Les segments de la structure articulée sont considérés comme des objets

indépendants  $\Gamma$  soumis à des contraintes géométriques simples (cf. Figure 3.8). [BB88] calculent sur chaque segment les forces nécessaires à la satisfaction des contraintes. Ils se servent de la mesure de l'erreur de déviation par rapport à la contrainte pour trouver les forces et les couples de contraintes. Ensuite  $\Gamma$  les accélérations correspondantes sont intégrées pour trouver les nouvelles positions et vitesses des segments. [GG94] résolvent localement les contraintes par des réglages itératifs de petits déplacements et [Fau98] utilise des contraintes de déplacement linéarisées avec une résolution globale et itérative. Ces méthodes permettent ainsi d'éviter l'intégration des forces au cours des itérations nécessaires à satisfaire les contraintes. Grâce à ce type d'approches  $\Gamma$  des structures avec boucles fermées peuvent être animées.

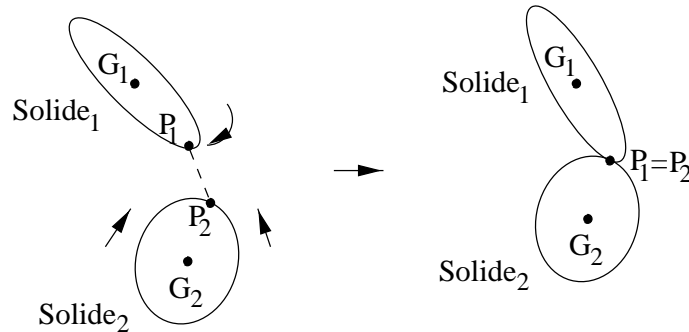


FIG. 3.8: Exemple de contrainte : contrainte point-point.

### c) Discussion

L'approche dynamique est très difficilement applicable à la marche d'un personnage articulé. En effet elle comporte deux problèmes majeurs liés. Le premier problème est le contrôle du mouvement parfois complexe. La dynamique directe est efficace pour prédire le mouvement d'objets inertes  $\Gamma$  mais cette technique ne convient pas pour l'animation d'êtres réactifs  $\Gamma$  qui sont décrits comme des systèmes actifs. En effet  $\Gamma$  pour de tels systèmes  $\Gamma$  les forces développées par les muscles varient au cours du temps  $\Gamma$  et ne sont pas connues a priori. On ne peut donc pas utiliser la dynamique directe seule pour calculer des mouvements complexes comme la marche de personnages articulés. De la même manière  $\Gamma$  les couples articulaires pour amener le modèle à faire un pas ne sont pas connus  $\Gamma$  et rien ne dit si le résultat visuel sera une marche naturelle.

Le second problème est le développement de techniques numériques pour résoudre les équations de mouvement. Les méthodes numériques qui résolvent les équations différentielles ordinaires ont un bon comportement tant que ces équations sont linéaires. Des problèmes surgissent dès que des non-linéarités  $\Gamma$  des discontinuités  $\Gamma$  des raideurs importantes apparaissent dans les équations différentielles. Or  $\Gamma$  les structures articulées sont définies par des systèmes non linéaires. Et le problème se complique encore lorsqu'elles constituent des boucles fermées. Ceci entraîne l'instabilité de certaines méthodes numériques  $\Gamma$  qui se comportent de manière inconsistante.

### 3.3.3 Animation par dynamique inverse

#### a) Principe : Dynamique directe et dynamique inverse

En utilisant le formalisme Lagrangien [IC87] le calcul du mouvement revient à calculer le vecteur de forces  $F$  et des couples  $C$  à partir de  $\ddot{X}$  où  $X$  représente le vecteur de coordonnées cartésiennes et de la matrice d'inertie  $M$ . Cette matrice dépend des positions relatives des segments et par conséquent elle change au cours du temps. Il faut donc soit l'inverser à chaque pas de temps soit utiliser une technique de type "descente de gradient" qui met à jour la solution incrémentalement. La dynamique inverse peut aisément être combinée avec la dynamique directe [IC87][IC88]. Certains éléments du mouvement d'un personnage articulé sont connus et permettent de déduire le reste du mouvement. Par exemple une valeur est donnée pour certaines inconnues  $\ddot{X}$  alors que certains composants de  $F$  deviennent inconnus. Cela mène à des animations dont le mouvement de certains degrés de liberté est spécifié par l'utilisateur alors que le mouvement d'autres degrés de liberté est automatiquement calculé à partir des forces appliquées (cf. Figure 3.9). [FDCGM97] utilisent la dynamique inverse pour analyser les forces et les couples requis pour un mouvement donné afin de savoir ceux qu'il faudra spécifier pour obtenir ce mouvement.

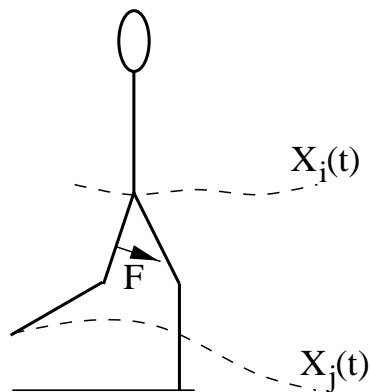


FIG. 3.9: Combinaison de dynamique directe et dynamique inverse : par exemple, le mouvement de la hanche droite  $X_i(t)$  et celui du pied droit  $X_j(t)$  sont donnés, tandis que le mouvement de la cuisse est produit par la force  $F$ .

Une amélioration introduite par [BB88] permet d'éviter le calcul coûteux de la matrice  $M$ . Chaque segment est simulé indépendamment à chaque pas de temps. Des forces supplémentaires sont ensuite calculées pour garantir les contraintes aux articulations. Ces forces restaurent les contraintes articulaires et un nouveau pas de temps peut être simulé. L'ajout de dynamique inverse est réalisé de la même manière que dans [IC87][IC88] : l'utilisateur contrôle certains segments en spécifiant leur mouvement et les segments restants connectés à ces segments contrôlés sont directement influencés par eux aussi leur mouvement est calculé par simulation.

### b) Discussion

La méthode basée sur la dynamique comporte toujours le problème des systèmes descriptifs : aucun réalisme n'est garanti puisque les mouvements, accélérations et forces de ces systèmes réactifs ne sont pas connus et sont donc décrits par l'utilisateur. De plus, la résolution du système matriciel est coûteuse. Elle nécessite de surcroît, pour chaque pas de temps, la réécriture des équations du système, excepté avec la méthode de [BB88].

### 3.3.4 Animation hybride cinématique et dynamique

Les modèles présentés dans le paragraphe précédent reposent sur l'animation physique d'objets articulés grâce aux équations de la dynamique. L'avantage est de ne spécifier qu'un nombre réduit de paramètres pour une animation donnée. Toutefois, les objets animés par ces modèles sont des objets inertes soumis uniquement à des actions externes. Pour tenir compte des actions internes que peuvent engendrer les muscles ou moteurs des personnages, des méthodes de contrôle ont été proposées. Elles permettent de fournir un outil automatique pour calculer au cours du temps les forces et les couples exercés par ces muscles ou moteurs afin de réaliser une action donnée.

Deux types de méthodes émergent. Les premières sont des méthodes d'optimisation des trajectoires qui calculent globalement le mouvement par rapport aux contraintes et à des objectifs à minimiser donnés par l'utilisateur. Les secondes sont des méthodes basées sur des contrôleurs : ils permettent d'analyser l'état du système et calculent les changements à effectuer sur celui-ci pour obtenir le mouvement désiré.

### a) Méthodes d'optimisation des trajectoires

La combinaison de la cinématique directe et de la cinématique inverse permet à des parties de personnage articulé de suivre des trajectoires pré-définies par positions clés, mais elle ne fournit aucune aide quant à la définition même de ces trajectoires. [WK88] proposent une nouvelle formulation du problème, appelée *contraintes spatio-temporelles*. Il s'agit de résoudre un problème d'optimisation sous contraintes pour trouver un mouvement physiquement valide qui satisfait les contraintes de "quoi faire" et optimise le critère de "comment le faire". Cela revient à définir des positions entre lesquelles il faut trouver un chemin (cf. Figures 3.10 et 3.11).

#### *Contraintes spatio-temporelles*

Il s'agit de contraintes au cours du temps sur le mouvement. Ces contraintes sont sur les positions et vitesses initiales, finales et intermédiaires. Elles codent directement les buts du mouvement, alors que les contraintes qui limitent les forces musculaires ou qui empêchent les interpénétrations définissent les propriétés de la situation physique. Toutes ces contraintes sont pré-définies avant le début de l'animation. Pendant le processus d'animation, l'équation de mouvement (3.5) fournit une contrainte reliant les fonctions de force et de position. Celle-ci doit être satisfaite à chaque instant.



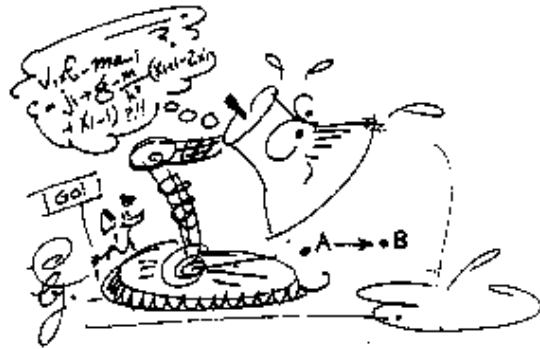


FIG. 3.10: *Contraintes spatio-temporelles : vision de la dessinatrice Laura Green (1988, tiré de [WK88]).*

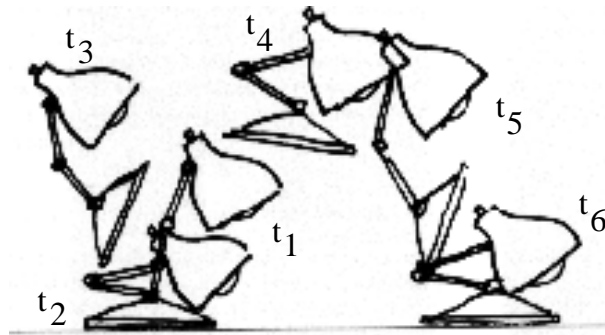


FIG. 3.11: *Contraintes spatio-temporelles : définition de postures à des instants donnés, pour donner les étapes importantes (buts) dans le mouvement de saut de Luxo [Las87].*

### Optimisation sous contraintes

Le système recherche un mouvement possible étant donnés les objectifs définis par l'utilisateur et qui soit optimal selon un critère à déterminer. En général un objectif est défini par une fonction de coût  $R$  à minimiser. Cette fonction est souvent définie comme étant la somme des carrés des forces articulaires  $f$  en fonction du temps :

$$R = \int_{t_0}^{t_1} f^T f(t) dt$$

### Discussion

La résolution du problème d'optimisation sous contraintes produit un compromis entre un mouvement valide physiquement et qui satisfait les contraintes données par l'animateur et un mouvement optimal au sens où il utilise le moins d'énergie possible. Il existe une variété de solutions-algorithmes standard pour résoudre les problèmes d'optimisation sous contraintes [Coh92].

Néanmoins ces méthodes souffrent de plusieurs limitations dans le cadre d'une utilisation en robotique. Tout d'abord par opposition aux systèmes qui simulent le mouvement par pas successifs directs dans le temps l'animation par optimisation nécessite la planification de séquences entières de mouvements. Les scènes sont conçues pour avoir des évènements particuliers qui auront lieu à des instants spécifiés. Le système fournit donc un contrôle sur toute l'animation tout en incorporant les principes de la dynamique. Or en robotique en général on ne connaît pas a priori les trajectoires qui seront suivies par un robot dans son environnement.

De plus étant donné que chaque contrainte dans l'animation doit être spécifiée a priori les collisions et les contacts entre objets sont modélisés en tant que contraintes sur les

positions. Ainsi il n'y a pas de module de détection et de réponse automatique aux collisions pendant l'animation. D'autre part, puisque la dynamique est traitée comme une contrainte entre des inconnues et que le système aboutit à un compromis entre toutes les contraintes pour finalement trouver une solution, les lois dynamiques peuvent ne pas être complètement respectées.

Un problème supplémentaire est que si l'on désire changer quoique ce soit à l'animation, il faut recalculer tout le mouvement. C'est un processus lent qui rend la mise au point délicate. Enfin, la spécification n'est pas interactive, ce qui rend difficile l'intégration de cette technique dans un système de conception d'animations. Et comme les buts et les contraintes peuvent ne pas être linéaires, l'utilisateur n'a pas de garantie quant à la convergence du processus numérique vers une solution acceptable. De plus, la complexité importante en calcul du problème limite la longueur des séquences d'animation.

### *Améliorations*

Une première amélioration est de se servir de la localité dans la résolution des contraintes spatio-temporelles [Coh92]. Elle s'appuie sur des fenêtres spatio-temporelles conçues interactivement, ce qui autorise une solution partielle en temps et en espace de l'animation. Une fenêtre est définie sur un sous-ensemble de degrés de liberté et sur un sous-intervalle de temps. Un processus d'optimisation itératif recherche des fonctions qui décrivent le mouvement de chaque degré de liberté dans la fenêtre et qui minimisent un but tout en maintenant un ensemble de contraintes pondérées. La solution pour chaque fenêtre spatio-temporelle représente une solution partielle de l'animation complète. La solution globale, continue en temps et espace, est obtenue en combinant les solutions partielles des fenêtres. Grâce aux retours visuels et numériques sur la progression de l'animation, l'animateur peut ajouter de nouvelles contraintes, augmenter ou diminuer le poids des contraintes antérieures, ou modifier les solutions intermédiaires. Ainsi, l'animateur interagit et guide l'optimisation dans l'optique de la faire converger vers une solution acceptable. Une seconde amélioration, proposée dans [LGC94, MA96], concerne l'évolution des degrés de liberté, qui est reformulée dans une représentation d'ondelettes hiérarchiques. Cette représentation offre une réduction significative du coût de calcul.

### *Autre application*

Hormis l'application directe de cette technique pour obtenir des mouvements, une autre application consiste à modifier des mouvements originaux pour les adapter à de nouvelles morphologies de personnages. Ainsi, [Gle98] propose dans sa méthode de réajustement de mouvement à de nouveaux personnages, de définir le mouvement dans l'espace articulaire - puisque les angles étant sans dimension, ils ne varient pas d'un personnage à un autre - et de changer l'échelle (la taille) du personnage. Cependant, cette modification peut entraîner des erreurs indésirées, telles que la pénétration des pieds dans le sol ou, au contraire, la marche à plusieurs centimètres au-dessus du sol. Des contraintes sont alors définies pour rajuster le mouvement du personnage à l'environnement considéré. Le nouveau mouvement est alors calculé par la technique d'optimisation de trajectoires, en optimisant le mouvement à effectuer tout en tenant compte des contraintes ajoutées.

## b) Contrôleurs

Une seconde méthode pour calculer au cours du temps les forces et les couples exercés par les muscles d'un personnage  $\Gamma$  est basée sur les contrôleurs. Cette méthode permet d'animer un personnage humain synthétique  $\Gamma$  en tenant compte automatiquement des effets des interactions entre le personnage et l'environnement virtuel. Le problème principal est alors de trouver les forces actionnées  $\Gamma$  qui modélisent l'action des muscles de l'humain virtuel  $\Gamma$  et qui lui font accomplir un mouvement désiré.

Les approches par contrôleurs sont généralement basés sur des modèles à couches  $\Gamma$  ou modèles hiérarchiques  $\Gamma$  de locomotion humaine. Les contrôleurs permettent de traiter la dynamique intrinsèque de la locomotion humaine  $\Gamma$  tout en considérant des connaissances empiriques et biomécaniques sur la marche humaine.

### *Principe*

Un contrôle hiérarchique calcule les actions à faire réaliser par le personnage articulé. Avant l'animation proprement dite  $\Gamma$  l'animateur spécifie le comportement qu'il souhaite simuler  $\Gamma$  en fournissant un contrôle de haut niveau. Le comportement peut être la vitesse de course désirée  $\Gamma$  le chemin à suivre  $\Gamma$  des informations événementielles telles que la date et la hauteur du saut à effectuer. L'animateur utilise des techniques classiques de positions clés ou de cinématique pour définir un mouvement désiré.

Des contrôleurs de bas niveau sont responsables de la régulation du comportement des articulations  $\Gamma$  mais peuvent aussi s'occuper  $\Gamma$  par exemple  $\Gamma$  de la stabilisation de la posture  $\Gamma$  du maintien du cycle de locomotion  $\Gamma$  du contrôle de la vitesse et de la direction du mouvement. Basés sur l'état courant du modèle et sur l'état que l'on désire voir atteindre  $\Gamma$  ces contrôleurs calculent les forces et les couples à appliquer sur chacun des actionneurs localisés sur les articulations pour atteindre le but. Ces forces peuvent être calculées au moyen de contrôleurs de type Proportionnel-Intégral-Dérivé (PID) (cf. Figure 3.12)  $\Gamma$  qui expriment le comportement dynamique des articulations :

$$f = -k_p(q - q^*) - k_d(\dot{q} - \dot{q}^*) - k_i \int (q - q^*) dt \quad (3.11)$$

où  $k_p$   $\Gamma$   $k_i$  et  $k_d$  sont les gains proportionnel  $\Gamma$  intégral et dérivé respectivement  $\Gamma$   $(q, \dot{q})$  est l'état courant du système (position et vitesse)  $\Gamma$   $(q^*, \dot{q}^*)$  est l'état désiré du système. Ces contrôles articulaires ont le même comportement dynamique qu'un mécanisme ressort-amortisseur avec longueur au repos programmable. Le terme  $k_i$  permet de corriger une erreur statique. L'équation (3.11) admet une solution générale de la forme  $f = Ae^{\omega t}$ . Le discriminant  $\Delta = k_p^2 - 4k_i k_d$  gouverne le type d'amortissement : si  $\Delta < 0$   $\Gamma$  le système oscille  $\Gamma$  si  $\Delta = 0$   $\Gamma$  le système est optimal  $\Gamma$  et si  $\Delta > 0$   $\Gamma$  le système est sous-optimal. En synthèse d'images  $\Gamma$  il est courant de négliger la correction d'une force systématique  $\Gamma$  aussi  $k_i$  est généralement pris à zéro.

Finalement  $\Gamma$  le modèle permet de calculer le mouvement résultant de l'application des forces et des couples trouvés par les contrôleurs de bas niveau  $\Gamma$  à chaque pas d'intégration.

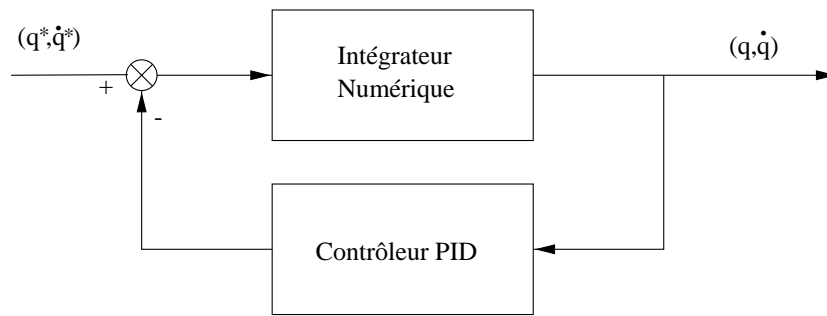


FIG. 3.12: Contrôleur avec boucle de rétro-action.

Ce modèle consiste en les équations de mouvements des corps rigides du système articulé, les modèles d'actionneurs (et de capteurs pour des systèmes plus complets), les équations de force avec le sol lorsque l'interaction avec le sol est prise en compte, et un intégrateur numérique qui produit le mouvement en fonction du temps. Le résultat est affiché à l'écran (cf. Figure 3.13).

Ainsi, la spécification du mouvement souhaité offre un contrôle de haut niveau pour l'animateur, alors que le réalisme physique est obtenu grâce à l'utilisation des contrôleurs dynamiques de bas niveau et à l'intégration des forces, qui incluent les forces de collisions et de frottement avec l'environnement. Des contrôleurs intermédiaires sont souvent ajoutés pour permettre des raffinements ainsi que la coordination des mouvements.

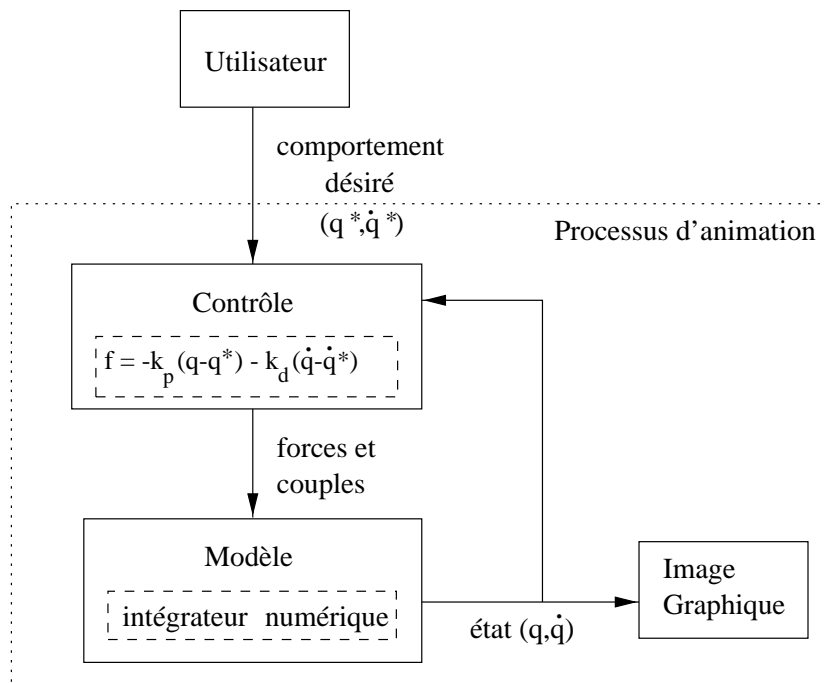


FIG. 3.13: Algorithme pour l'animation par contrôleur.

### Graphe de contrôle de poses

[vdPKF94] nomment *graphe de contrôle de poses* une machine d'états finis qui forme un contrôleur de mouvement pour un personnage articulé. Le graphe décrit un mode de locomotion (cf. Figure 3.14). Un mouvement est obtenu en passant d'une pose à l'autre dans le graphe. En effet, chaque état du graphe est associé à une pose ou posture du personnage, c'est-à-dire sa forme géométrique, et non sa position dans le repère du monde. La transition entre deux états adjacents détermine le mouvement entre les poses correspondantes. Cette transition est réalisée grâce à des contrôleurs PD placés en chaque articulation, qui calculent les forces et les couples requis pour conduire le personnage articulé vers la pose spécifiée dans le prochain état du graphe.

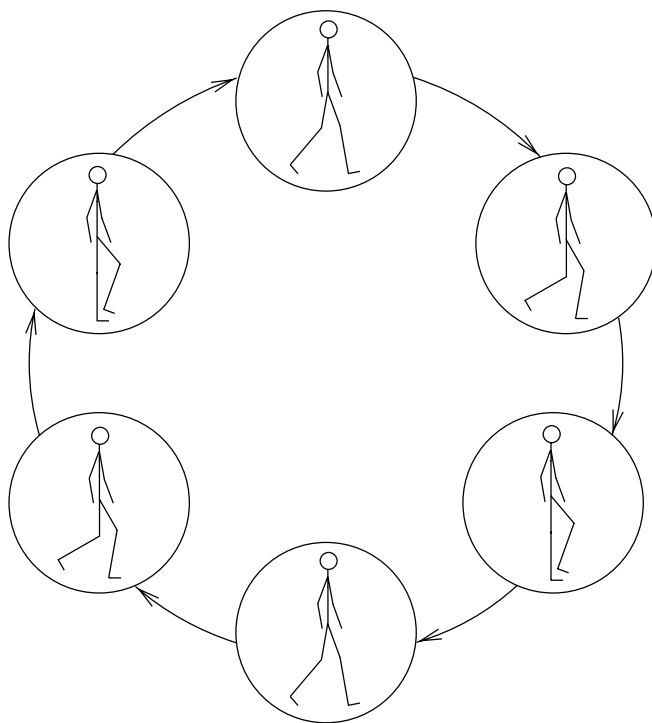


FIG. 3.14: Exemple de graphe de contrôle de poses pour la marche humaine.

Ce concept de graphe de contrôle de poses est particulièrement adapté aux mouvements cycliques, tels que la marche : le mouvement est généré par un graphe cyclique qui décrit un cycle du mouvement considéré. Cette approche a été largement étudiée pour concevoir des contrôleurs particuliers de mouvements de types variés [RH91, HWBO95, KKB96, vdPKF94, LvdPF96]. [RH91] définissent un contrôleur pour animer la course dynamique de créatures bipèdes et quadrupèdes, en régulant la vitesse de course, en organisant l'utilisation des jambes et en maintenant l'équilibre. D'autres approches [HWBO95, Hod96] permettent l'animation de comportements athlétiques dynamiques tels que la course, le cyclisme, le saut de cheval et le plongeon, en contrôlant les caractéristiques appropriées des mouvements considérés. [vdPKF94, LvdPF96] définissent des graphes de poses pour la locomotion humaine.

Les graphes de contrôle de poses cycliques fournissent un moyen de représenter et contrôler les mouvements périodiques en boucle ouverte c'est-à-dire sans considérer l'interaction possible avec l'environnement. Cette technique peut être étendue au contrôle en boucle fermée qui se sert d'informations sur l'environnement données par des capteurs [BBG<sup>+</sup>94]. Par exemple les transitions entre les poses peuvent être déclenchées lorsqu'un contact est détecté [LvdPF96].

### *Réglages des contrôleurs*

#### *– Par ajustement manuel*

Ces techniques semblent nécessiter une longue période de réglage manuel interactif car les paramètres sont trouvés par tâtonnements puis ajustés [HWBO95]. En effet ce sont des lois empiriques qui déterminent les angles pour les poses des machines d'états finis qui donnent une description de haut niveau du comportement à simuler. Ainsi une stratégie de contrôle en boucle ouverte avec ajustement fin de paramètres de contrôle est nécessaire. Ce processus rend difficile la possibilité de réutilisation des paramètres pour le réajustement à un personnage de morphologie différente car après avoir changé d'échelle certains paramètres du personnage et du contrôle associé il est à nouveau nécessaire de régler manuellement ces nouveaux paramètres [HP97].

#### *– Par génération automatique de contrôleurs*

Une alternative aux contrôleurs réglés manuellement est la génération automatique de ceux-ci. Certaines de ces approches tentent de partir d'aucune donnée initiale : l'utilisateur définit un personnage articulé et une fonction objectif à optimiser par exemple aller le plus loin possible en un minimum de temps [vdPKF94]. Le système génère et optimise automatiquement un ensemble de contrôleurs qui engendrent parfois des modes de locomotion très différents et surprenants.

[vdPF93] modélisent les contrôleurs par des réseaux de capteurs-actionneurs c'est-à-dire par des graphes orientés arborescents avec connexions pondérées. Ces graphes calculent les couples actionneurs à partir des nœuds internes liés à des données de détecteurs telles que des informations sur l'environnement comme le contact. Les paramètres internes du système sont d'abord générés aléatoirement puis optimisés selon un critère donné. [NM93] utilisent une approche similaire dans laquelle les valeurs des paramètres du contrôleur sont recherchées avec des algorithmes génétiques. [Sim94] va plus loin dans l'utilisation des algorithmes génétiques en calculant même l'évolution de la morphologie de la créature qui convient le mieux à un mode de locomotion donné.

#### *– Discussion*

Ces types de méthodes nécessitent beaucoup d'essais et par conséquent ont besoin de nombreuses ressources de calcul. De plus il y a peu de chance pour que le mouvement résultant ressemble à des mouvements connus. D'autre part la plupart des

méthodes ont été seulement utilisées dans des cas 2D. Or l'application de méthodes de contrôle dynamique pour des personnages 3D complique le problème du maintien de l'équilibre.

### Définition de contrôles intermédiaires

Les contrôles intermédiaires permettent de réduire le nombre de paramètres de contrôle que l'animateur doit ajuster et rendent le contrôle plus intuitif. Des exemples de contrôles intermédiaires sont présentés ci-dessous.

#### – Cas de la course

[RH91] définissent des contrôles intermédiaires pour contrôler la course d'un bipède. Ce type de locomotion est plus simple que la marche. En effet la course comporte seulement des phases de simple support et des phases de vol. Or dans la marche la difficulté de calcul provient des phases de double support à cause des boucles cinématiques fermées. Cependant on peut déjà remarquer la complexité du contrôle nécessaire.

Le premier contrôle proposé est le *contrôle du saut* qui fait faire un pas aux jambes en changeant de support. Le modèle des actionneurs est équivalent à un ressort-amortisseur :

$$f = k(x - x_r) + b\dot{x}$$

avec  $f$  la force actionnée,  $k$  la constante du ressort,  $b$  la constante d'amortissement,  $x$  la longueur du ressort,  $x_r$  la longueur au repos du ressort. Le contrôle de la longueur au repos du ressort permet d'injecter ou d'enlever l'énergie au système afin d'amorcer l'oscillation ou de la moduler ou de l'arrêter.

Le deuxième contrôle est le *contrôle de vitesse* qui fournit un équilibre pour régler la vitesse de course. Les systèmes à pattes sont assimilables à des pendules inversés. En effet ils penchent et accélèrent chaque fois que le point de support est déplacé par rapport à la projection du centre de masse. Aussi le système de contrôle calcule la position du pied désirée :

$$x_{fh,d} = \frac{\dot{x}T_s}{2} + k_{\dot{x}}(\dot{x} - \dot{x}^*)$$

avec  $x_{fh,d}$  le déplacement vers l'avant du pied à partir de la projection du centre de gravité,  $\dot{x}$  la vitesse vers l'avant,  $\dot{x}^*$  la vitesse vers l'avant désirée,  $k_{\dot{x}}$  un gain et  $T_s$  la durée prédite pour la prochaine période de support. Le système de contrôle déplace le pied du point neutre d'une distance proportionnelle à la différence de la vitesse réelle et celle désirée. Une fois  $x_{fh,d}$  trouvé une transformation cinématique détermine les angles articulaires qui positionneront le pied comme spécifié.

Le troisième contrôle est le *contrôle de posture* qui maintient le corps dans une posture droite. Les couples de contrôle de posture sont générés par un contrôle linéaire de type PD :

$$\tau = -k_p(\phi - \phi^*) - k_d(\dot{\phi})$$

$\phi$  étant l'angle du corps et  $\dot{\phi}$  la vitesse angulaire du corps  $\phi^*$  l'angle du corps désiré  $k_p$  et  $k_v$  des gains.

Chacune de ces mises en œuvre utilise une machine d'états finis pour suivre le comportement à venir du modèle et pour synchroniser les actions de contrôle avec le comportement de course.

– *Cas de la marche*

Dans le système de [Ko94][KB96] la dynamique est traitée en tant que contrainte. Le mouvement est d'abord calculé avec un modèle cinématique et la dynamique inverse sert à calculer les couples nécessaires pour engendrer un mouvement. Puis une opération de post-traitement vérifie la validité physique du mouvement par contrôle de l'équilibre et du confort. Le mouvement est corrigé au besoin en ajustant manuellement les coefficients qui lient les positions cinématiques du personnage articulé aux résultats de dynamique inverse. Par exemple un acteur synthétique ajuste sa posture automatiquement en penchant son bassin lorsqu'il transporte un poids lourd.

Dans ce système un premier contrôle est défini : le *contrôle de l'équilibre* basé sur le ZMP (*Zero-Moment Point*) [VBSD90]. L'équilibre statique peut être obtenu en gardant la projection de centre de masse dans le polygone de sustentation du personnage même si une seule jambe est en appui. Pour la locomotion il est nécessaire de considérer l'équilibre dynamique car l'effet de l'inertie n'est pas négligeable. Lors d'une phase de simple support même si le corps semble être en appui sur tout le pied on peut trouver un point sur la plante du pied pour lequel le moment est nul. De façon similaire pendant la marche il est un point appelé ZMP où les couples exercés devraient être nuls. Le ZMP n'est pas un point fixe il effectue une translation du talon jusqu'au bout de l'orteil pendant le support. Il est approximé par une fonction monotone qui avance du talon à l'orteil pendant la durée du support. Normalement le moment du ZMP doit rester nul tout le temps. Si le résultat de la dynamique inverse indique une valeur non nulle cela veut dire qu'un tel couple aurait dû être exercé pour le mouvement. Cette valeur est interprétée ici comme la mesure du déséquilibre à ce moment. Le couple au ZMP est appelé le vecteur d'équilibre  $b$ . Soit  $b^T$  le vecteur obtenu par la rotation de  $b$  de 90 degrés dans le sens indirect. Le contrôle de l'équilibre sur la translation du bassin  $\Delta t_{\text{bassin}}$  et sur la rotation du torse  $\Delta r_{\text{torse}}$  est alors :

$$\begin{aligned}\Delta t_{\text{bassin}} &= \alpha b^T \\ \Delta r_{\text{torse}} &= \beta b\end{aligned}$$

où  $\alpha$  et  $\beta$  sont des coefficients de proportionnalité.



Un second contrôle est ajouté au précédent. Il s'agit du *contrôle du confort* qui compare le couple articulaire exercé avec celui permis qui est préalablement déterminé par les données biomécaniques et qui constitue une contrainte sur l'articulation. Si le couple articulaire est supérieur à ce couple permis la contrainte articulaire est violée et le contrôle met à jour certains paramètres afin de réduire le couple exercé. Cependant une animation d'une marche dynamiquement équilibrée et qui est aussi "confortable" "agréable" au sens où les violations de forces sont évitées peut se montrer quand même un peu différente d'une marche humaine normale.

Pendant la marche les contacts au sol rendent difficile la tâche de calcul des forces et des couples équivalents à l'instant de contact. [KB96] simplifient le problème en admettant un déplacement linéaire du point de contact sur le pied durant la phase de support. Une solution plus précise est basée sur un modèle d'interaction [Fau96]. Ce modèle inclut la détection de collisions et les frottements de Coulomb avec le sol [FDCGM97].

Une autre méthode présentée par [LvdPF96] permet d'ajouter un contrôle en boucle fermée à un mouvement de base qui a été obtenu par un contrôle en boucle ouverte (périodique instable). Cette méthode est basée sur le *contrôle du cycle limite*. Un cycle limite périodique est une courbe fermée dans le diagramme de phase c'est-à-dire une projection du mouvement dans le repère (*position, vitesse*) (cf. Figure 3.15). Le but du contrôle est d'amener activement les mouvements instables vers un cycle limite fixe en faisant une approximation linéaire. Le mouvement est alors représenté par un système dynamique discret :

$$x_{n+1} = g(x_n, u_0 + \Delta u_n)$$

Pour de petites perturbations  $\Delta u_n$  appliquées à chaque cycle le changement d'état résultant  $\Delta x_{n+1}$  a pour approximation d'ordre 1 :

$$\Delta x_{n+1} = J \Delta u_n$$

avec  $J$  le Jacobien calculé pour chaque cycle du mouvement lequel définit la relation entre les perturbations de contrôle appliquées et les changements résultants. Le calcul de la perturbation du contrôle requise pour ramener le système vers le cycle limite désiré est alors :

$$\Delta u_n = J^{-1} \Delta x_{n+1}$$

Le contrôle de cycle limite peut être simplifié en réduisant le nombre de variables de régulation du vecteur d'état. Les mouvements deviennent alors moins contraints. Ainsi ce module de contrôle supplémentaire permet d'adapter les contrôleurs en boucle ouverte en régulant certaines de leurs variables selon des perturbations observées sur le mouvement. Cet ajout automatique de contrôle en boucle fermée produit des mouvements de marche stables et robustes mais qui n'ont toutefois toujours pas la qualité visuelle d'un mouvement humain réel. Toutefois il serait possible d'obtenir un résultat raisonnable en réglant le comportement en boucle ouverte à partir d'un mouvement humain enregistré.

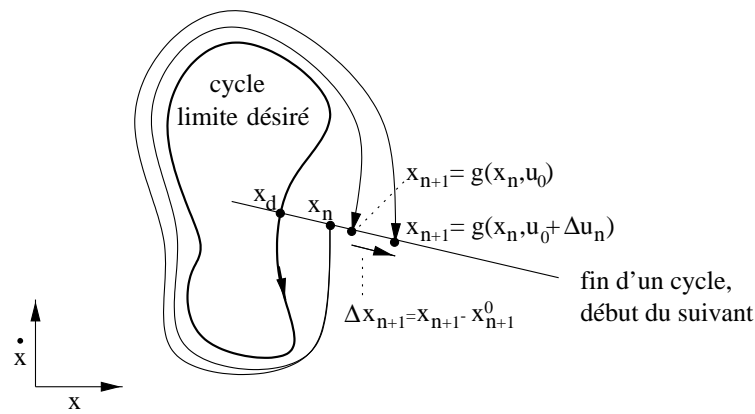


FIG. 3.15: Contrôle du cycle limite.

### 3.3.5 Discussion

L'ajout de connaissances biomécaniques concernant le mouvement humain au module de dynamique inverse  $\Gamma$  permet d'assurer davantage que le mouvement calculé automatiquement est plausible. Si les forces sont calculées proprement et de manière robuste  $\Gamma$  le personnage est capable d'interagir correctement avec son environnement  $\Gamma$  ce qui est essentiel pour un acteur autonome animé dans des applications de réalité virtuelle. Les collisions avec les objets et la marche sur terrain accidenté peuvent être simulées avec ces méthodes  $\Gamma$  offrant ainsi un outil puissant aux animateurs.

Cependant  $\Gamma$  ces techniques basées sur des contrôleurs et la simulation dynamique souffrent d'un coût de calcul élevé par rapport aux techniques de cinématique. Cela rend difficile leur utilisation dans des outils d'animation à caractère interactif. De plus  $\Gamma$  les paramètres mécaniques sont très difficiles à calibrer par rapport à l'anatomie humaine. Par exemple  $\Gamma$  pour les effets d'amorti  $\Gamma$  il n'y a pas de comparaison possible avec les humains  $\Gamma$  à cause de la trop grande complexité des tissus mous humains. Par conséquent  $\Gamma$  souvent les mouvements résultants ne sont pas aussi réalistes que ceux générés manuellement avec la cinématique  $\Gamma$  laquelle offre un contrôle direct sur le mouvement plutôt que d'essayer de modéliser les causes du mouvement. En outre  $\Gamma$  la dynamique n'est pas toujours suffisante pour faire évoluer un personnage articulé  $\Gamma$  celui-ci pouvant notamment tomber. Actuellement  $\Gamma$  ces méthodes sont quelque peu mises de côté au profit des méthodes présentées dans le paragraphe suivant.

## 3.4 Utilisation de mouvements capturés

Actuellement  $\Gamma$  les techniques de capture de mouvements sont en progrès. Elles permettent de fournir rapidement des données de mouvements d'humains. Des techniques sont alors apparues  $\Gamma$  permettant de travailler directement sur des données capturées  $\Gamma$  largement utilisées pour animer des squelettes de corps rigides 3D. Les animateurs ont recours à ces techniques  $\Gamma$  pour réaliser des parties d'animation ou des animations complètes  $\Gamma$  par exemple pour "Star Wars"  $\Gamma$  "Men In Black"  $\Gamma$  "Prince of Egypt" ou "Antz".

### 3.4.1 Capture de mouvements

#### a) Principe

La capture de mouvement consiste à mesurer et enregistrer les actions directes d'une personne ou d'un animal réel afin d'en faire une analyse directe ou différée et un enregistrement. De nos jours cette technique est en particulier utilisée pour produire des animations de personnages 3D. Elle implique une mise en correspondance des mesures sur le mouvement du personnage virtuel. Cette mise en correspondance peut être directe : par exemple le mouvement du bras humain contrôle le mouvement du bras du personnage virtuel ; ou indirecte : par exemple le mouvement de la souris contrôle la direction des yeux et de la tête du personnage virtuel. Différents types de systèmes peuvent être distingués [Mai95].

#### *Systèmes mécaniques*

Les *systèmes mécaniques ou de marionnettes digitales* permettent l'animation de personnages 3D à partir de l'utilisation de plusieurs appareils d'entrée temps-réel (souris, joysticks, data gloves, clavier, boîtes de dialogue ou marionnettes instrumentées ("animatronics") comme pour "La Mutante"). L'information fournie par de tels appareils est utilisée pour contrôler la variation au cours du temps des paramètres de chaque caractéristique animée d'un personnage.

#### *Systèmes optiques de capture de mouvement*

Les *systèmes optiques de capture de mouvement* requièrent des petits dispositifs réfléchissants appelés marqueurs placés sur le corps d'un acteur ainsi que plusieurs caméras réglées sur l'espace de performance. La plupart des systèmes optiques opèrent avec quatre ou six caméras. On attache les marqueurs sur les membres d'une personne et on enregistre la position des marqueurs depuis plusieurs directions différentes. On reconstruit alors la position 3D de chaque point clé à chaque instant.

Le principal avantage de cette méthode est la liberté de mouvement car aucun câble n'est requis. Il existe cependant un problème de taille : l'occlusion de marqueurs par moment puisque les membres effectuent des mouvements et peuvent alors cacher des parties d'autres membres aux caméras et par conséquent les marqueurs placés sur ces membres. Ce problème peut être minimisé par l'ajout de caméras mais cela engendre alors un coût beaucoup plus important. Un second problème provient du manque de distinction automatique des réflecteurs lorsqu'ils sont proches l'un de l'autre pendant le mouvement. Ce problème nécessite du matériel spécialisé et coûteux.

#### *Systèmes magnétiques de capture de mouvement*

Les *systèmes magnétiques de capture de mouvement* nécessitent de la part de l'acteur de porter une combinaison munie d'un ensemble de capteurs capables de mesurer leurs positions relatives à un transmetteur magnétique localisé au centre de la combinaison. Le flot de données des récepteurs vers un ordinateur consiste en des positions et des

orientations 3D pour chaque récepteur. La position et l'orientation de chaque capteur sont alors utilisées pour guider le personnage animé (cf. Figure 3.16). Pour le mouvement d'un corps humain en général une douzaine de capteurs sont utilisés et l'information complémentaire est calculée par cinématique inverse.

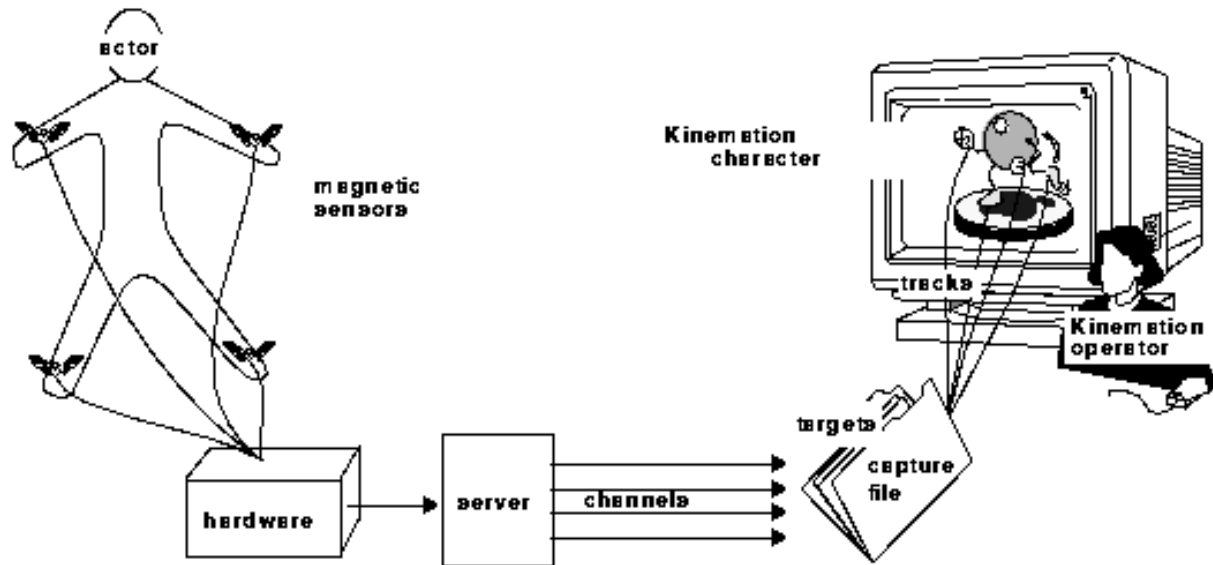


FIG. 3.16: *Processus de capture de mouvements (exemple de Kinemation).*

Cependant ces systèmes présentent des problèmes de “bruit magnétique” et nécessitent d’avoir recours à des environnements en bois sans la moindre trace de fer y compris dans le béton des bâtiments. Un autre problème des capteurs magnétiques est le besoin de synchroniser les récepteurs. D’autre part les petites configurations ne permettent pas d’avoir des mouvements en temps réel car la fréquence d’enregistrement est proportionnelle au nombre  $n$  de capteurs par interface :  $f = \frac{f_0}{n}$ . Pour un capteur usuel la fréquence est  $f_0 = 120Hz$ . Pour  $n = 12$  le mouvement est enregistré à une fréquence  $f = 10Hz$  ce qui donne 10 images par seconde au lieu des 25 requises pour un film de qualité cinéma. Une solution consiste alors à brancher simultanément trois interfaces (ou plus) composée chacune de quatre capteurs (cf. Figure 3.17). La fréquence  $f$  est alors de  $30Hz$  soit 30 images par seconde.

### b) Mouvements de marche capturés

L’utilisation de technologies magnétiques ou optiques permet de stocker les positions et les orientations de points localisés sur le corps humain. Un calcul supplémentaire donne la correspondance entre le squelette synthétique et le squelette réel de manière à adapter les données à de nouvelles morphologies. Le résultat est un mouvement humainement réaliste car il provient directement d’un acteur réel. Cependant la capture de mouvement ne fournit aucun nouveau concept de méthodologie pour l’animation. Pour n’importe quel nouveau mouvement il est à nouveau nécessaire d’enregistrer le mouvement d’un acteur.

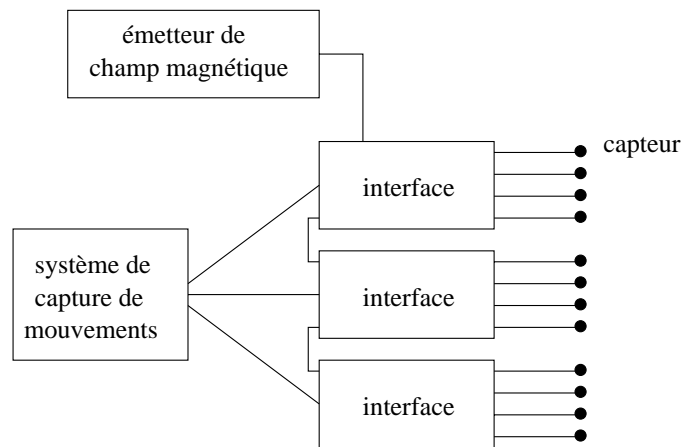


FIG. 3.17: *Système de capture de mouvements, composé de trois interfaces à quatre capteurs.*

Cependant, plusieurs approches [MBT96, MHBT97, BRRP97] proposent des techniques pour adapter les trajectoires capturées à des squelettes synthétiques différents. De tels systèmes traitent aussi des erreurs introduites lors du processus de capture, telles que les approximations numériques, les erreurs de calibration, le bruit électronique, etc. La méthode consiste à retrouver les trajectoires angulaires pour les appliquer au personnage articulé synthétique. Étant données les positions et les orientations des capteurs, un algorithme d'optimisation de cinématique inverse modifié permet de produire les trajectoires articulaires désirées. Le squelette synthétique rejoue alors exactement le mouvement de l'acteur réel. Le fait d'utiliser des trajectoires angulaires repose sur l'hypothèse qu'elles varient très peu avec la morphologie, contrairement aux trajectoires cartésiennes des segments. En effet, si on utilisait ces dernières sur un personnage de taille différente de celle du sujet de l'expérimentation, elles produiraient des artéfacts, tels que la déconnexion des segments.

D'autres approches tentent de modifier le mouvement capturé, dans le but de créer une variété d'animations spécifiques, en particulier pour prendre en compte l'environnement synthétique. Par exemple, lorsque deux acteurs synthétiques interagissent, leur mouvement doit être modifié pour modéliser cette interaction, en traitant des problèmes de contact, de poursuite de trajectoires, etc. Le mouvement est défini par un ensemble de paramètres, sur lesquels on agit par des méthodes spécifiques. Les techniques peuvent aussi être appliquées à tout mouvement calculé à partir d'une des précédentes techniques physiques. Deux types de familles de techniques ont émergé : les déformations de mouvement et le fondu de mouvement.

### 3.4.2 Déformations de mouvement

Les techniques de déformations de mouvement modifient des trajectoires connues, décrites par des positions clés ou par des mouvements capturés, de manière à changer le mouvement. Deux types d'approches sont considérés : l'étude du signal dans le domaine temporel et dans le domaine fréquentiel.

### a) Domaine temporel

[WP95] modifient la trajectoire de référence  $q_i(t)\Gamma$  où  $i$  représente le  $i^e$  paramètre du système. Ils ajustent interactivement la position  $q_i(t)$  des images clés sélectionnées  $\Gamma$  en modifiant le facteur d'échelle  $a(t)$  et en décalant le mouvement de  $b(t)$ . La nouvelle trajectoire  $q'_i(t)$  est alors :

$$\forall i, q'_i(t) = a(t)q_i(t) + b(t) \quad (3.12)$$

Le temps  $t$  peut lui aussi être déformé grâce à une interpolation contrainte basée sur une Cardinal spline  $g$  :

$$t = g(t') \quad (3.13)$$

Ainsi  $\Gamma$  la séquence résultante satisfait les contraintes des nouvelles positions clés par rapport à la structure du mouvement initial (cf. Figure 3.18). De plus  $\Gamma$  deux mouvements peuvent être mélangés en calculant la somme pondérée des deux courbes de mouvements  $q_1(t)$  et  $q_2(t)$  :

$$q'(t) = w(t)q_1(t) + (1 - w(t))q_2(t), \text{ avec } w(t) \in [0, 1]$$

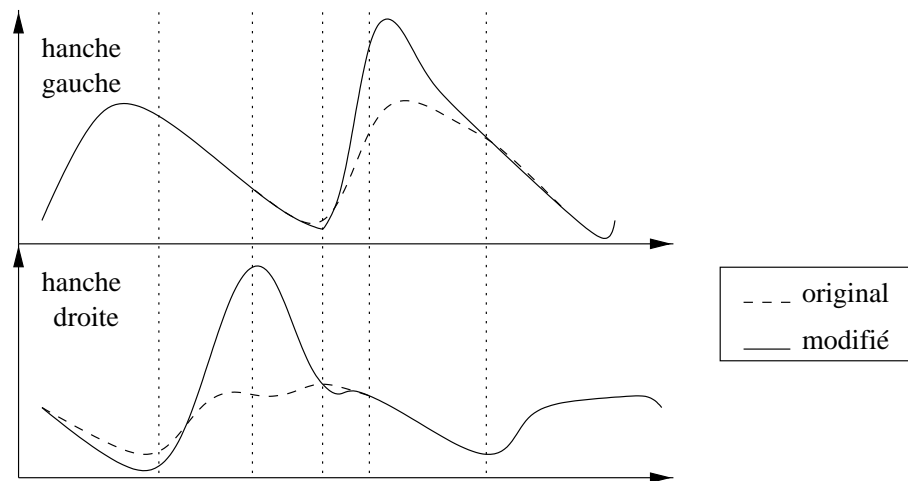


FIG. 3.18: Courbes des mouvements original et déformé pour les articulations des hanches gauche et droite. Les lignes verticales correspondent aux images clés utilisées pour déformer le mouvement.

[KB93] présentent une méthode pour modifier un mouvement de référence enregistré sous forme de séquences de positions clés :

$$\mathcal{Q} = \{(t_i, v_i) | i = 1, \dots, n\}$$

où  $t_i$  est une valeur réelle telle que  $t_i < t_{i+1}\Gamma$  correspondant au temps  $\Gamma$  et  $v_i$  est un vecteur de dimension quelconque de paramètres de mouvements à choisir. En modifiant la morphologie du personnage synthétique  $\Gamma$  de nouveaux paramètres  $(t_i, v_i)$  sont automatiquement calculés pour créer les nouvelles démarches du personnage. Alors que les paramètres ont changé  $\Gamma$  les caractéristiques de la démarche originelle sont préservées.

## b) Domaine fréquentiel

Au lieu d'étudier les trajectoires dans le domaine temporel [BW95] emploient des techniques de traitement du signal et d'images dans le domaine fréquentiel dans le but de réutiliser, de modifier et d'adapter le mouvement humain animé. Il s'agit de créer des bibliothèques de mouvements animés avec un outil d'édition de mouvements de haut niveau et interactif.

Le filtrage multi-résolution est appliqué sur les paramètres de mouvement d'un personnage articulé. Chaque paramètre de mouvement est traité comme un signal en une dimension sur lequel sont calculés les filtres passe-bas ( $G$  : filtre Gaussien) et passe-bande ( $L$  : filtre Laplacien). Les fréquences basses contiennent les allures générales du mouvement tandis que les hautes fréquences contiennent les détails et les subtilités ainsi que le bruit. Le signal du mouvement  $G_0$  peut alors être reconstruit :

$$G_0 = L_0 + L_1 + \dots + L_{bf-1} + G_{bf}$$

où  $L_i$  est le passe-bande de niveau  $i$ ,  $bf = \log_2(m)$  avec  $m$  le nombre d'images clés dans  $G_0$  et  $G_{bf}$  est l'intensité moyenne du signal.

L'algorithme de multi-résolution pour calculer un nouveau mouvement  $G'_0$  se décompose en cinq étapes successives réalisées simultanément pour chaque paramètre du signal du mouvement original  $G_0$ . Tout d'abord une séquence de filtres passe-bas est appliquée sur toutes les bandes de fréquence  $bf$  en convoluant systématiquement le signal avec des noyaux étendus :

$$G_{k+1} = w_{k+1} G_k$$

Ce calcul est rendu rapide en conservant le noyau  $w$  constant et en éliminant les points trop éloignés de chaque point considéré. Ainsi seuls les points proches ont une influence sur le point considéré. Pour un filtre de longueur 5 cela donne :

$$G_{k+1}(i) = \sum_{m=-2}^{m=2} w(m) G_k(i + 2^k m)$$

Les filtres passe-bande sont alors calculés :

$$L_k = G_k - G_{k+1}, \text{ avec } k \in [0, bf]$$

Ensuite les poids  $g_k$  sont ajustés pour chaque bande et  $L_k$  est multiplié par les valeurs courantes de ces poids. Puis les bandes de différents mouvements sont mélangés. Finalement le signal du nouveau mouvement est construit par :

$$G'_0 = G_{bf} + \sum_{k=0}^{bf-1} (g_k L_k)$$

### c) Discussion

Dans le cas de mouvements capturés, les techniques de modification de mouvement utilisent des mouvements réalistes intrinsèquement, puisque capturés, pour animer des personnages ressemblant à des humains. Cependant, le réalisme des trajectoires capturées peut être perdu à la suite d'importantes modifications. En fait, les modifications appliquées aux trajectoires ne prennent pas en compte les effets dynamiques et n'assurent pas la préservation de la dynamique intrinsèque au mouvement humain. Tout comme dans les approches cinématiques, les trajectoires sont découplées de sorte que le fort couplage naturel dans un mouvement articulé est perdu dans beaucoup de cas. Même si toutes les fréquences sont changées de la même manière, il faut quand même faire attention au décalage qui apparaît naturellement entre les articulations dans le cycle de locomotion réel.

Bien que la qualité des animations produites dépende du modèle, la qualité des séquences résultantes dépend elle-même de l'habileté des utilisateurs. Pour les techniques de déformations du mouvement, aucun contrôle n'assure que la séquence résultante est celle désirée. En outre, ces techniques ne préservent pas le contact avec le sol. En conséquence, l'utilisateur d'un tel système doit modifier les paramètres des trajectoires ainsi que les poids de l'interpolation et ce par tâtonnements, afin de produire le mouvement désiré.

### 3.4.3 Fondu de mouvements

Le fondu de mouvement nécessite une base de données de mouvements caractéristiques qui sont décrits soit dans le domaine fréquentiel, soit dans le domaine temporel. La méthode consiste à interpoler les paramètres de plusieurs mouvements pour produire de nouveaux mouvements.

#### a) Domaine temporel

Une première méthode d'interpolation dans l'espace de positions paramétrées des mouvements est présentée dans [GR96]. Chaque mouvement de structure à  $m$  articulations est exprimé dans un espace  $P^m = (a_1, a_2, \dots, a_m)$  de dimension  $m$ . Ainsi, pour  $n$  positions clés  $K_1^m, K_2^m, \dots, K_n^m$ , il est possible de calculer les trajectoires angulaires par une fonction d'interpolation  $F(s)$  de dimension  $m$  avec  $s$  l'abscisse curviligne. Les trajectoires angulaires résultantes sont appliquées sur un segment de ligne horizontal par une conversion de paramètres, afin d'obtenir un espace de positions 1D. Une fois ces opérations effectuées sur les  $k$  mouvements référencés, de nouveaux mouvements sont créés par combinaison linéaire de fonctions d'interpolation de référence  $F_i(x)$ , ( $i = 1, 2, \dots, k$ ) où  $x$  est la coordonnée de l'espace de positions 1D. Pour le mouvement humain, quatre locomotions différentes sont considérées : la marche avec petites enjambées, la marche avec grandes enjambées, la course avec petites enjambées, la course avec grandes enjambées (cf. Figure 3.19).

Pour interpoler des mouvements capturés, de nombreux mouvements sont nécessaires pour définir un espace de paramètres, par exemple, la position d'un effecteur terminal ou les types de marche. De nouveaux mouvements sont créés par interpolation de paramètres



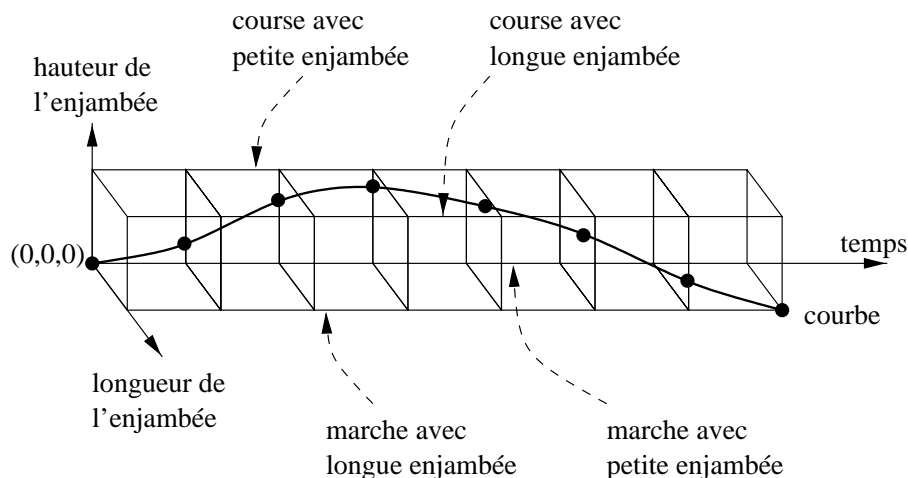


FIG. 3.19: Courbe de positions définie dans l'espace de positions, pour un cycle de locomotion humaine.

de certains mouvements capturés par des splines cubiques. Le problème de cette technique est la taille de la base de données qui est fortement dépendante de la précision requise pour le mouvement résultant. De plus, pour le mouvement humain, différentes postures correspondent en général à un seul paramètre, par exemple la position de la main. Ces postures peuvent être distinguées dans ce type de modèle si de nouveaux paramètres sont ajoutés. Par conséquent, le volume de la base de données de mouvements pré-enregistrés augmente proportionnellement à ce nombre de paramètres. Pour diminuer le volume de cette base de données, [KC96, Ko94] définissent un système appelé VRLOCO qui mélange automatiquement des mouvements calculés avec des modèles cinématiques de marche humaine. Dépendant de la direction et de la vitesse, le système génère automatiquement des empreintes et sélectionne le type de démarche approprié.

## b) Domaine fréquentiel

Une autre méthode consiste à utiliser des expansions de Fourier de mouvements humains expérimentaux pour interpoler ou extrapoler la locomotion humaine [UAT95]. Les paramètres du mouvement sont traités comme des signaux échantillonnés. Chaque signal contient les valeurs d'un degré de liberté particulier pour chaque position. Les trajectoires angulaires sont, dans un premier temps, exprimées par des séries de Fourier dans le domaine fréquentiel :

$$q_n = \alpha_0 + \sum_{k \geq 1} \alpha_k \sin\left(2\pi k \frac{n}{N} + \phi_k\right) \quad (3.14)$$

où  $\alpha_0$  est la valeur centrale,  $\alpha_k$  l'amplitude de la  $k^{\text{ième}}$  composante du signal,  $\phi_k$  la phase,  $N$  le nombre total de pas de temps,  $n$  un pas de temps.

Un filtre passe-bas est ensuite appliqué pour ne garder que les premiers paramètres ( $\tilde{q}_p$  avec  $p \in [0 \dots 3]$ ). Une fois que les paramètres sont obtenus pour différents styles de

locomotion il est possible d'interpoler un ensemble de paramètres  $(\alpha_k^1, \phi_k^1)$  avec un autre  $(\alpha_k^2, \phi_k^2)$  :

$$\tilde{\alpha}_k = s\alpha_k^2 + (1 - s)\alpha_k^1 \quad (3.15)$$

où  $s$  est une valeur réelle appartenant à l'intervalle  $[0, 1]$ . De nouveaux mouvements peuvent être obtenus en interpolant deux mouvements pré-calculés. Les mouvements résultants sont complètement définis par la valeur de  $s$  et par les deux ensembles de paramètres des mouvements originaux. Par exemple pour rendre une marche de personnage synthétique plus ou moins fatiguée il suffit d'interpoler une marche normale avec une marche fatiguée.

### c) Discussion

Les principaux avantages des techniques de fondu de mouvements sont le faible coût de calcul et l'utilisation de trajectoires de mouvements calculés comme données primaires afin d'assurer le réalisme du mouvement. L'inconvénient majeur est que le nombre d'effets possibles tels que la fatigue, la nervosité, l'adaptation de nouveaux environnements reposent sur le nombre de mouvements pré-enregistrés. Créer une marche de personnage humain qui soit plus ou moins fatiguée n'est possible que si une marche normale et une marche fatiguée ont été au préalable enregistrées. De plus pour ces techniques les auteurs supposent généralement que les transitions entre deux ensembles de paramètres sont linéaires ou au mieux cubiques et donc continues dans le temps. Cependant dans les domaines fréquentiel et temporel rien n'assure que l'ensemble de paramètres interpolés ou extrapolés produise des mouvements réalistes. Enfin dans certains cas l'interpolation entre des ensembles de paramètres peut donner des résultats inexacts. Le personnage peut se retrouver dans une configuration non réalisable par un humain par manque de synchronisation de mouvements par exemple (cf. Figure 3.20).

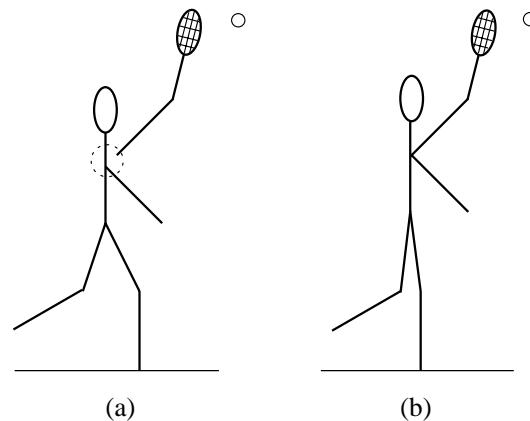


FIG. 3.20: *Fondu d'un mouvement de marche et d'un mouvement de smash : (a) Problème de synchronisation qui a pour conséquence le détachement du bras. (b) Bonne synchronisation, donnant un résultat correct physiquement.*

### 3.5 Conclusion

Les approches cinématiques et basées sur l'interpolation tentent de reproduire certains comportements spécifiques en offrant un outil de description du mouvement. Ce type d'approche nécessite un faible coût de calcul. Les degrés de liberté sont calculés séparément comme s'ils étaient découplés ce qui implique un coût de calcul en  $\mathcal{O}(n)$  avec  $n$  le nombre de degrés de liberté. Lorsque les contraintes de contact entre pied et sol sont prises en compte un coût additif en  $\mathcal{O}(n^3)$  provient de l'utilisation de l'algorithme de cinématique inverse. Néanmoins la qualité du mouvement dépend majoritairement de la qualité et de la quantité de connaissances nécessaires à la reproduction des effets désirés et de l'habileté de l'animateur. Pour la locomotion humaine la façon dont évoluent les trajectoires angulaires doit être décrite a priori. Et pour créer un nouveau modèle le concepteur doit généralement partir de rien pour réaliser un nouvel effet.

Plusieurs approches ont tenté d'améliorer ces modèles sous l'hypothèse que les lois dynamiques sont responsables de la majorité de ces effets. Les méthodes pour produire des mouvements réalistes au sens dynamique ont été développées. Certaines modifient seulement les résultats obtenus avec des modèles cinématiques tandis que d'autres vont plus loin en intégrant directement la dynamique dans le calcul des trajectoires angulaires. Néanmoins pour des modèles aussi complexes que le corps humain le coût en calcul est très important au mieux en  $\mathcal{O}(n^2)$ . De plus les mouvements dynamiques ne semblent pas toujours réalistes visuellement.

Ainsi les méthodes classiques d'animation d'objets que ce soit par cinématique ou par dynamique apparaissent insuffisantes pour animer un personnage articulé de par sa nature complexe. En effet la difficulté majeure provient du manque de contrôle sur le mouvement calculé par de telles méthodes. De nouvelles approches se sont donc intéressées à ce contrôle. Pour contrôler le mouvement d'un personnage articulé il faut savoir comment ses muscles agissent pour le faire avancer dans la direction choisie.

La première approche est l'application des lois de la physique comme contraintes sur le mouvement de parties différentes du personnage en même temps que l'optimisation d'une fonction dans un but donné. L'idée pour avoir une trajectoire possible du personnage est de raffiner itérativement jusqu'à l'obtention d'une trajectoire qui à la fois satisfait les contraintes physiques et optimise la fonction de but. Cependant ce type d'animation comporte encore de nombreux problèmes tels que le non respect des contraintes dynamiques puisqu'un processus d'optimisation est appliqué et le manque d'interactivité puisque la séquence doit être entièrement planifiée à l'avance ce qui limite aussi la complexité de la séquence d'animation.

La seconde approche est la synthèse d'un contrôleur lequel prend des décisions de contrôle basées sur des informations sensorielles mais ne calcule pas explicitement une trajectoire. Il utilise un retour d'informations pour accomplir une tâche. Toutefois comme pour les méthodes dynamiques classiques cette approche nécessite un coût de calcul élevé ce qui rend l'interactivité difficile. De plus le résultat obtenu n'est toujours pas réaliste.

En définitive la dynamique accroît le problème d'animation sans pour autant améliorer de manière significative le résultat.

Les méthodes précédentes font actuellement place aux méthodes de capture de mouvements. Ces approches supposent que des mouvements réalistes peuvent être obtenus par édition et réglage d'une bibliothèque de données de mouvements. Ces approches sont basées sur des mouvements capturés mais peuvent également s'appliquer à des mouvements artificiels calculés avec les approches précédentes. Le coût en calcul est en  $\mathcal{O}(n)$  car chaque articulation est calculée séparément. Bien que des animations convaincantes aient été calculées avec ces méthodes le contrôle de paramètres n'est pas évident et un certain nombre de tests et d'itérations sont souvent nécessaires avant d'obtenir un résultat désiré. De plus les modifications appliquées aux trajectoires pré-définies doivent être faibles de façon à maintenir le réalisme de la séquence résultante. Les problèmes tels l'évitement de pénétration des pieds dans le sol et l'assurance de contact non glissant pour la jambe de support ne sont généralement pas pris en compte dans ces approches.

Ainsi à court terme la capture de mouvement est une solution de choix pour l'animation de personnages. Cependant comme les modèles de mouvements nécessitent d'être prédits ils ne peuvent produire que des variations limitées des données capturées. Par conséquent à long terme l'animation dynamique devrait continuer à se développer pour produire de nouveaux mouvements.

Finalement les méthodes répondant le mieux à nos attentes sont les méthodes basées sur la capture de mouvements. En effet ce sont elles qui permettent d'avoir un mouvement de marche humainement réaliste puisque ce mouvement est directement enregistré sur un humain. Nous proposons dans le chapitre suivant une méthode pour obtenir les mouvements de référence pour le robot bipède. Nous calculons un cycle de marche à partir de mouvements de marche capturés sur un être humain. Nous complétons ce cycle par des mouvements transitoires de départ et d'arrêt que nous calculons par interpolations de polynômes cubiques.



## Chapitre 4

# Mouvements de référence pour le robot

### 4.1 Introduction

Le but de ce chapitre consiste à produire non pas des mouvements optimaux mais des mouvements proches de la marche humaine pour le robot. Nous avons besoin de plusieurs types de trajectoires de référence de sorte que le robot puisse avoir la possibilité de choisir et d'adapter son mouvement à l'environnement. La capture de mouvements fournit des trajectoires de référence cycliques. Les progrès récents des techniques de capture de mouvements permettent de générer facilement et rapidement des données de mouvements humains. Ces techniques de capture de mouvements ont été largement utilisées pour animer des squelettes de corps rigides 3D. En utilisant des technologies magnétiques ou optiques il est possible de stocker les positions et les orientations de marqueurs localisés sur le corps humain. L'ensemble de toutes les positions représente le mouvement réalisé par l'humain le mouvement qui est ensuite rejoué par le personnage virtuel.

Nous présentons la méthode que nous avons utilisée pour obtenir un cycle de marche à partir des données qui nous ont été fournies. N'ayant aucune donnée sur des mouvements transitoires du robot nous présentons ensuite une méthode qui permet de calculer de tels mouvements. Ces mouvements transitoires nous paraissent nécessaires car il s'avère insuffisant de n'avoir qu'un cycle de marche : avant d'avoir un rythme de marche cyclique il faut que le robot puisse démarrer puis s'arrêter après.

### 4.2 Adaptation de mouvements capturés pour l'obtention d'un cycle de marche

Les données qui nous ont été fournies sont des groupes de positions cartésiennes des marqueurs comme celles fournies dans [Win90]. Chaque groupe comprend les positions cartésiennes de tous les marqueurs placés sur le sujet dans le repère du monde et correspond à une posture du sujet à un instant donné. L'ensemble de ces groupes correspond à l'ensemble des postures du sujet c'est-à-dire à son mouvement.

Nous avons vu au chapitre 2 que le contrôle du mouvement devrait être défini par fonction de tâches sur des trajectoires articulaires ou angulaires. En effet il est plus courant de travailler avec ce type de données en robotique en particulier lorsqu'un environnement est pris en compte. Ainsi la trajectoire définie est indépendante du référentiel de base et peut être rejouée par le robot n'importe où dans l'environnement du moment que les éléments de cet environnement le permettent. D'ailleurs ce type de représentation est d'autant plus nécessaire dès que nous souhaitons considérer des mouvements cycliques. Cela n'a effectivement que peu de sens de parler de trajectoires cycliques si ces trajectoires sont définies en coordonnées cartésiennes. Le premier travail a donc consisté à transformer ces données cartésiennes en coordonnées articulaires.

#### 4.2.1 Transformation en coordonnées articulaires

Le mouvement de marche comme nous l'avons noté au chapitre refch :marche se déroule principalement dans le plan sagittal. Il constitue donc un mouvement 2D ce qui est vérifié sur les données capturées les valeurs des coordonnées en  $Z$  restant quasiment constantes au cours du mouvement pour chaque articulation. Par conséquent les segments sont coplanaires avec pour normale l'axe  $(OZ)$ . Nous déterminons les coordonnées articulaires comme étant les angles entre les segments adjacents aux articulations. Plus précisément un repère local est attaché à chaque segment. Le repère qui lui est adjacent est soit son fils soit son père dans la structure arborescente. Un segment fils est entièrement défini par rapport à son père. Ainsi l'origine de son repère local correspond à l'extrémité du repère de son père et il est orienté d'un angle  $\theta$  par rapport à son père. C'est cet angle que nous souhaitons déterminer.

Pour cela nous utilisons la méthode des tangentes. Nous calculons la valeur de l'angle  $\theta_i$  que fait un segment  $i$  avec l'horizontale par :

$$\theta_i = \arctan \frac{y_B - y_A}{x_B - x_A}$$

où les points  $A = (x_A, y_A, z_A)$  et  $B = (x_B, y_B, z_B)$  sont les extrémités du segment  $i$  (cf. Figure 4.1 (a)).

Puis pour une articulation donnée nous calculons les angles  $\theta_i$  et  $\theta_j$  pour les deux segments  $i$  et  $j$  qui lui sont adjacents et nous en déduisons la valeur de l'angle  $\theta$  associé à cette articulation par :

$$\theta = \theta_j - \theta_i$$

si le segment  $j$  est défini par rapport au segment  $i$  (cf. Figure 4.1 (b)). Sinon c'est l'opposé.

#### 4.2.2 Détermination du cycle

Une fois que nous avons le mouvement en coordonnées articulaires nous déterminons le cycle de référence de ce mouvement. Cependant le mouvement capturé s'est avéré

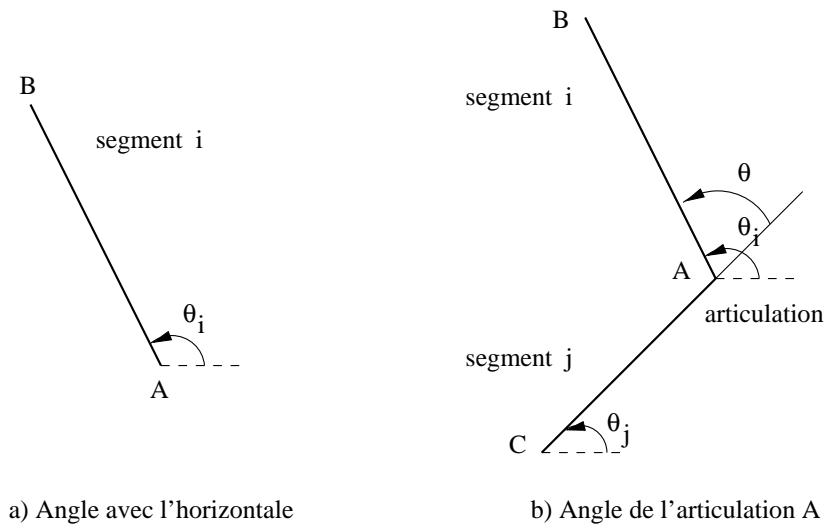


FIG. 4.1: Calcul des angles par la méthode des tangentes.

n'être qu'un cycle et demi de marche. Nous n'avons pu le découper en plusieurs cycles de manière à vérifier la stabilité du mouvement. Nous nous sommes contentés de découper le mouvement en un cycle à partir de deux événements identiques à savoir le décolllement des orteils. Nous avons ensuite vérifié que les coordonnées articulaires de toutes les articulations étaient identiques à ces deux instants. L'erreur entre ces deux postures calculée à partir des angles en radians par :

$$e = \sum_i (q_i(t_1) - q_i(t_2))^2$$

avec  $t_1$  et  $t_2$  les instants correspondants des deux postures est de l'ordre de 0.005 ce que nous avons jugé comme tout à fait acceptable.

### 4.2.3 Longueur des segments

Nous avons ensuite déterminé la longueur des segments-membres du sujet humain c'est-à-dire entre chaque couple de marqueurs adjacents. Les données capturées de positions étaient bruitées certainement à cause du bruit lié à la capture du mouvement et du déplacement des capteurs sur la peau du sujet pendant le mouvement. Nous avons donc calculé la moyenne de ces longueurs sur l'ensemble du mouvement. Nous avons utilisé ces longueurs pour notre robot virtuel. Elles sont légèrement différentes des longueurs prévues pour le robot réel. Le fait de les utiliser évite d'éventuels problèmes de cinématique. En effet si nous prenons pour notre simulation des longueurs de segments différentes de celles de l'humain lors de la capture du mouvement et si nous rejouons les mêmes valeurs des angles au cours du temps il est possible que les pieds de notre robot virtuel pénètrent le sol. Il devient alors nécessaire de faire appels aux techniques de réajustement de mouvement à de nouveaux personnages (cf. paragraphe 3.3.4).



#### 4.2.4 Mouvement de la racine de la structure arborescente

Il n'est cependant pas suffisant de déterminer le mouvement articulaire d'une structure pour en obtenir son mouvement à proprement parler. En effet, pour l'instant, le mouvement fait l'effet d'une marche sur place. La raison provient du fait que la racine de la structure arborescente est fixe. Pour rétablir le mouvement original, il nous faut fournir à cette racine un mouvement de base. La racine de notre robot est l'orteil du pied droit. Par conséquent, nous récupérerons dans les données de mouvement initiales les positions cartésiennes du marqueur associé. Puis, nous les translatons pour faire débiter le mouvement à l'origine du repère du monde. Nous déterminons aussi la longueur d'une foulée. Ainsi, la racine pour chaque nouveau cycle sera translatée de cette longueur, par rapport à sa valeur au cycle précédent.

#### 4.2.5 Discussion

Nous avons présenté une méthode pour calculer un cycle de référence dans le cas de la marche. Nous avons ainsi un cycle complet de marche en coordonnées articulaires pour le sujet humain. Ce cycle constitue le mouvement cyclique de référence pour notre robot virtuel (cf. Figure 4.2). Les résultats obtenus semblent visuellement corrects. Lorsque la jambe gauche devient la jambe de support, nous avons également vérifié que le pied était en contact avec le sol sans pénétration. Si cela n'avait pas été le cas, nous aurions pu utiliser la cinématique inverse pour rétablir la contrainte de contact sans pénétration du pied dans le sol en calculant les angles des articulations de la jambe à partir de la position cartésienne du pied au sol.

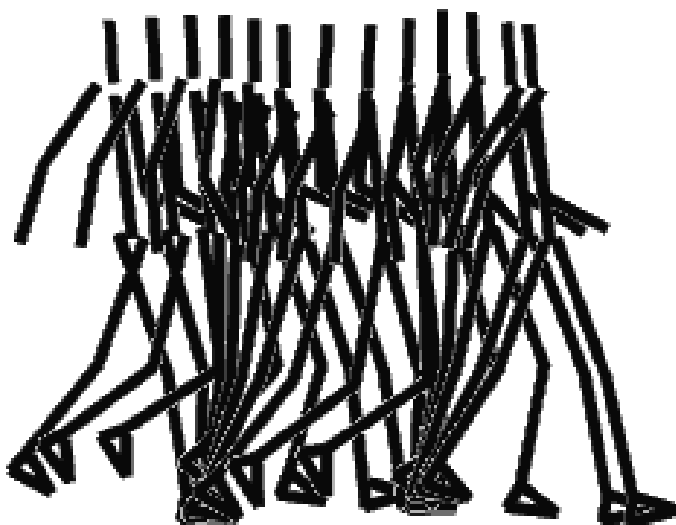


FIG. 4.2: *Cycle de marche pour le robot bipède.*

Nous avons ainsi obtenu des cycles de référence pour la marche sur sol plat. La même technique peut être étendue à d'autres types de marchesΓtels que la montée et la descente d'escaliers. ToutefoisΓil est à noter queΓbien que physiquement réalistes du point de vue du contact au sol et de la non-pénétration dans le solΓces mouvements ne sont pas obligatoirement physiquement corrects. En effetΓles signaux obtenus lors de l'enregistrement du mouvement ne sont pas exempts de bruit lié à l'électronique du matériel ou au déplacement des capteurs sur la peau. AinsiΓdes perturbations apparaissent dans les signauxΓsous forme de hautes fréquences. CependantΓil ne faut pas aveuglément éliminer toutes les hautes fréquencesΓcertaines pouvant relever d'une caractéristique de la marche et non pas de bruit. Une méthode consiste à capturer plusieurs cycles de mouvementΓet à réaliser des analyses statistiquesΓdes moyennesΓdans le but d'obtenir des données les plus exactes possibleΓpour les mouvements à faire effectuer par le robot. Une amélioration concernant les cycles de référence pour le robotΓconsiste donc à utiliser cette méthode sur d'autres cycles de mouvement capturé.

Nous avons vu que le robot a besoinΓnon seulement de cycles de référence pour évoluer dans un environnementΓmais également de mouvements transitoiresΓen particulier pour le démarrage et l'arrêt du robot. Ces mouvements pourraient être capturésΓde la même manière que les mouvements cycliques. Pour les mouvements cycliquesΓdes cycles quasi identiques peuvent être reproduits aisément de par la périodicité de la démarcheΓpermettant ainsi la validation des mouvements obtenusΓgrâce à la méthode d'analyse statistique sur les cycles. Il en va tout autrement pour les mouvements transitoires. En effetΓil est difficile de capturer des mouvements transitoires suffisamment similairesΓmême à partir de conditions initiales identiquesΓen raison de la non-périodicité de tels mouvements. Par conséquentΓnous ne pouvons appliquer la méthode des moyennes sur plusieurs de ces mouvementsΓceux-ci pouvant être trop différents. AinsiΓces transitions sont généralement calculéesΓpar des méthodes cinématiques [Mul98] ou des méthodes de déformations de mouvements [RGBC96]. Nous présentons dans le paragraphe suivantΓune méthode cinématique qui permet de calculer ces mouvements transitoiresΓen essayant de garantir le réalisme physique.

### 4.3 Synthèse de trajectoires transitoires

Les mouvements transitoires consistent à connecter ensemble une trajectoire de référence et une posture particulière du robotΓou deux trajectoires de référence. La trajectoire transitoire de départ est la trajectoire depuis la posture debout standard (cf. Figure 4.3 (a)) vers une position dans le cycle de marche (cf. Figure 4.3 (b)). La trajectoire transitoire d'arrêt est la trajectoire depuis une position dans le cycle de marche vers la posture debout standard.

Des méthodes ont été proposées en synthèse d'imagesΓpour générer des transitions entre deux mouvementsΓen particulierΓdans le cas de mouvements capturés (cf. para-



a) La posture debout standard. b) Une posture lors de la marche.

FIG. 4.3: *Postures du robot bipède.*

graphe 3.4.3). Cependant ce type de transitions est généralement calculé par interpolation de paramètres du mouvement. Ces paramètres interpolés n'ont pas de réalité physique il n'est par conséquent pas intuitif de comprendre les causes et les effets et donc de déterminer leur valeur pour obtenir des mouvements désirés.

Une approche similaire en robotique concerne la simulation de transition pour un robot marcheur anthropomorphe [BOW98]. Les auteurs définissent plusieurs modes de locomotion tous obtenus à partir de transformations d'un seul mouvement original paramétré. Pour simuler la transition entre deux de ces modes certains paramètres du mouvement sont simplement interpolés linéairement. Aucune contrainte physique n'est vérifiée et en l'occurrence le sol ne semble pas être considéré.

Un autre type de méthodes traitant des transitions utilise une optimisation sous contraintes pour réaliser un mouvement transitoire [RGBC96]. Toutefois cette technique nécessite que l'animation soit entièrement planifiée pour qu'elle soit pré-calculée. Le mouvement calculé est ensuite joué pendant la phase proprement dite d'animation. Cette méthode ne nous convient pas car nous voulons que le robot qui ne connaît pas a priori son environnement soit capable d'adapter sa trajectoire en temps réel selon l'environnement courant et les obstacles rencontrés.

Nous présentons cette méthode pour générer les trajectoires transitoires de départ et d'arrêt pour une marche normale. Nous verrons comment adapter cette méthode à d'autres cas tels que la montée et la descente d'escaliers.

### 4.3.1 Méthode

Nous souhaitons obtenir une trajectoire de référence depuis la posture debout standard  $q_0$  vers cette même posture (cf. Figure 4.3 (a)). Cette trajectoire consiste en une trajectoire transitoire de départ suivie d'un ou plusieurs cycles de marche et s'achevant par une trajectoire transitoire d'arrêt. Le cycle de marche est celui calculé dans le paragraphe précédent. Nous désirons alors calculer une trajectoire transitoire  $q(t)$  entre deux positions extrêmes qui doivent être spécifiées. Nous notons la position initiale  $q_i$  et la position finale  $q_f$ . Ces positions extrêmes dépendront du type de trajectoires transitoires à calculer.

Par exemple s'il s'agit d'une trajectoire transitoire entre deux trajectoires de référence données la position initiale  $q_i$  sera une position à déterminer de la première trajectoire de référence et la position finale  $q_f$  sera une position à déterminer de la seconde trajectoire de référence. Un autre exemple est lorsque nous désirons commencer un mouvement nous calculons la trajectoire transitoire entre la posture debout  $q_0$  et une position particulière de la trajectoire de référence qu'il reste à choisir.

Pour calculer la trajectoire transitoire nous décrivons la trajectoire en termes de polynômes cubiques :

$$q(t) = A_0 + A_1t + A_2t^2 + A_3t^3 \quad (4.1)$$

où  $q(t)$  et  $A_i$  sont des vecteurs dans  $\mathbb{R}^{14}$ . Les coefficients  $A_i$  déterminent quatre contraintes pour définir le mouvement de chaque articulation. Nous obtenons ces contraintes à partir des conditions initiales et finales du mouvement en dérivant l'équation (4.1) deux fois par rapport au temps. Ceci assure la continuité des dérivées premières et secondes du mouvement c'est-à-dire la vitesse et l'accélération. Les coefficients  $A_i$  sont donc calculés pour les valeurs prises aux extrémités de la trajectoire transitoire lorsque  $t = 0$  et  $t = T$  avec  $T$  la durée totale de la trajectoire transitoire :

$$\begin{cases} q(0) = q_i \\ \dot{q}(0) = \dot{q}_i \\ q(T) = q_f \\ \dot{q}(T) = \dot{q}_f \end{cases} \Rightarrow \begin{cases} A_0 = q_i \\ A_1 = \dot{q}_i \\ A_2 = \frac{1}{T} \left( \frac{3(q_f - q_i)}{T} - \dot{q}_f - 2\dot{q}_i \right) \\ A_3 = \frac{1}{T^2} \left( -\frac{2(q_f - q_i)}{T} + \dot{q}_f + \dot{q}_i \right) \end{cases}$$

Puisque les vitesses et les accélérations articulaires sont continues l'équation (4.1) donne un mouvement lisse ce qui est nécessaire pour que le mouvement soit réaliste. Dans le prochain paragraphe nous présentons une méthode pour déterminer les positions initiale  $q_i$  et finale  $q_f$  suivie d'une méthode pour calculer la durée  $T$  dans le paragraphe suivant. Ceci nous donnera des contraintes pour obtenir un mouvement réaliste.

### 4.3.2 Détermination des positions extrêmes

Notre trajectoire de référence cyclique  $q^r$  contient  $N$  poses ou instantanés (*frames*) elle est décrite par  $q^r(n)$   $n = 1..N$ . Tout d'abord nous déterminons la position  $q^r(s)$  dans la trajectoire cyclique correspondant au moment du milieu de support. Ainsi toutes les

poses précédentes ( $n < s$ ) correspondent à une posture où la jambe de balancement est derrière la jambe de support et toutes les poses suivantes ( $n > s$ ) correspondent à une posture où la jambe de balancement est devant la jambe de support.

Dans le cas des trajectoires transitoires de départ et d'arrêt une des positions extrêmes  $q_i$  ou  $q_f$  est la posture debout standard pour laquelle le vecteur d'état est défini par  $X_0 = (q_0, \dot{q}_0)$  où  $\dot{q}_0$  est un vecteur de vitesse nulle. En effet lorsque le robot est en posture debout standard il est immobile. L'autre position extrême est une des positions de la trajectoire cyclique considérée qu'il faut déterminer.

Pour choisir cette seconde position extrême notée  $q^r(n^*)$  nous souhaitons optimiser un critère énergétique. Nous utilisons donc une méthode de minimisation aux moindres carrés qui calcule toutes les distances angulaires entre les positions  $q^r(n)$  dans la trajectoire cyclique et la position debout standard  $q_0$  :

$$R(n) = (q^r(n) - q_0)^T Q (q^r(n) - q_0) \quad (4.2)$$

où  $Q$  est une matrice symétrique définie positive qui spécifie le poids de chaque articulation. Nous déterminons ensuite la valeur  $n^*$  qui minimise  $R(n)$  ce qui donne la distance angulaire minimale entre  $q_0$  et  $q^r(n^*)$ .

Nous avons testé une autre minimisation prenant en compte non seulement les différences d'angles mais également les différences de vitesses angulaires entre les deux postures. Cependant cette minimisation s'est avérée moins intéressante que la précédente les résultats obtenus semblant visuellement moins satisfaisants peut-être en raison du vecteur de vitesse angulaire nul associé à la posture standard debout.

### 4.3.3 Durée de transition basée sur les accélérations

Dans le cas du robot réel des moteurs placés aux articulations devront délivrer des forces et des couples pour réaliser les mouvements souhaités. L'équation de la dynamique met en relation ces forces et ces couples avec les accélérations articulaires. Une idée est donc de contraindre les accélérations articulaires  $\ddot{q}$  pendant la trajectoire transitoire de manière à avoir une limitation sur les forces et les couples nécessaires à développer par les moteurs. Nous cherchons donc à assurer que ces accélérations articulaires restent dans un domaine que nous astreignons par les accélérations minimales et maximales atteintes pendant le cycle de référence :

$$\ddot{q} \in [\ddot{q}_{min}^r, \ddot{q}_{max}^r]$$

#### *Principe*

Pour les trajectoires transitoires que nous avons définies avec l'équation (4.1) les accélérations sont données en fonction de la durée  $T$  de la trajectoire transitoire selon l'équation suivante :

$$\ddot{q}(t) = 6A_3 t + 2A_2 \quad (4.3)$$

$$= \frac{6t}{T^2} \left( 2 \frac{q_i - q_f}{T} + \dot{q}_i + \dot{q}_f \right) + \frac{2}{T} \left( -3 \frac{q_i - q_f}{T} - 2\dot{q}_i - \dot{q}_f \right) \quad (4.4)$$

Il s'agit de l'équation d'une droite dont la valeur maximale est atteinte soit pour  $t = 0$  soit pour  $t = T$ . Par conséquent l'accélération angulaire maximale est :

$$\ddot{q}_{max} = \max(|\ddot{q}(0)|, |\ddot{q}(T)|)$$

$$\text{avec} \quad \begin{cases} \ddot{q}(0) &= \frac{2}{T} \left( -3 \frac{q_i - q_f}{T} - 2\dot{q}_i - \dot{q}_f \right) \\ \ddot{q}(T) &= \frac{2}{T} \left( 3 \frac{q_i - q_f}{T} + \dot{q}_i + 2\dot{q}_f \right) \end{cases}$$

Il faut donc que toutes les accélérations angulaires  $\ddot{q}$  du mouvement transitoire respectent la contrainte suivante :

$$\ddot{q}_{max} \leq \ddot{q}_M^r$$

avec  $\ddot{q}_M^r = \max(|\ddot{q}_{min}^r|, |\ddot{q}_{max}^r|)$ . Nous calculons donc pour chaque articulation  $i$  le temps maximal  $T_i$  pour que cette articulation  $i$  ne rentre pas en butée à cause de l'accélération lors du mouvement transitoire c'est-à-dire le temps vérifiant l'équation :

$$\ddot{q}_{i,M}^r = \ddot{q}_{i,max} \quad (4.5)$$

$$-U = \frac{2}{T_i} \left( \frac{V}{T_i} + W \right) \quad (4.6)$$

où  $U = \ddot{q}_{i,M}^r$  et  $V$  et  $W$  sont fonctions de  $q_i$  et  $q_f$  (cf. équation (4.4)) :

$$\text{Si } \ddot{q}_{max} = \ddot{q}(0) : \begin{cases} V &= -3(q_i - q_f) \\ W &= -2\dot{q}_i - \dot{q}_f \end{cases}$$

$$\text{Si } \ddot{q}_{max} = \ddot{q}(T_i) : \begin{cases} V &= 3(q_i - q_f) \\ W &= \dot{q}_i + 2\dot{q}_f \end{cases}$$

L'équation (4.6) se réécrit :

$$UT_i^2 - 2WT_i - 2V = 0$$

qui est un polynôme de degré 2 dont les racines dépendent du discriminant  $\Delta$  :

$$\Delta = W^2 + 2VU$$

$$\text{si } \Delta > 0 : \text{ deux solutions : } T_i = \frac{W \pm \sqrt{W^2 + 2VU}}{U}$$

$$\text{si } \Delta = 0 : \text{ une solution : } T_i = \frac{W}{U}$$

$$\text{si } \Delta < 0 : \text{ pas de solution}$$

Pour le cas où  $\Delta > 0$  nous prenons la première racine qui correspond à la plus grande valeur :  $T_i = \frac{W + \sqrt{W^2 + 2VU}}{U}$ . Nous n'avons pas rencontré de cas avec  $\Delta < 0$ . Nous prenons finalement  $T_{max} = \max(T_i)$  comme temps minimal qui assure qu'aucune articulation ne rentre en butée en accélération.

### Discussion

Nous obtenons par ce procédé une durée de trajectoire transitoire qui correspond à un tiers de la durée du cycle de référence. Cette durée ne convient pas car elle nous paraît trop grande pour être effectivement réaliste. En effet les transitions entre mouvements ou postures sont généralement des mouvements courts et donc rapides en particulier pour les cas du départ et de l'arrêt de la marche. Aussi il apparaît normal et naturel de penser que l'accélération doit être plus importante au début et à la fin de la marche. D'ailleurs cela semble assez intuitif que les êtres humains développent une force d'initiation ou de cessation de mouvement plus importante que celle requise pour le maintien du mouvement cyclique pour lequel un mode économique est naturellement adopté.

D'autre part nous avons implicitement fait l'hypothèse que les trajectoires transitoires devaient être des mouvements courts de par leur définition en termes de polynômes cubiques. Or si nous calculons la trajectoire transitoire avec la durée obtenue le mouvement réalisé pendant cette durée est plus long que ce que nous sommes en droit d'attendre et plus complexe aussi en raison de l'utilisation de polynômes cubiques. Effectivement nous obtenons un mouvement où le pied de balancement part initialement en arrière avant d'aller vers l'avant pour entamer le cycle de marche ce qui donne un résultat plutôt artificiel. Ainsi cette méthode de détermination de la durée de la trajectoire transitoire est inappropriée et a été abandonnée au profit d'une méthode basée sur les contraintes géométriques que nous exposons dans le paragraphe suivant.

#### 4.3.4 Durée de transition basée sur des contraintes géométriques

Puisque nous ne connaissons pas de critère objectif pour déterminer la durée totale  $T$  de la trajectoire transitoire nous choisissons donc un critère subjectif à savoir le réalisme visuel. Nous imposons que  $T$  soit dans l'intervalle  $[T_{min}, T_{max}]$  lequel est obtenu par l'approximation suivante : nous calculons le temps entre  $q^r(s)$  et  $q^r(n^*)$  dans la trajectoire cyclique et nous le notons  $T_{min}$ . Puis nous choisissons  $T_{max} = 3 T_{min}$  de manière arbitraire. Nous vérifions a posteriori que ce temps correspond à une borne jamais atteinte car le mouvement obtenu avec cette durée ne respecte pas les contraintes que nous définissons ci-dessous.

Dans notre trajectoire cyclique le temps entre deux poses consécutives  $q^r(i)$  et  $q^r(i+1)$  est constant et nous le notons  $dT$ . Pour rester compatible nous décidons de prendre  $T$  proportionnel à  $dT$  c'est-à-dire  $T = k dT$   $k$  étant un entier. Pour chaque  $t = k_i dT$  dans l'intervalle  $[T_{min}, T_{max}]$  où  $T_{min} = k_{min} dT$  et  $T_{max} = k_{max} dT$  nous calculons les positions dans le monde de l'orteil et du talon du pied libre pendant la trajectoire transitoire. Ensuite nous vérifions que ni le talon ni l'orteil ne pénètrent le sol ou n'effectuent un mouvement arrière. Nous interdisons les mouvements arrières car ils signifient par rapport à notre méthode de calcul de trajectoires transitoires que le temps mis pour atteindre le but est trop grand.

Nous choisissons finalement  $T$  comme étant le temps  $t$  maximal qui remplit toutes ces conditions. Nous aurions pu prendre n'importe quel temps de l'intervalle de temps valides

qui permettent de respecter toutes les contraintes. Toutefois ce temps maximal permet d'avoir les accélérations angulaires les plus faibles de par la définition de la trajectoire transitoire toujours par souci de faire développer par les moteurs aux articulations du vrai robot les forces et les couples les plus faibles possibles.

### 4.3.5 Application au mouvement de marche

Nous appliquons maintenant notre méthode au calcul des trajectoires transitoires pour la marche cyclique. Nous souhaitons calculer les trajectoires transitoires de départ et d'arrêt pour obtenir une trajectoire de référence complète pour la marche. Notre trajectoire cyclique de référence pour la marche contient 48 poses et  $dT$  vaut  $0.02s$ . La position  $q^r(s)$  la plus proche de  $q_0$  est la 15<sup>e</sup> pose ( $s = 15$ ).

#### a) Départ de la marche

La trajectoire transitoire de départ est un mouvement depuis la position debout  $q_0$  et qui rejoint la position  $q^r(n^*)$  de la trajectoire cyclique de référence c'est-à-dire  $q_i = q_0$  et  $q_f = q^r(n^*)$ . Pour la trajectoire transitoire de départ seules les positions  $q^r(n)$  pour lesquelles la jambe de balancement est devant la jambe de support c'est-à-dire telles que  $n > s$  sont prises en compte. En effet nous ne voulons pas considérer que les positions avec jambe libre à l'arrière soient des positions atteignables pour une trajectoire transitoire de départ. Cela signifierait que la jambe libre a un mouvement arrière lors du démarrage de la marche ce qui n'est pas le cas dans la réalité.

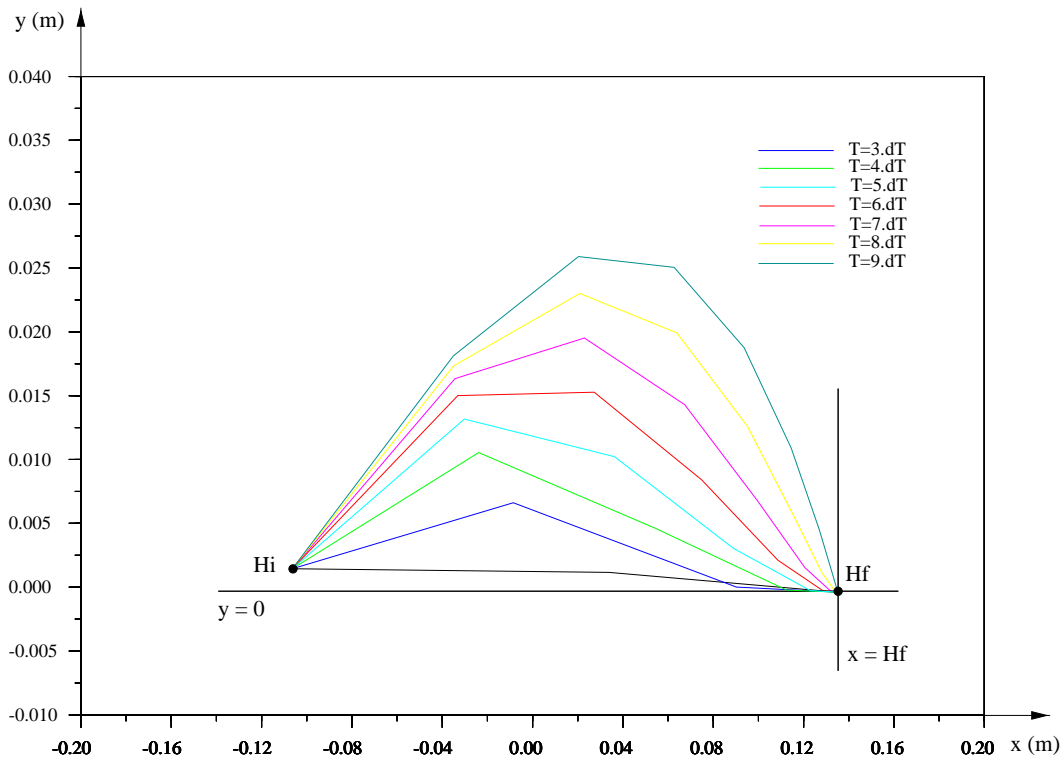
Pour déterminer  $q^r(n^*)$  la jambe de support doit d'abord bouger légèrement tandis que la jambe libre s'élève et commence le cycle de marche. Aussi les coefficients de la matrice de pondération  $Q$  pour la jambe de support doivent être élevés alors que ceux pour la jambe libre doivent être faibles. Nous avons utilisé la matrice diagonale suivante :

$$Q = \text{diag}(15, 5, 5, 10, 5, 3, 5, 5, 5, 0, 0, 0, 0, 0)$$

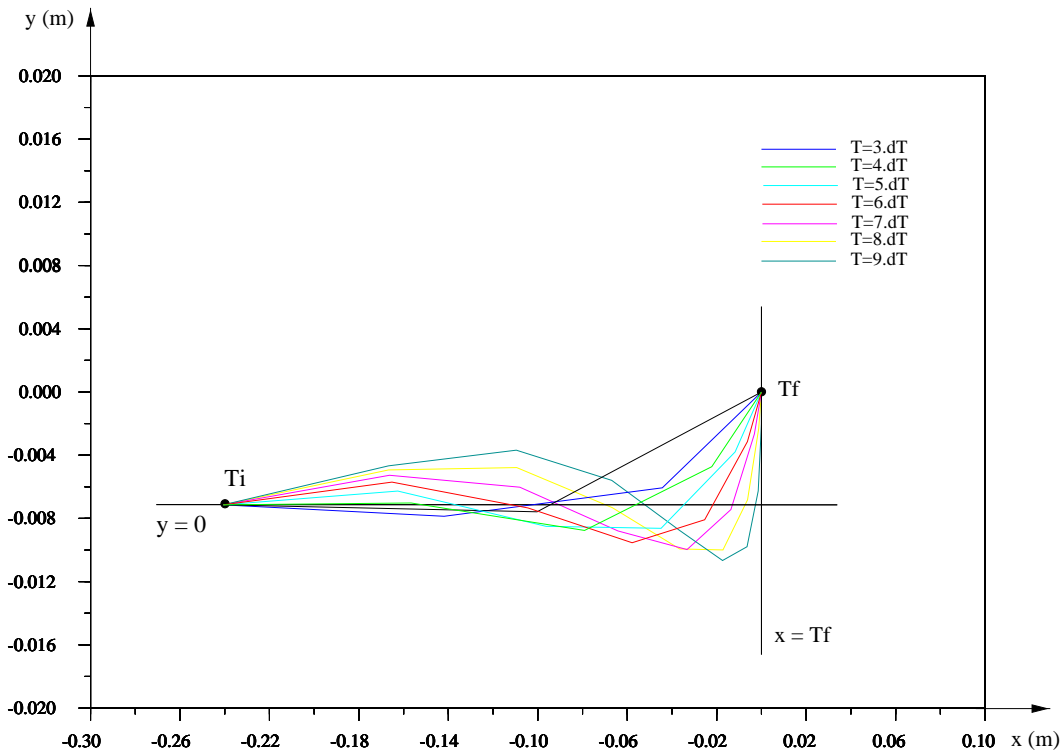
Nous restreignons la recherche de  $q^r(n^*)$  dans l'intervalle  $[q^r(s), q^r(N)]$ . Nous calculons  $R(n)$  donné par l'équation (4.2) pour chaque pose avec  $s \leq n \leq N$  et nous trouvons  $q^r(n^*)$  comme étant la 18<sup>e</sup> pose ( $n^* = 18$ ). Nous calculons ensuite les courbes de mouvement de l'orteil et du talon libres pour  $T$  dans l'intervalle  $[3 dT, 9 dT]$  (cf. Figure 4.4).

La Figure 4.4(a) représente les trajectoires du talon libre calculées pour les différentes durées  $T$  de l'intervalle déterminé.  $H_i$  est sa position dans la pose  $q_i$  et  $H_f$  celle dans  $q_f$ . Toutes les trajectoires qui traversent la droite  $x = H_f$  correspondent à des mouvements arrières du talon et ils sont interdits. Ceci ne donne cependant pas de contrainte ici aucun mouvement calculé ne présentant cette caractéristique. La Figure 4.4(b) donne les trajectoires de l'orteil libre calculées pour les différentes durées  $T$  de l'intervalle déterminé.  $T_i$  est sa position dans la pose  $q_i$  et  $T_f$  celle dans  $q_f$ . Toutes les trajectoires qui passent sous la droite  $y = 0$  correspondent à des pénétrations dans le sol. Les trajectoires calculées pour  $T > 5 dT$  sont donc interdites. Finalement nous obtenons  $T = 5 dT$ . La trajectoire transitoire de départ résultante connectée à la trajectoire cyclique est donnée en Figure 4.5.





a) Talon libre



b) Orteil libre

FIG. 4.4: *Mouvement du pied libre pendant la trajectoire transitoire de départ, pour plusieurs temps totaux  $T$ .*

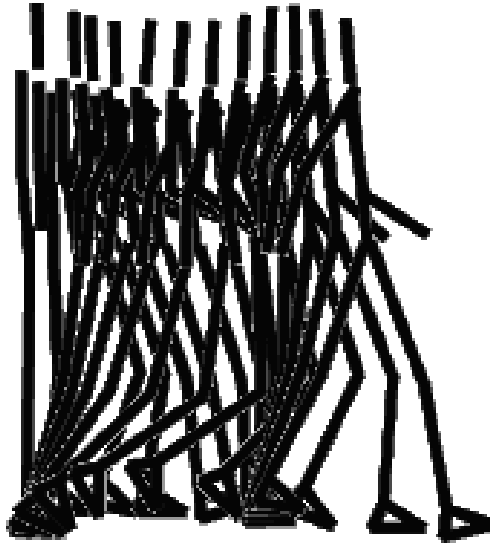


FIG. 4.5: *Mouvement transitoire de départ du robot.*

#### b) Arrêt de la marche

La trajectoire transitoire d'arrêt est un mouvement depuis la position  $q^r(n^*)$  de la trajectoire cyclique de référence  $\Gamma$  qui rejoint la position debout  $q_0$  c'est-à-dire  $q_i = q^r(n^*)$  et  $q_f = q_0$ . Pour la trajectoire transitoire d'arrêt  $\Gamma$  seules les positions  $q^r(n)$  du cycle pour lesquelles la jambe de balancement est derrière la jambe de support c'est-à-dire  $\Gamma$  telles que  $n < s$  sont prises en compte. De manière symétrique au cas du départ  $\Gamma$  nous ne voulons pas que les positions avec jambe libre à l'avant soient des positions atteignables pour une trajectoire transitoire d'arrêt. Cela signifierait que la jambe libre va devant puis revient en arrière  $\Gamma$  pour poser le pied à côté du pied de support.

Pour déterminer  $q^r(n^*)$   $\Gamma$  la jambe de support doit d'abord bouger légèrement  $\Gamma$  tandis que la jambe libre s'abaisse et termine le cycle de marche. Aussi  $\Gamma$  les coefficients de la matrice de pondération  $Q$  pour la jambe de support doivent être élevés  $\Gamma$  alors que ceux pour la jambe libre doivent être faibles. Ce cas est similaire à celui du départ  $\Gamma$  aussi  $\Gamma$  nous reprenons la matrice  $Q$  définie précédemment. Nous restreignons la recherche de  $q^r(n^*)$  dans l'intervalle  $[q^r(1), q^r(s)]$ . Nous calculons  $R(n)$  donné par l'équation (4.2) pour chaque pose  $\Gamma$  avec  $1 \leq n \leq s$  et nous trouvons  $q^r(n^*)$  comme étant la 12<sup>e</sup> pose ( $n^* = 12$ ). Nous calculons ensuite les courbes de mouvement de l'orteil et du talon libres  $\Gamma$  pour  $T$  dans l'intervalle  $[3 dT, 9 dT]$  (cf. Figure 4.6).

La Figure 4.6(a) correspond aux trajectoires du talon libre  $\Gamma$  calculées pour les différentes durées  $T$  de l'intervalle déterminé.  $H_i$  est sa position dans la pose  $q_i$  et  $H_f$  celle dans  $q_f$ . Toutes les trajectoires qui traversent la droite  $x = H_f$  correspondent à des mouvements arrières du talon. Ces mouvements sont interdits  $\Gamma$  et donnent le résultat intermédiaire suivant :  $T \leq 3 dT$ . La Figure 4.6(b) donne les trajectoires de l'orteil libre  $\Gamma$  calculées

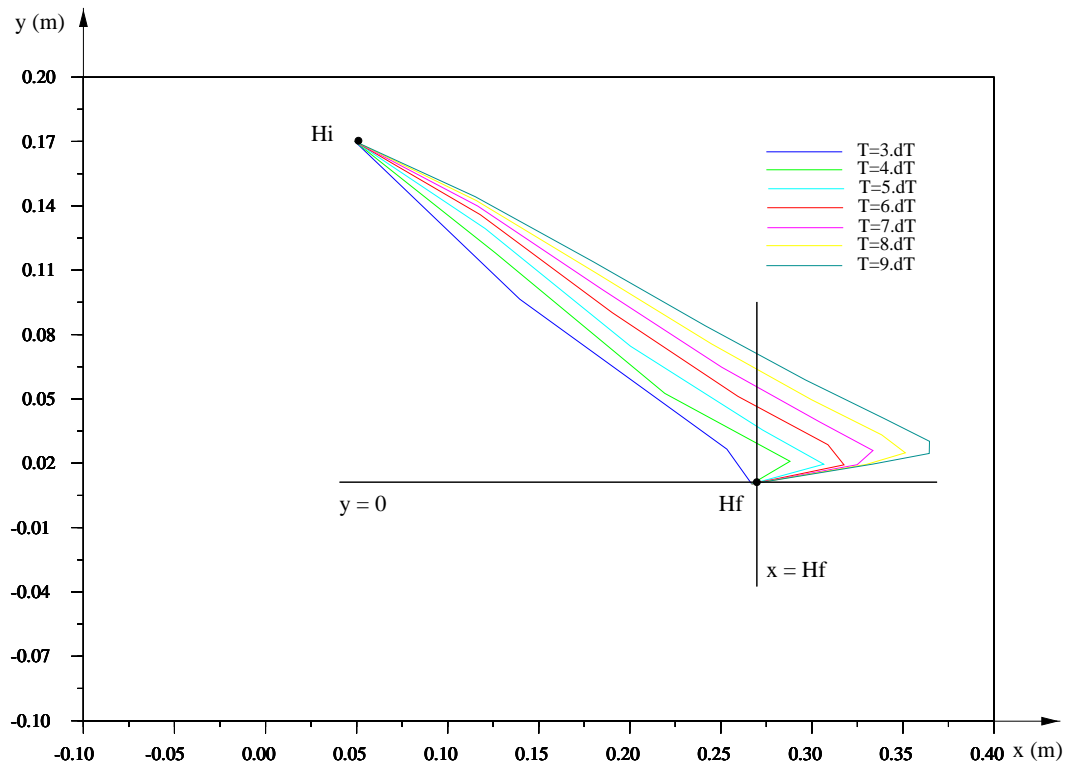
pour les différentes durées  $T$  de l'intervalle déterminé.  $T_i$  est sa position dans la pose  $q_i$  et  $T_f$  celle dans  $q_f$ . Toutes les trajectoires qui traversent la droite  $x = H_f$  correspondent à des mouvements arrière de l'orteil. Les trajectoires calculées pour  $T > 3 dT$  sont donc interdites. Et toutes les trajectoires qui passent sous la droite  $y = 0$  correspondent à des pénétrations dans le sol. La trajectoire calculée pour  $T = 3 dT$  doit être également interdite. Cependant cette pénétration de l'orteil dans le sol n'est que de quelques millimètres. Cette faible erreur peut facilement être corrigée en utilisant des capteurs de contact et le contrôle par fonctions de tâches qui rectifient la trajectoire du pied lors de l'approche de celui-ci au sol [Gén98]. Finalement nous gardons  $T = 3 dT$ . La trajectoire transitoire de départ résultante connectée à la trajectoire cyclique est donnée en Figure 4.7.

### 4.3.6 Discussion

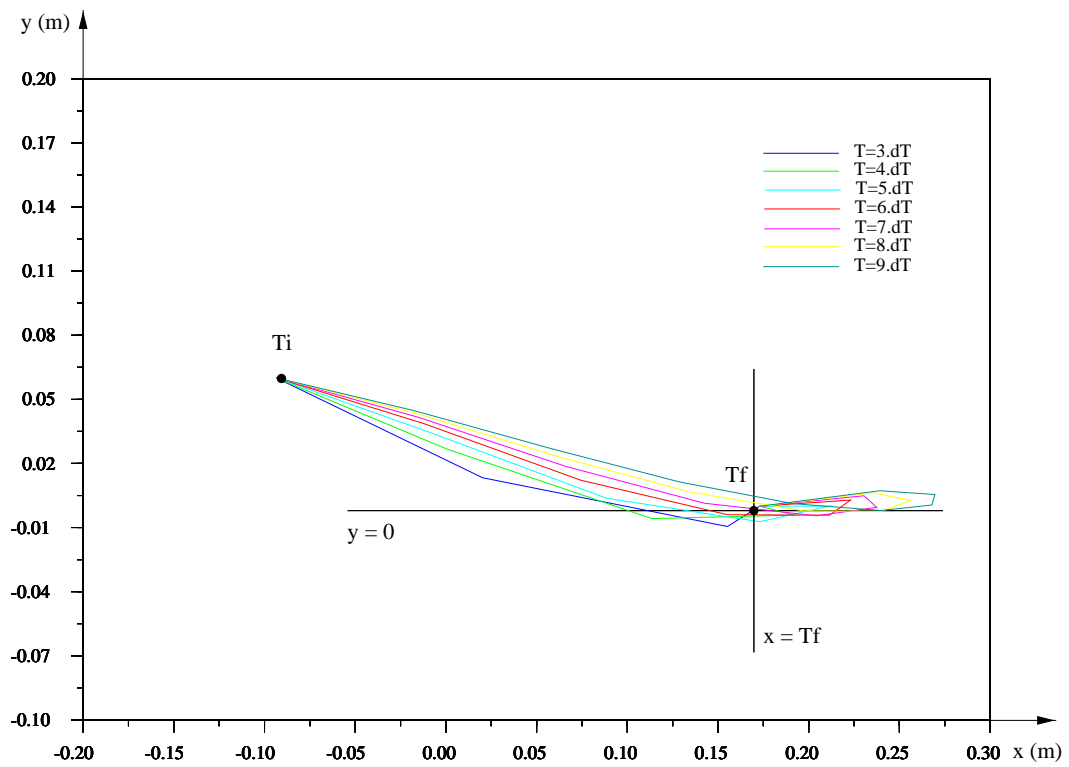
Nous avons proposé une méthode pour générer des trajectoires transitoires que nous avons appliquée aux cas de départ et d'arrêt pour la marche sur sol plat. Nous utilisons des polynômes cubiques pour calculer chaque trajectoire articulaire d'une position donnée à une autre. Une des positions est la posture debout standard et l'autre position appartient à la trajectoire cyclique de référence. Nous déterminons cette position de manière à minimiser un critère énergétique concernant la distance articulaire à effectuer entre la posture debout standard et les positions du cycle de référence. La dernière inconnue qu'il reste à déterminer est la durée  $T$  de la transitoire. Nous avons proposé une première méthode basée sur les accélérations mais le résultat obtenu n'est pas acceptable. Nous avons donc proposé une seconde méthode basée sur les contraintes géométriques de non-pénétrations dans le sol et de non retour arrière (non rebroussement) pour le talon et l'orteil de la jambe libre. Cette seconde méthode peut être améliorée en tenant compte également des butées articulaires à respecter. Finalement cette méthode de génération de trajectoires transitoires est très simple à mettre en œuvre et donne des mouvements réguliers et réalistes. Les résultats sont visuellement de bonne qualité ce critère étant toutefois subjectif.

Cette méthode de génération de trajectoires transitoires est aisément applicable à d'autres démarches telles que la montée et la descente d'escaliers ou de plans inclinés ou tout autre démarche cyclique afin de générer les mouvements transitoires de départ et d'arrêt. Nous pouvons prendre le même critère énergétique pour calculer la position finale ainsi que la même matrice  $Q$ . Et concernant la durée de la transitoire nous pouvons reprendre les contraintes géométriques précédemment utilisées. Toutefois il est possible qu'elles ne soient désormais plus suffisantes et qu'il soit par conséquent nécessaire de définir des contraintes supplémentaires. En particulier pour les marches d'escaliers il faut s'assurer que pendant le mouvement transitoire aucune partie des jambes et des pieds ne pénètre les marches.

Une autre application un peu moins directe est le calcul d'une transition entre deux mouvements. La transition peut être réalisée soit en passant par la position debout standard soit en passant d'une position du premier mouvement à une position du second mouvement. La première façon de faire facilite le calcul et évite les erreurs potentielles concernant les limites articulaires notamment. La seconde façon est un peu plus délicate



a) Talon libre



b) Orteil libre

FIG. 4.6: *Mouvement du pied libre pendant la trajectoire transitoire d'arrêt, pour plusieurs temps totaux  $T$ .*



FIG. 4.7: *Mouvement transitoire d'arrêt du robot.*

car il faut déterminer les deux poses extrêmes dans les deux mouvements. En effet, comme ces deux poses ne sont pas fixes et dépendent l'une de l'autre, il est nécessaire de trouver un critère supplémentaire pour les déterminer de manière unique. Une possibilité est de restreindre la zone de recherche aux zones des deux mouvements à mettre en correspondance. Cette zone peut correspondre à un événement particulier, tel que le contact au sol du talon, utilisé dans [Mul98] pour réaliser les transitions de la marche à la course et de la course à la marche d'un personnage. Ensuite, il reste à minimiser le critère énergétique entre toutes les poses de ces deux zones, en modifiant peut-être la matrice de pondération  $Q$ . Il faut également vérifier les contraintes géométriques, voire en définir de nouvelles correspondant à l'environnement, afin de déterminer la durée de la transition.

## 4.4 Conclusion

Puisque notre robot possède une morphologie anthropomorphe, il est apparu naturel que, pour calculer sa marche, le plus simple était de reproduire la marche humaine acquise par capture de mouvement. Nous avons donc adapté le mouvement capturé d'un cycle de marche, pour obtenir les trajectoires articulaires de référence pour notre robot. Toutefois, les mouvements récupérés étant cycliques, nous avons eu besoin de générer des trajectoires transitoires de départ et d'arrêt, en vue d'obtenir des marches complètes. Nous avons donc proposé une méthode basée sur l'interpolation par polynômes cubiques. Cette méthode permet de calculer des trajectoires transitoires réalistes entre deux poses données, en assurant la continuité des vitesses et des accélérations. Nous avons ainsi obtenu avec succès des trajectoires transitoires de départ et d'arrêt pour la marche sur sol plat.

---

Une fois que le robot est capable de marcher il peut évoluer dans un environnement. Néanmoins il lui manque une aptitude d'une importance capitale pour lui éviter d'agir en aveugle dans l'environnement : la perception. À partir du moment où le robot peut percevoir son environnement il lui faut reconnaître les différents éléments de l'environnement afin d'adapter son mouvement d'ajuster sa trajectoire et en particulier d'éviter les obstacles. La perception et la reconnaissance sont présentées d'un point de vue robotique dans le prochain chapitre dans le cas de robots réels avant de les appliquer dans le chapitre suivant à notre robot virtuel.



## Chapitre 5

# Perception et reconnaissance de l'environnement par un robot réel

### 5.1 Introduction

Nous souhaitons faire évoluer notre robot bipède dans un environnement qui lui est inconnu. Nous nous intéressons par conséquent à rechercher des méthodes de robotique qui permettent d'acquérir des informations sur l'environnement par un robot et de traiter ces informations dans le but d'obtenir une connaissance de l'environnement qui puisse être utile au robot lors de son déplacement. La première tâche qu'il nous importe est donc la perception de l'environnement et la seconde tâche est la reconnaissance des objets de l'environnement.

En robotique la perception de l'environnement par un robot est réalisée par des caméras et des capteurs. Ces dispositifs consistent à interagir avec l'environnement par le biais de divers moyens et fournissent au robot des données relatives à cet environnement. Ces données brutes étant généralement peu expressives elles sont exploitées pour obtenir des informations plus pertinentes directement compréhensibles par le robot. Le contrôle du robot doit alors adapter le comportement de celui-ci en fonction de ces informations et de la tâche que le robot doit exécuter.

Il existe une grande variété de capteurs de différents types et fonctions. Nous nous intéressons en particulier aux capteurs dits proximétriques qui permettent d'acquérir une perception locale de l'environnement. Ces capteurs sont utilisés en robotique pour la détection d'obstacles proches. Cependant ces capteurs ne fournissent au mieux qu'une information de distance qui permet seulement de calculer les coordonnées du point détecté dans l'environnement. Des techniques de reconstruction de formes sont alors utilisées pour reconstruire l'environnement détecté. Puis des techniques de reconnaissance de formes sont appliquées afin d'obtenir une meilleure connaissance des objets constituant l'environnement.

Tout d'abord nous montrons la nécessité d'interaction entre un robot réel et son environnement. Puis nous présentons les capteurs utilisés en robotique pour la perception



de l'environnement en nous attachant plus particulièrement aux capteurs proximétriques. Enfin nous présentons divers moyens pour traiter les données acquises par les capteurs : la fusion multi-capteurs dès lors que plusieurs capteurs sont utilisés ce qui est généralement le cas en robotique mobile ; la reconstruction de formes et la reconnaissance d'objets.

## 5.2 Interactions avec l'environnement en robotique

### 5.2.1 Le robot

Le but de la robotique est de spécifier des tâches à un robot correspondant à un objectif d'intervention sur l'environnement [And83a]. La capacité opérationnelle du robot dépend directement de la maîtrise de l'interaction robot-environnement et la complexité de ce processus peut être très variable. La connaissance des caractéristiques géométriques et physiques de l'environnement constitue donc une contrainte majeure qui est prise en charge par le robot car c'est à lui qu'il appartient de s'adapter à l'environnement.

Contrairement aux premières machines programmables dont le comportement était régi par un simple automate qui générait des trajectoires "en aveugle" dans un environnement immuable le concept de robot implique la faculté de perception de l'environnement l'analyse de ces informations la prise de décision et enfin la correction du comportement en fonction des informations captées sur l'environnement (cf. Figure 5.1). Ainsi ce qui distingue le robot d'un automatisme c'est précisément cette capacité d'appréhender son environnement et d'en établir une certaine représentation pour en tenir compte dans son comportement. En cela la perception de l'environnement constitue une fonction indispensable et caractéristique de tout robot.

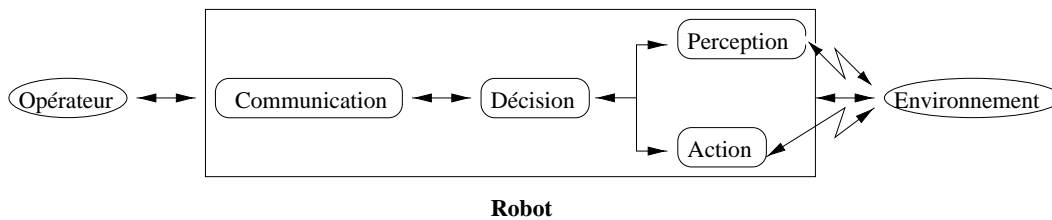


FIG. 5.1: *Fonctions de base d'un robot.*

La perception de l'environnement comme tout système de mesure peut constituer une source d'interaction supplémentaire avec cet environnement par exemple par un éclairage spécial pour caméra une projection de faisceau par un télémètre laser des échos ultra-sonores etc. L'interaction du robot avec l'environnement intervient donc sous deux formes distinctes : au niveau de la perception puis au niveau de l'action physique exercée sur l'environnement.

### 5.2.2 Perception de l'environnement

Elle peut se définir par l'ensemble des fonctions d'acquisition de mesures et de traitement d'informations permettant l'analyse et/ou la modélisation de l'environnement du

robot dans le but de prendre des décisions et de générer des commandes. Pour percevoir leur environnement les robots utilisent des capteurs d'environnement (cf. Figure 5.2).

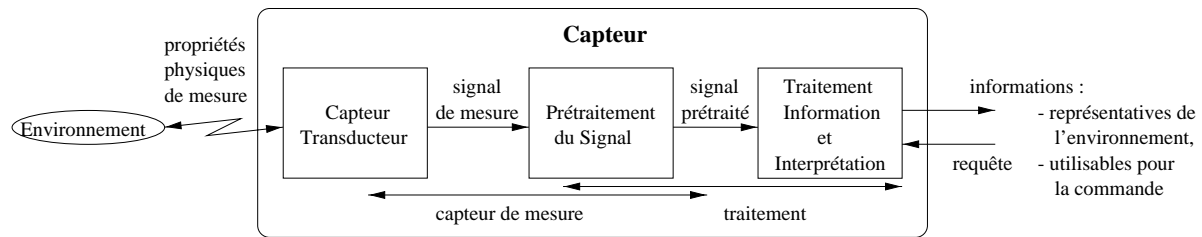


FIG. 5.2: Schéma de fonctionnement d'un capteur.

### 5.2.3 Commande adaptative vis-à-vis de l'environnement

L'évolution de la robotique dans le sens de l'autonomie du système se trouve étroitement liée à l'accroissement et à l'amélioration des capacités sensorielles. Le rôle des capteurs d'environnement est fondamental dans la mesure où ceux-ci contribuent à la réalisation de la commande et donc en définitive à l'accomplissement de la tâche effectuée par le robot malgré l'existence d'un environnement mal connu a priori et/ou évolutif.

L'objectif de la robotique est donc de sélectionner ou concevoir des dispositifs multi-capteurs adaptés à l'environnement et à la tâche assignée au robot puis de prendre en compte l'information de ces capteurs dans le système de commande du robot sous forme de *commande adaptative* par rapport à l'environnement.

### 5.2.4 Identification des besoins en capteurs

L'identification des besoins en capteurs résulte de l'analyse de la tâche attribuée au robot et des conditions d'environnement. Pour caractériser la tâche à accomplir on cherche à asservir de façon idéale le robot à cette tâche en présence d'incertitudes. Or il est très difficile d'extraire des grandeurs directement représentatives de la tâche. Cela conduit à modéliser l'interaction du robot avec l'environnement et à utiliser des capteurs d'environnement dont on peut extraire des informations utiles pour l'accomplissement de la tâche.

Cependant développer un système sensoriel capable de fournir des informations pertinentes nécessaires et suffisantes pour la bonne réalisation de la tâche dans un environnement inconnu et/ou évolutif est un problème complexe. Dans la pratique la tâche est simplifiée en décomposant le travail en différentes étapes qui mettent en œuvre des types d'interactions simples. Des capteurs spécialisés sont alors sélectionnés chacun pour mesurer un des paramètres nécessaires à la réalisation de la tâche. Ces paramètres sont généralement en nombre limité.

## 5.3 Capteurs

### 5.3.1 Rôle et fonctions

#### a) Acquisition et modélisation de l'environnement

Le capteur est un équipement de perception capable d'effectuer l'acquisition de mesures et l'analyse de la représentation de l'environnement réalisée à partir de ces mesures. Il permet ainsi l'élaboration des données pertinentes du point de vue de la décision et de la commande du robot. Le capteur peut aussi permettre de reconstruire un modèle de représentation de l'environnement utilisable pour la tâche à exécuter. La construction du modèle géométrique complet d'un environnement inconnu permet la planification de missions et l'optimisation de trajectoires. Toutefois l'identification de caractéristiques particulières sur l'environnement comme les formes les positions peut s'avérer souvent suffisante pour réaliser cette planification.

#### b) Exécution d'une tâche en présence d'incertitudes

Les capteurs doivent permettre l'exécution d'une tâche par le robot malgré la présence d'incertitudes sur l'environnement. Le système sensoriel est configuré en conséquence pour générer ou modifier les consignes de manière à rendre la tâche exécutable dans l'environnement réel observé. La commande par capteurs permet de réaliser un recalage ou un véritable asservissement sur certains paramètres de l'environnement. Cette fonction intervient directement dans le processus de correction de trajectoire en position et en force sans passer par le contrôle de haut niveau.

### 5.3.2 Classification

#### a) Les capteurs en robotique

Les capteurs d'environnement [And83a] sont encore appelés capteurs extéroceptifs ou capteurs externes par opposition aux capteurs internes ou proprioceptifs qui délivrent des informations sur l'état interne du robot. Ces capteurs extéroceptifs délivrent une information relative à l'environnement dans un but de reconnaissance de reconstruction d'un modèle ou relative aux interactions entre le robot et son environnement pour connaître les positions les forces.

Nous nous intéressons plus particulièrement aux interactions à distance. Les capteurs concernés peuvent être classés en deux catégories selon le type de perception employé. Le premier type de perception est la vision globale qui concerne l'analyse de scène et l'environnement dans son ensemble. Le second type de perception est la vision locale qui s'intéresse quant à elle à une portion limitée de l'espace pour des distances de 0 à environ 2 ou 3 mètres.

#### b) Perception globale

La perception globale de l'environnement en robotique met en œuvre des systèmes de vision. Ce sont des techniques de perception sans contact basées sur l'acquisition d'images

bidimensionnelles (2D) ou tridimensionnelles (3D). La vision 2D est la technique classique qui analyse une information de type image numérique 2D transmise par une caméra observant un environnement statique. La vision 3D désigne les techniques actives (par exemple à partir de télémétrie laser) ou passives (comme la stéréo-vision) qui permettent de percevoir le relief et d'élaborer un modèle 3D de l'environnement. Les systèmes de vision s'intéressent aux fonctions les plus complexes et permettent de traiter l'analyse de scène, la reconnaissance de formes, la reconnaissance de position et la localisation. Ils jouent également un rôle en inspection et contrôle.

De tels systèmes permettent de déterminer l'identité et la localisation d'objets cibles dans un repère de coordonnées absolues. Mais les informations obtenues sont denses et ne fournissent pas directement des relations (par exemple géométriques) entre le robot et son environnement. De plus, le temps de traitement devient prohibitif pour la commande en temps réel du robot, excepté dans le cas d'approche "asservissement visuel" qui relève de la perception visuelle locale.

### c) Perception locale

Les capteurs de perception locale mesurent directement les interactions entre le robot et son environnement. Les capteurs proximétriques opèrent sans contact physique. Ils sont localisés sur le robot en des points précis d'observation. Ils ont pour but d'informer non seulement sur la présence d'un objet dans le champ du capteur, mais peuvent également déterminer la position et l'orientation relative entre le robot et cet objet. Les capteurs proximétriques sont basés sur des techniques de mesure de distance plus ou moins précises. On distingue plusieurs types de capteurs selon la technologie utilisée, les principaux étant les capteurs optiques, les capteurs magnétiques et les capteurs ultra-sonores.

Cette perception locale permet de fournir des informations simples, directement utilisables, sans post-traitement et donc rapides, de réaliser une commande en "boucle fermée" à partir de mesures directes sur l'environnement, d'accroître l'adaptativité locale et ainsi de réduire le degré de précision nécessaire au robot, et enfin d'améliorer la dynamique de l'interaction par la rapidité des traitements.

### d) Complémentarité perception globale-perception locale

En fait, la perception globale et la perception locale de l'environnement sont pleinement complémentaires. L'objectif de la perception globale est de disposer d'informations de haut niveau, sous forme d'un modèle d'environnement réactualisé par des mesures sur site de travail. La perception locale vise, quant à elle, à délivrer des mesures spécialisées en temps réel pour asservir la trajectoire du robot sur l'environnement rencontré localement.

### 5.3.3 Discussion

En robotique, les informations de perception de l'environnement sont fournies aux robots par les caméras et les capteurs. La caméra fournit des informations globales et denses. Par conséquent, elle est employée pour des tâches générales, telles que la reconnaissance

d'objets dans l'environnement et la planification de trajectoire pour la navigation dans l'environnement. Cependant le problème majeur non négligeable des caméras provient de la complexité de l'analyse des images enregistrées qui nécessitent des calculs longs et compliqués pour extraire les informations acquises du monde.

D'autres capteurs sont alors utilisés pour faciliter la tâche de la caméra. Ces capteurs traitent des détails dont la caméra n'a plus à se soucier. Par exemple les capteurs de proximité donnent des informations simples et rapides sur la distance locale aux objets ce qui est utile pour éviter les collisions avec les obstacles.

Ainsi pour des raisons de simplicité et de rapidité de détection mais aussi parce que les capteurs sont plus légers et moins consommateurs d'énergie que les caméras la perception locale nous semble très avantageuse. Elle permet de plus d'avoir un contrôle local de niveau intermédiaire concernant la détection d'obstacles tendant ainsi à réduire la tâche du contrôle global de haut niveau qui concernera la planification de la trajectoire du robot. Nous souhaitons donc utiliser des capteurs proximétriques pour notre application. Nous présentons maintenant ces capteurs plus en détails.

## 5.4 Capteurs proximétriques

### 5.4.1 Physique du capteur

Un capteur de proximité est un mécanisme qui détecte les objets de manière locale en utilisant une interaction physique particulière avec l'environnement [Lau83]. Plusieurs modes d'interaction avec l'environnement existent dépendant des principes physiques mis en jeu. En général ce type de capteurs utilise l'interaction d'une onde avec la surface des objets de l'environnement. On distingue alors les ondes acoustiques (principalement ultra-sonores) et les ondes électromagnétiques dans le visible (ondes lumineuses) ou dans d'autres bandes (radar, lidar, infrarouge).

Un capteur consiste en un émetteur qui émet une onde et un récepteur qui reçoit l'onde réfléchi. Ses caractéristiques sont :

- l'encombrement : la forme du capteur
- la géométrie : la zone de détection et la résolution de l'image
- l'énergie : la puissance de l'émetteur, la sensibilité du récepteur et le bruit.

Le principe de détection de ces capteurs est celui de la propagation d'ondes et par conséquent s'appuie sur les propriétés intrinsèques des objets de réflexion des ondes reçues. En effet les objets sont plus ou moins capables de réfléchir l'énergie reçue selon les caractéristiques de leur surface. Nous étudions ces caractéristiques dans le paragraphe suivant.

### 5.4.2 Propriétés des surfaces

Nous présentons les propriétés des surfaces qui ont trait à la réflexion d'onde. Deux modèles physiques de base permettent de décrire les caractéristiques des surfaces. Le premier modèle est le modèle de *surface parfaitement diffuse*. Ce type de surface ré-émet entièrement l'énergie incidente dans toutes les directions  $\Gamma$  et ce  $\Gamma$  indépendamment de la direction originale de l'onde incidente (cf. Figure 5.3 (a)). Le second modèle est celui de *surface parfaitement spéculaire*. Ce type de surface ré-émet toute l'énergie incidente dans la direction symétrique de la direction originale de l'onde  $\Gamma$  par rapport à la normale de la surface (cf. Figure 5.3 (b)).

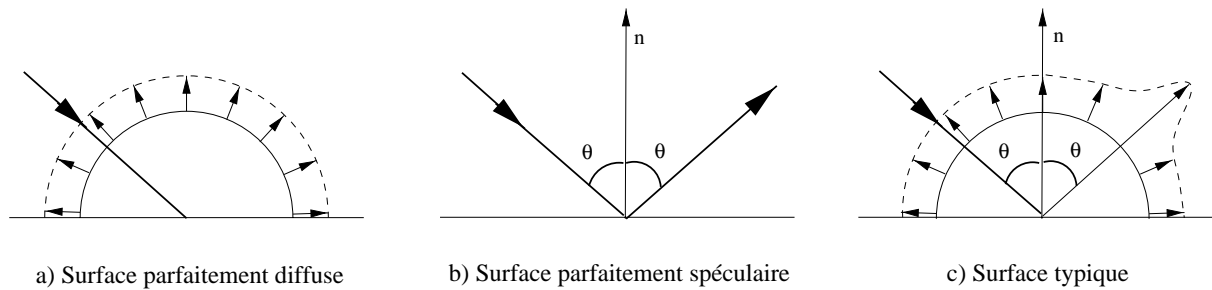


FIG. 5.3: Types de base de surfaces.

Ces deux modèles de surfaces caractérisent des modèles parfaits. En synthèse d'images  $\Gamma$  il existe un modèle plus complexe qui combine ces deux types de modèles  $\Gamma$  dans des proportions variées (cf. Figure 5.3 (c)). Ces proportions dépendent de la granularité de la surface : plus la surface est lisse  $\Gamma$  et plus l'énergie est réfléchie de manière spéculaire  $\Gamma$  tandis que plus la surface est rugueuse  $\Gamma$  et plus l'énergie est réfléchie de manière diffuse (cf. Figures 5.4 et 5.5).

D'autre part  $\Gamma$  la réflexion n'est pas le seul phénomène d'interaction d'une onde avec les objets. Il existe d'autres phénomènes  $\Gamma$  tels que l'absorption  $\Gamma$  la diffraction et l'interférence  $\Gamma$  qui ont pour conséquence une diminution de l'énergie réfléchie par rapport à celle reçue.

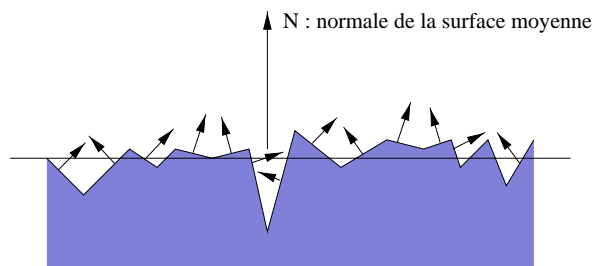


FIG. 5.4: Modèle par micro-facette de la surface de réflexion.

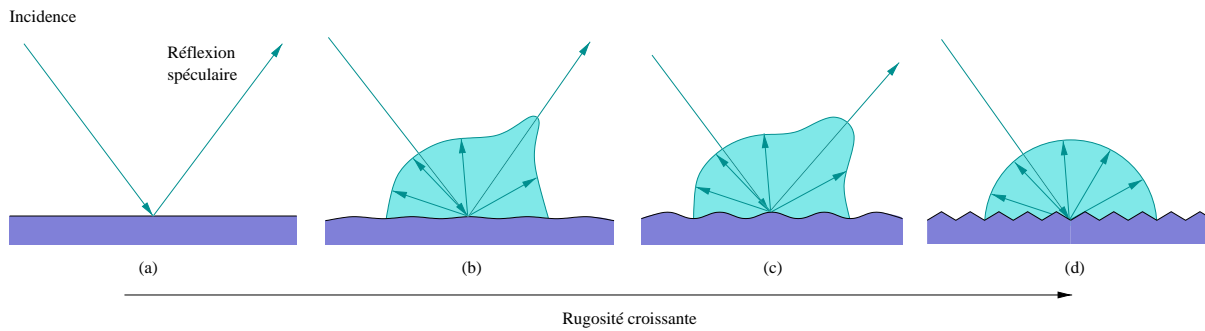


FIG. 5.5: *Réflexion comme fonction de la rugosité de la surface.*

### 5.4.3 Principes de détection

Le principe de base est la propagation d'onde. Une onde est émise par le capteur et propagée dans l'environnement. Si l'onde rencontre un obstacle elle est réfléchie dans une certaine direction. Cette direction dépend de la direction originale et des propriétés de l'onde ainsi que de la position de l'obstacle et des propriétés de sa surface. L'onde réfléchie est propagée à nouveau dans l'environnement. Si le capteur se trouve sur son trajet elle peut être reçue et analysée par celui-ci (cf. Figure 5.6).

Le trajet de l'onde n'est pas seul à rentrer en considération dans la réception de l'onde par le capteur. L'intensité de l'onde est tout aussi importante. En effet en dehors du vide toute onde propagée dans un milieu perd de l'énergie au fur et à mesure de son parcours. Ainsi l'intensité de l'onde diminue avec la distance parcourue. Or le récepteur possède un seuil de détection en-dessous duquel il n'est plus sensible à l'onde. Ainsi lorsque l'intensité de l'onde réfléchie est trop faible le récepteur ne détecte pas cette onde. Par conséquent tout ce qui se trouve à une distance trop éloignée ne peut être détecté puisque l'onde est atténuée par la distance. Il en découle une approximation justifiée par la localité de la détection qui est de ne pas considérer les réflexions multiples (cf. Figure 5.7). En effet la distance parcourue par l'onde est plus grande que si c'était une réflexion simple. Par conséquent soit l'onde n'est pas reçue car trop atténuée soit elle est reçue et comme rien n'indique qu'il y a eu réflexion multiple la distance est erronée correspondant à un obstacle virtuel plus éloigné.

Nous présentons dans ce qui suit quelques principes employés en robotique pour la détection des obstacles rencontrés par les ondes propagées dans l'environnement.

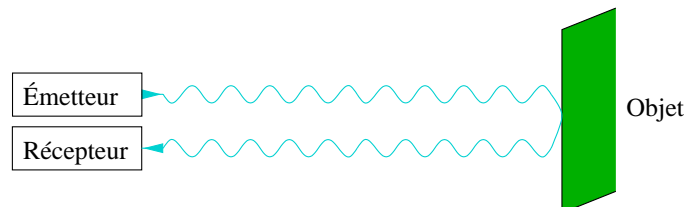


FIG. 5.6: *Réflexion d'onde simple.*

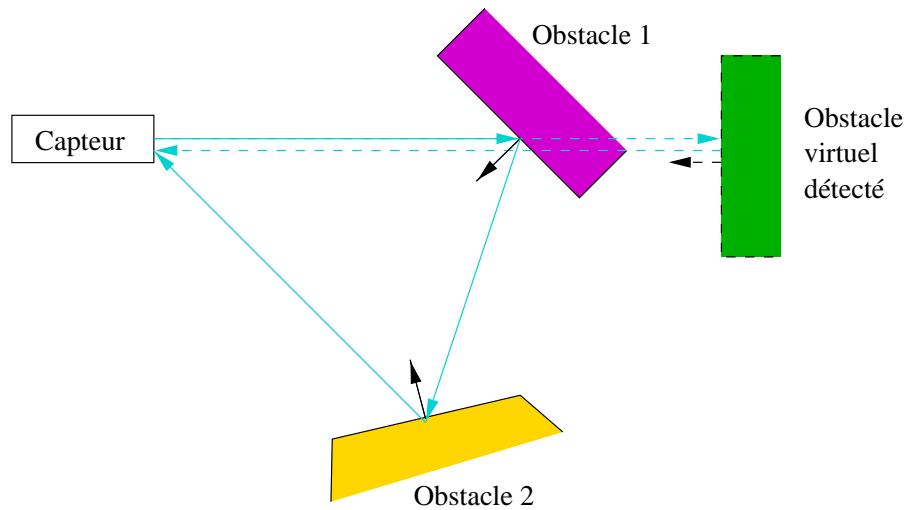


FIG. 5.7: Réflexion multiple d'onde.

### a) Mesure du temps de vol

Il s'agit de mesurer le temps de vol d'une impulsion (cf. Figure 5.8). Un émetteur émet une onde qui est soit perdue dans la nature, auquel cas le récepteur ne reçoit rien, soit réfléchi par un obstacle et alors elle peut être captée par le récepteur. Le temps  $\tau$  séparant l'émission de la réception de l'onde est mesuré et donne la distance  $D$  capteur-obstacle par l'équation :

$$D = \frac{1}{2} v \tau \quad (5.1)$$

avec  $v$  la vitesse de propagation de l'onde.

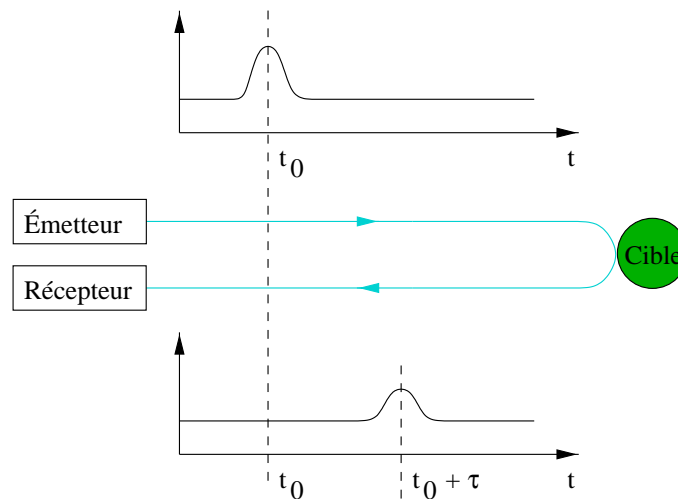


FIG. 5.8: Principe par mesure du temps de vol.



## b) Mesure de la phase

*Déphasage par temps de vol*

Pour de faibles distances il est impossible de mesurer le temps de vol d'une impulsion lumineuse car celui-ci est trop court. En effet l'équation (5.1) donne pour la vitesse de propagation de la lumière  $v = 3 \cdot 10^8 \text{ m/s}$  et une distance de  $D = 30 \text{ cm}$  un temps  $\tau = 1$  nanoseconde. Une méthode pratique est de moduler en fréquence le signal émis. La mesure de distance est calculée à partir de la mesure de la différence de phase entre le signal émis et celui reçu (cf. Figure 5.9). La différence de longueur de chemin parcouru ( $L_2 - L_1$ ) induit un déphasage entre les signaux reçus par le récepteur. Ce déphasage  $\Delta\Phi$  dépend de l'angle  $\theta$  de la distance  $d$  entre l'émetteur et le récepteur et de la longueur d'onde  $\lambda$  :

$$\Delta\Phi = \frac{2\pi(L_1 + L_2)}{\lambda} \text{ avec } L_2 = \sqrt{L_1^2 + d^2 + 2L_1d \sin(\theta)}$$

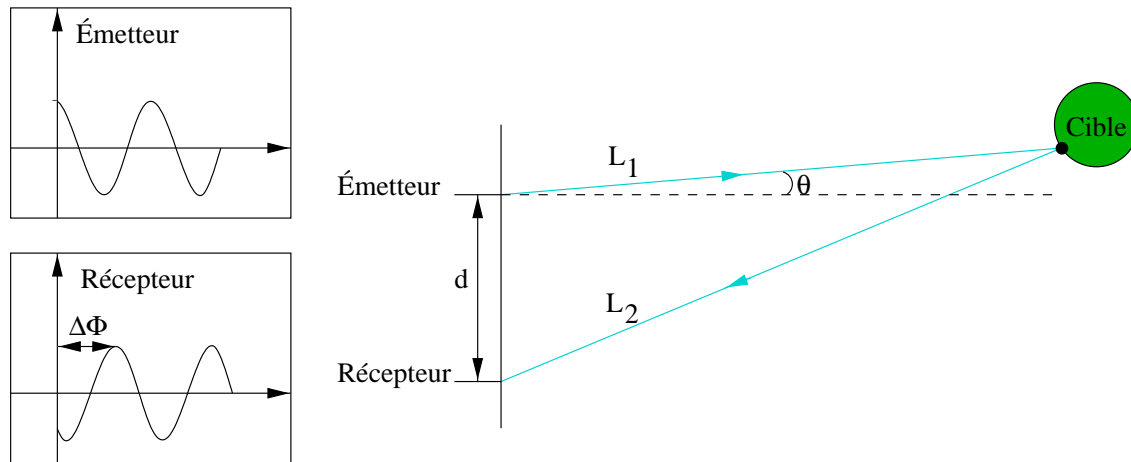


FIG. 5.9: Mesure de la différence de phase entre le signal émis et celui reçu.

*Déphasage par photométrie et géométrie*

La différence de phase implique l'utilisation d'une onde porteuse qui peut être modulée à différentes fréquences. En mesurant la différence de phase entre le signal transmis et le signal reçu après avoir été réfléchi par un objet la distance  $L_1$  peut être déterminée comme un nombre entier  $n_1$  inconnu de longueurs d'onde  $\lambda_1$  plus une fraction de longueur d'onde connue par la comparaison de phase  $\Delta\Phi_1$  :

$$2\pi n_1 + \Delta\Phi_1 = 2\pi \frac{L_1}{\lambda_1}$$

L'ambiguïté de  $n_1$  peut être levée en utilisant un intervalle de fréquences de modulation.

### c) Mesure de la réflectance

Les capteurs par réflectance mesurent l'énergie ré-émise par une cible. Un capteur de réflectance est composé d'un émetteur et d'un récepteur. Le signal de sortie représente l'intensité de lumière diffusée de la cible au récepteur qui dépend non seulement de la distance mais aussi des propriétés photométriques et de l'orientation locale de l'objet.

### d) Mesure par triangulation

Les capteurs par triangulation se servent de la distribution géométrique de points projetés pour calculer la distance et l'orientation. Le principe de base est d'utiliser un photo-récepteur (point, ligne ou plan) qui capte l'image de l'intersection d'un faisceau de lumière émis et d'une cible. Connaissant la position du point sur le récepteur ainsi que la géométrie de l'émetteur-récepteur la distance peut être calculée par (cf. Figure 5.10) :

$$D = f \left( 1 + \frac{x_0}{x} \right)$$

avec  $f$  la distance focale,  $x_0$  la distance émetteur-récepteur et  $x$  la position du point du faisceau détecté sur le détecteur.

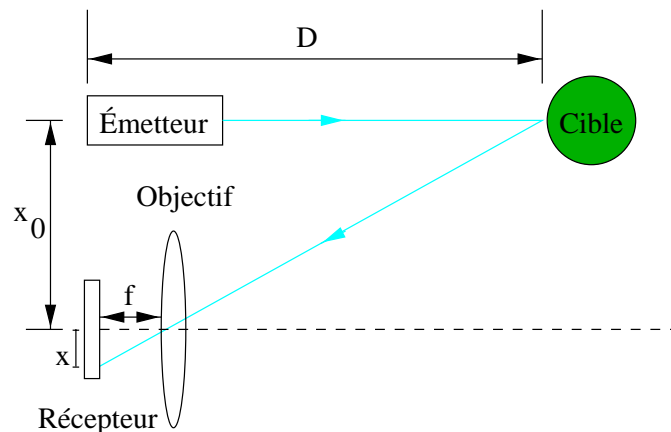


FIG. 5.10: *Mesure par triangulation.*

En utilisant un émetteur mobile ou plusieurs émetteurs en conjonction avec un récepteur 2D des informations supplémentaires peuvent être fournies comme la forme locale d'un objet ou l'orientation de la cible. Cependant la tendance est d'éviter les dispositifs de mesure mécaniques afin d'augmenter la fiabilité du mécanisme et de réduire sa taille totale. Les principaux problèmes des méthodes de triangulation sont les erreurs dues à la distorsion ou à la séparation du point lorsque le faisceau n'est pas une ligne idéale ainsi que les zones cachées qui peuvent apparaître lors de discontinuités abruptes.

#### 5.4.4 Caractéristiques des capteurs optiques et acoustiques

##### a) Capteurs à ultra-sons

###### *Faible directivité*

Les ondes ultra-sonores peuvent être de deux types soit des hautes fréquences (100-400 kHz) soit des basses fréquences (40-100 kHz). Le défaut des capteurs à ultra-sons est leur faible directivité (cf. Figure 5.11). Ce défaut peut être compensé par l'utilisation d'ondes de hautes fréquences qui sont plus directives que celles de basses fréquences (cf. Figure 5.12). Mais la portée est alors plus faible.

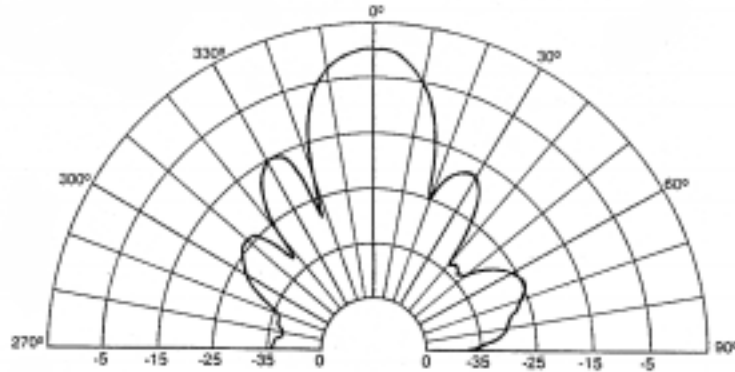


FIG. 5.11: *Forme typique de rayon ultra-son à 50 kHz (source RoboSoft).*

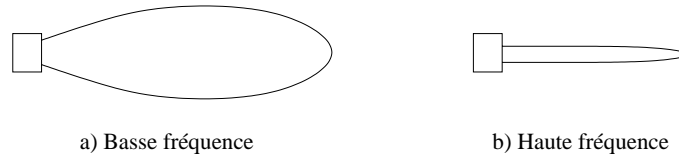


FIG. 5.12: *Lobes principaux d'émissions pour des capteurs à ultra-sons.*

###### *Types de surfaces*

Nous avons vu au paragraphe 5.4.2 qu'il existe deux types de base de surfaces : les surfaces diffuses et les surfaces spéculaires. Dans le cas des ultra-sons la nature de la réflexion des ondes sur les surfaces dépend de la longueur d'onde et de la granularité des surfaces. Une surface est diffuse pour une onde quand sa granularité est de l'ordre de la longueur d'onde  $\lambda$ . Elle est spéculaire dès que sa granularité est de l'ordre de  $10 \lambda$ . Pour les ondes à haute fréquence  $\lambda$  varie entre  $0.75 \text{ mm}$  et  $3 \text{ mm}$ <sup>1</sup>. Aussi la majorité des surfaces apparaissent comme un miroir. Le signal réfléchi est significatif uniquement lorsque l'angle entre la normale de la surface de l'objet et la direction du capteur est inférieur à  $15^\circ$  environ.

<sup>1</sup>Calcul de  $\lambda$  :  $\lambda[m] = \frac{300 \text{ [m/s]}}{\text{fréquence [Hz]}}$ , ce qui donne  $\lambda \in [0.75 \text{ mm}, 3 \text{ mm}]$  pour des hautes fréquences de 100 kHz à 400 kHz.

### *Distance minimale*

La distance minimale de détection d'un capteur à ultra-sons est supérieure à 10 cm en général. En effet il s'agit généralement du même dispositif qui émet le train d'ondes puis qui reçoit l'onde réfléchi. Ainsi les capteurs à ultra-sons possèdent une zone morte correspondant au lancement du train d'impulsion suivi de la commutation émetteur-récepteur. Une amélioration significative est obtenue dès lors que l'émetteur et le récepteur sont séparés.

### *Étalonnage*

Pour que la mesure soit précise un capteur à ultra-sons nécessite un étalonnage dans les conditions strictes d'utilisation. En effet la réflexion du faisceau dépend fortement de l'absorption par la surface réfléchissante (*albedo*) qui dépend de sa nature et de sa couleur et de l'angle entre le faisceau incident et la normale à la surface. Dès que ce faisceau n'est plus perpendiculaire à la surface les mesures sont fortement modifiées. De surcroît le bruit parasite de l'environnement perturbe facilement ces capteurs.

## **b) Capteurs optiques**

### *Sensibilité à l'environnement*

Un capteur optique peut être perturbé par un milieu pollué par des cendres la lumière ambiante qui modifie le signal reçu etc. D'autre part la mesure directe de distance et/ou d'orientation n'est pas possible lorsque certaines caractéristiques de la cible ne sont pas connues. Cependant il est possible de venir à bout de cette difficulté lorsque le mouvement relatif d'un capteur et de sa cible est connu.

### *Sensibilité à la nature de la cible*

Les capteurs utilisent la propriété de diffusion. Or certains objets métalliques agissent comme des miroirs et il existe un risque de saturation de lumière dans la réflexion spéculaire alors qu'aucun signal significatif n'apparaît pour les autres orientations relatives. De plus les objets hautement absorbants (les "corps noirs") peuvent conduire à un mauvais rapport signal-bruit.

## **5.4.5 Discussion**

Les capteurs proximétriques utilisés en robotique sont principalement de type ultra-sons ou optiques. Les inconvénients des capteurs à ultra-sons ont limité leur utilisation à la détection de proximité. Toutefois ces capteurs sont particulièrement simples et bon marché et sont largement employés en robotique mobile ou de manipulation mais plutôt en évitement d'obstacles c'est-à-dire en détecteur de présence sans avoir grande foi en la mesure de la distance. Les capteurs optiques basés sur une mesure de déphasage quant à eux permettent d'avoir une mesure de distance plus précise. Cependant un inconvénient majeur est le coût important de ces capteurs ce qui limite intrinsèquement leur nombre dans les applications robotiques.

Ces capteurs proximétriques renseignent donc sur la distance ou la présence des obstacles locaux au robot. Or pour que le robot puisse s'adapter à son environnement il est nécessaire de lui fournir des informations supplémentaires concernant la nature des objets rencontrés. En effet c'est par la connaissance de son environnement que le robot sera capable de s'adapter au mieux à son environnement. Il est donc nécessaire de pourvoir le robot en plus de la perception d'une autre fonctionnalité : la reconnaissance de formes.

## 5.5 Conclusion

Nous avons présenté des méthodes utilisées en robotique pour réaliser les fonctions de perception de l'environnement par le robot. La perception est réalisée par des capteurs qui peuvent être de divers types. Des caméras sont généralement utilisées pour traiter la perception globale de l'environnement pour des tâches complexes telles que la navigation. Nous préférons nous limiter à l'étude de capteurs plus simples qui réalisent une perception locale de l'environnement et permettent déjà d'avoir une idée de l'interaction à effectuer pour obtenir des informations sur l'environnement. L'avantage de ces capteurs est de fournir une information simple et rapide contrairement aux caméras qui fournissent des images complexes nécessitant des temps de traitement souvent lourds pour obtenir des informations pertinentes. Ainsi ces capteurs sont habituellement employés pour simplifier la tâche de la caméra en sous-traitant des problèmes particuliers. L'étude de la perception devra cependant être complétée ultérieurement par le cas des caméras.

Nous nous intéressons plus précisément aux capteurs proximétriques qui informent le robot sur son environnement local permettant notamment la détection d'obstacles afin d'éviter les collisions du robot avec les divers objets de l'environnement. Ces capteurs fournissent des informations de présence ou mieux de distance des obstacles. Ce dernier type d'information est généralement traité de manière à obtenir une meilleure compréhension de ce qui est détecté en l'occurrence en reconstruisant l'environnement perçu puis en reconnaissant les objets détectés. Ce traitement permet d'établir et de maintenir un modèle symbolique de l'environnement sur lequel peut s'appuyer le robot pour adapter son comportement à l'environnement considéré (cf. Annexe B).

Nous souhaitons fournir à notre robot virtuel ces fonctions de perception et de reconnaissance de l'environnement. En effet notre simulateur doit prendre en compte les interactions du robot avec son environnement de manière similaire à ce qui est fait dans la réalité. Nous proposons dans le chapitre suivant divers outils pour réaliser ces fonctions en modélisant tout d'abord des capteurs proximétriques au moyen d'outils provenant de l'infographie puis en analysant les données capteurs pour reconnaître l'environnement perçu.

# Chapitre 6

## Modélisation de la perception

### 6.1 Introduction

De manière générale l'utilité d'un robot repose sur les moyens d'actions qu'il possède dans un environnement. Or pour pouvoir agir il est nécessaire au robot d'avoir une connaissance sur cet environnement. Généralement cette connaissance ne lui est pas fournie directement. Il lui est nécessaire de l'acquérir au moyen de systèmes de perception plus ou moins complexes tels que les caméras ou les capteurs.

Le robot a en particulier besoin d'informations locales qui sont habituellement fournies par des capteurs proximétriques en robotique. En effet ces capteurs permettent de délivrer des informations simples et rapides sur la distance aux objets proches du robot. Ces informations sont prises en compte par le contrôle de haut niveau du robot pour modifier continûment le comportement du robot en fonction de l'environnement.

Nous souhaitons modéliser des capteurs optiques et à ultra-sons étudiés dans le chapitre précédent. Les caractéristiques générales des capteurs qui nous importent sont la taille pour avoir une idée de l'encombrement du capteur sur le robot ; la position sur le robot ; la direction de détection ; le seuil minimal et le seuil maximal déterminant ainsi son champ de détection ; et enfin le signal de sortie. Un intérêt particulier est accordé à la modélisation du signal. En effet le plus important est d'émuler la fonctionnalité du capteur c'est-à-dire trouver la réponse du capteur selon la distance à un obstacle.

Cependant ces capteurs ne délivrent qu'une information relative à la distance aux obstacles. Or pour pouvoir évoluer dans son environnement le robot doit reconnaître les objets qu'il rencontre. Aussi nous proposons des outils d'aide à l'analyse des données capteurs pour la reconnaissance de l'environnement.

Tout d'abord nous présentons les modèles de capteurs que nous avons développés. Ensuite nous proposons plusieurs outils complémentaires pour aider à déterminer la nature des objets rencontrés par le robot. Enfin nous présentons quelques résultats obtenus par des capteurs réels pendant la marche d'un sujet humain afin de valider nos modèles de capteurs.

## 6.2 Modèles de capteurs

Des modèles de capteurs ont déjà été mis en œuvre mais surtout dans le cadre de modélisation et de simulation pour des systèmes de CAO pour la robotique et la téléopération [And83b]. Cependant nous nous intéressons plutôt à la synthèse d'images comme dans [Bou86]. Ainsi pour modéliser l'effet des capteurs nous utilisons des méthodes graphiques qui renseignent sur la distance. En synthèse d'images ce qui se rapproche le plus des ondes lumineuses ou sonores est le modèle d'éclairage (cf. Annexe A). Nous pouvons considérer qu'un capteur simulé émet un rayon. Il faut trouver mathématiquement l'intersection entre le rayon et l'obstacle pour en déduire la distance à cet obstacle. Ensuite nous appliquons un post-traitement pour tenir compte de la distorsion du signal due à la distance et à l'angle d'incidence ainsi que du bruit dû aux erreurs et/ou incertitudes liées au capteur.

### 6.2.1 Outils graphiques

La programmation graphique est réalisée avec l'ensemble d'outils 3D orientés objets *Open Inventor*<sup>TM</sup> et *Open GL*<sup>TM</sup> sur des stations Silicon Graphics. Deux outils graphiques nous intéressent : la caméra et le lancer de rayon.

#### a) Caméra

Un objet *caméra* génère une image d'une partie de la scène depuis la position et la direction de cet objet *caméra* dans la scène. Une seule *caméra* est utile pour rendre la scène mais d'autres *caméras* peuvent être ajoutées. Les images alors captées peuvent être affichées soit dans la fenêtre principale représentant la scène soit dans des fenêtres annexes hors-écran. Il existe plusieurs types de *caméras* mais celle qui nous intéresse est la *caméra perspective* (cf. Figure 6.1). Une *caméra perspective* émule les dispositifs optiques standard (appareil photo chambre noire etc.) Les objets éloignés apparaissent plus petits. Le volume de vue formé par une *caméra perspective* est une pyramide tronquée. Les paramètres associés à l'objet *caméra perspective* sont donnés dans la table 6.1.

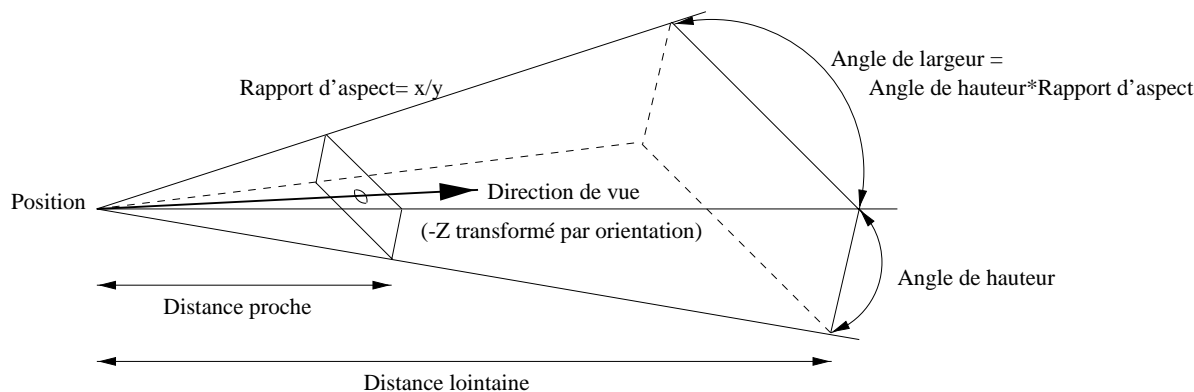


FIG. 6.1: *Caméra perspective.*

<i>position</i> :	c'est la localisation du point de vue de la caméra.
<i>orientation</i> :	c'est l'orientation de la direction de vue de la caméra. Cela décrit la rotation de la caméra par rapport à l'orientation initiale de la caméra, à savoir de (0., 0., 1.) vers l'origine (0., 0., 0.), la direction vers le haut étant (0., 1., 0.).
<i>rapport d'aspect</i> :	il s'agit du rapport largeur sur hauteur de la vue de la caméra. Cette valeur doit être à 1.33 pour un format télévision.
<i>distance proche</i> :	c'est la distance du point de vue de la caméra au premier plan de coupure (le plus proche).
<i>distance lointaine</i> :	c'est la distance du point de vue de la caméra au second plan de coupure (le plus éloigné).
<i>angle de hauteur</i> :	il spécifie l'angle vertical du volume de vue de la caméra.
<i>angle de largeur</i> :	il spécifie l'angle horizontal du volume de vue de la caméra. Il est directement déterminé par l'angle de hauteur et le rapport d'aspect : $angle\ de\ largeur = angle\ de\ hauteur * rapport\ d'aspect.$

TAB. 6.1: *Champs d'un objet caméra perspective.*



Pendant l'action de rendu de la scène, la *caméra* est placée dans la scène selon sa position et son orientation spécifiées. La *caméra* crée un volume de vue basé sur les plans de coupe proche et éloigné, le rapport d'aspect et l'angle de hauteur. Un volume de vue ou vue tronquée est un volume à six côtés contenant la géométrie à voir. Les objets hors du volume de vue ne sont pas pris en compte. Ensuite ce volume de vue 3D est projeté en une image 2D en tenant compte de la perspective de manière analogue à un instantané qu'un appareil photographique aurait fait d'une scène réelle. Cette projection 2D est affichée dans une fenêtre sur l'écran.

Cette projection représente une image de manière similaire à une image par points (*bitmap*) mais au lieu de contenir un seul bit pour chaque pixel, cette image peut contenir bien plus d'informations, notamment la couleur (RGB = (*red*, *green*, *blue*) = (rouge, vert, bleu)), la transparence (*alpha*), la profondeur (*Z*) ou la luminance (*L*) quand on ne travaille pas en couleur. Ces données des pixels peuvent être manipulées directement grâce à des actions de lecture, d'écriture, de copie. La lecture permet de récupérer les informations concernant l'image en lisant dans le tampon d'image (*frame buffer*). L'écriture est l'action opposée : une image a été calculée ou modifiée et on l'écrit dans le tampon d'image de manière à l'afficher dans la fenêtre. On peut enfin copier directement un tampon d'image dans un autre tampon d'image.

## b) Lancer de rayon

*Open Inventor*<sup>TM</sup> offre la possibilité d'appliquer des actions sur les objets de la scène. Notamment, il existe une action de sélection par rayon qui permet de trouver des objets le long d'un rayon. Cette action donne tous les objets rencontrés par le rayon de sélection, triés du plus proche au plus éloigné. Par défaut, tous les objets de la scène sont sélectionnables, on peut cependant rendre certains objets invisibles. Avant d'appliquer l'action de sélection par rayon, on définit le rayon grâce auquel s'effectue la sélection, par un point de départ, un vecteur directeur, un plan de coupe proche et un plan de coupe éloigné pour les objets sélectionnés (cf. Figure 6.2) et on spécifie si l'on désire tous les objets touchés par le rayon ou seulement l'objet le plus proche, s'il existe.

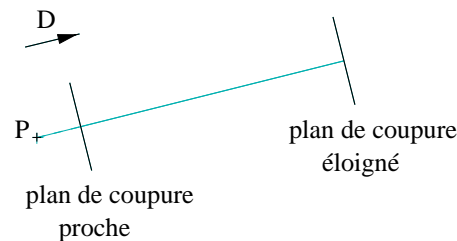


FIG. 6.2: Définition du rayon pour l'action de sélection.

Les résultats de la sélection sont stockés sous forme d'une liste éventuellement vide de points sélectionnés. Un point sélectionné représente un point de la surface d'un objet qui a été touché par le rayon. Ce point sélectionné contient diverses informations, celles qui nous intéressent étant le point d'intersection, dans le repère du monde ou dans le

repère local à l'objet  $\Gamma$  et la normale à la surface en ce point d'intersection  $\Gamma$  dans le repère du monde ou dans le repère local à l'objet.

### 6.2.2 Modélisation des signaux capteurs

Nous souhaitons utiliser les outils graphiques précédemment présentés  $\Gamma$  afin de calculer à chaque instant  $\Gamma$  la distance capteur-obstacle. Cependant  $\Gamma$  il faut transformer cette distance exacte en une valeur raisonnable correspondant à la sortie d'un capteur dans un cas réel. Aussi  $\Gamma$  nous décrivons dans ce paragraphe  $\Gamma$  la mise en œuvre de ces outils graphiques ainsi que les traitements effectués  $\Gamma$  pour modéliser la propagation d'onde  $\Gamma$  de manière à réaliser des modèles de capteurs réels.

Un capteur optique envoie un train d'impulsions. Nous modélisons ce mode de fonctionnement par un lancer de rayon  $\Gamma$  ce qui correspond à une détection ponctuelle des obstacles. Aussi appelons-nous ce type de capteur modélisé  $\Gamma$  un capteur 1D (cf. Figure 6.3 (a)). Nous pouvons également utiliser ce type de capteur pour émuler un capteur à balayage par exemple  $\Gamma$  ou plus simplement en prenant une batterie de capteurs  $\Gamma$  de manière à couvrir un plan. Ainsi  $\Gamma$  nous obtenons un capteur 2D (cf. Figure 6.3 (b)).

L'autre type de capteurs que nous souhaitons modéliser est le capteur à ultra-sons. Ce capteur est beaucoup moins directionnel que le capteur optique. En effet  $\Gamma$  son champ d'action est plus étendu  $\Gamma$  volumique (cf. Figure 5.11). Nous modélisons ce mode de fonctionnement par une caméra  $\Gamma$  permettant ainsi de détecter des obstacles dans un certain volume. Ce capteur est ainsi de type 3D (cf. Figure 6.3 (c)).



a) Ligne de vision de capteur 1D. b) Plan de vision de capteur 2D. c) Cône de vision de capteur 3D.

FIG. 6.3: Types de perception.

#### a) Calcul de la distance par mise en œuvre graphique

##### Capteur 1D

Pour modéliser le rayon d'un capteur 1D  $\Gamma$  nous utilisons le lancer de rayon. Nous définissons le rayon par sa position d'origine  $P$   $\Gamma$  sa direction  $\vec{D}$   $\Gamma$  son seuil minimal  $d_{min}$   $\Gamma$  et son seuil maximal  $d_{max}$  de détection. L'action de sélection permet de calculer l'intersection du rayon ainsi défini avec les objets de la scène. Si le rayon ne touche aucun objet  $\Gamma$  alors aucun point n'est sélectionné  $\Gamma$  et par conséquent  $\Gamma$  aucun objet n'est détecté. Sinon  $\Gamma$  nous récupérons les coordonnées relatives du premier point sélectionné  $S$  par le rayon  $\Gamma$  et nous en déduisons la distance  $d$  à l'obstacle détecté en calculant la norme du vecteur  $\vec{PS}$  (cf. Figure 6.4 (a)) :

$$d = \|\vec{PS}\|$$

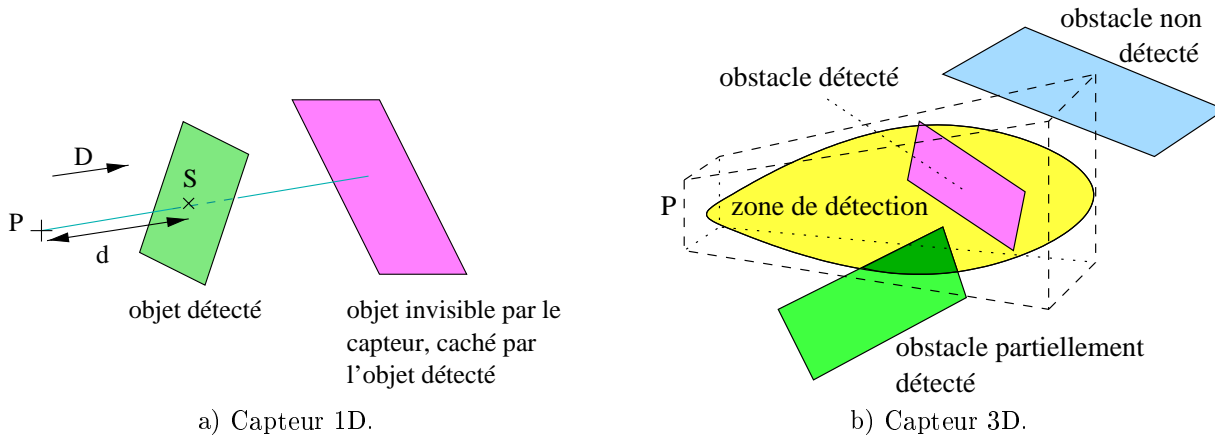


FIG. 6.4: Calcul de la distance à l'obstacle.

### Capteur 3D

De la même manière que dans [NRTMT95] pour modéliser la vision synthétique nous utilisons le rendu câblé pour obtenir les informations sur la distance aux objets. Pour modéliser un capteur 3D nous utilisons le volume de vue donné par la caméra perspective. Nous définissons tout d'abord les paramètres de la caméra : sa position  $P$  son orientation  $\vec{D}$  les distances proche  $d_{min}$  et éloignée  $d_{max}$  l'angle de hauteur  $h$  et l'angle de largeur  $l$ . Ensuite nous appliquons une action de rendu de la scène à travers cette caméra dont le résultat est une image 2D ayant plusieurs informations (cf. paragraphe 6.2.1). Nous récupérons la composante de profondeur de l'image c'est-à-dire les coordonnées  $z$  de chaque pixel de l'image qui sont contenues dans le tampon de profondeur ( $z$ -buffer ou *depth buffer*). Nous obtenons ainsi une carte de distances ou carte de profondeur pour tous les objets détectés dans le volume de vue de la caméra.

Toutefois un capteur à ultra-sons n'a pas un champ de vue de forme pyramidale tronquée. D'une part il faut choisir un rapport d'aspect égal à 1 car la largeur du volume et la hauteur sont égales ( $l = h$ ) et d'autre part le champ de vue du capteur est plutôt ellipsoïdal. Nous appliquons donc un filtre à la carte de distances de manière à ignorer tous les pixels correspondant à des objets ne pouvant être détectés par le capteur. Chaque pixel contient un point  $R = (x, y, z)$  d'un objet s'il existe  $z = +\infty$  sinon. Ces coordonnées sont données dans le repère de la caméra. Nous déterminons la distance maximale  $z_{max}$  associée au pixel par rapport aux coordonnées  $(x, y)$  de manière à définir une enveloppe (cf. Figure 6.4 (b)). Si  $z > z_{max}$  alors l'objet est hors de l'enveloppe il est ignoré  $z$  prend la valeur  $+\infty$  et le pixel est noir. Si  $z \leq z_{max}$  alors l'objet se trouve à l'intérieur de cette enveloppe il est donc traité puis affiché dans le pixel de l'image correspondant.

Pour calculer la valeur  $z_{max}$  associée à chaque couple  $(x, y)$  nous déterminons l'angle  $\theta$  entre le vecteur  $\vec{OR}$  et l'axe  $(OZ)$  de vecteur directeur  $\vec{u}_z$ . Ensuite nous définissons  $z_{max}$  comme étant la distance maximale du capteur multipliée par le cosinus de cet angle élevé à la puissance  $\nu$  pour agir sur l'aplatissement de l'enveloppe (par exemple  $\nu = 2$ )

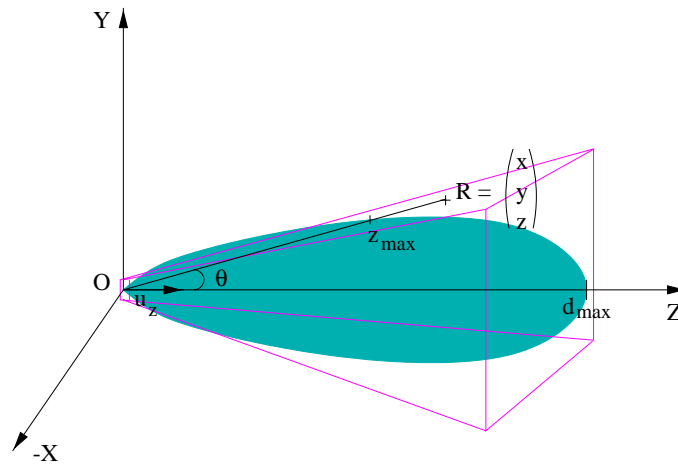


FIG. 6.5: Calcul de l'enveloppe pour définir la zone de détection du capteur 3D.

(cf. Figure 6.5) :

$$O = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad R = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad u_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$OR = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = (\overrightarrow{OR}, \vec{u}_z)$$

$$\cos(\theta) = \frac{z}{OR}$$

$$z_{max} = d_{max} \times \cos^\nu(\theta) = d_{max} \times \left(\frac{z}{OR}\right)^\nu$$

### Discussion

Ces caractéristiques sont valables pour des capteurs idéaux mais demeurent incomplètes pour notre étude. Nous souhaitons en effet modéliser des capteurs réels. Nous devons donc apporter des modifications au signal de sortie actuel de manière à tenir compte des imperfections du capteur ou des perturbations liées à l'environnement. Nous décidons de tenir compte de l'atténuation du signal selon la distance selon l'angle d'incidence du rayon sur l'obstacle ainsi que du bruit intrinsèque au capteur.

#### b) Atténuation du signal selon la distance

Le premier type d'atténuation pris en compte est l'atténuation en fonction de la distance. C'est le cas pour les capteurs de réflectance. Dans la réalité un capteur de réflectance fournit en sortie un signal électrique qui doit être interprété pour connaître

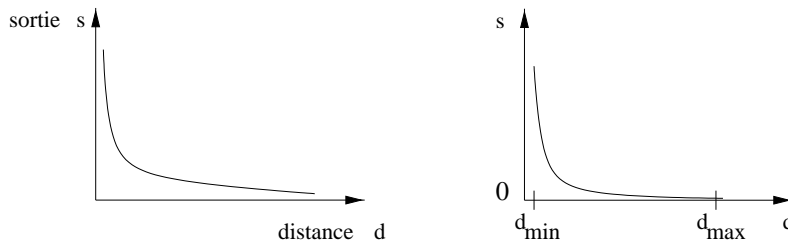
la distance capteur-obstacle. En synthèse d'images c'est cette distance que nous connaissons. Comme nous souhaitons simuler la réalité il est nécessaire de transformer cette distance en un signal de sortie réaliste. Nous altérons donc la distance calculée par le capteur pour rendre la sortie réaliste.

Contrairement aux autres types de capteurs proximétriques qui donnent directement la distance la sortie d'un capteur de réflectance est proportionnelle à l'intensité du signal laquelle est inversement proportionnelle à la distance (cf. Figure 6.6). En effet l'intensité de l'onde s'atténue avec la distance<sup>1</sup> (cf. Figure 6.6 (a)) :

$$s(d) = \frac{k}{d}$$

avec  $s(d)$  la sortie capteur fonction de la distance  $d$  et  $k$  une constante. Cependant nous voulons que le signal capteur s'annule à partir de la distance qui correspond au seuil maximal du capteur. Nous seuillons donc le signal à la distance maximale en utilisant la formule suivante pour modéliser le signal (cf. Figure 6.6 (b)) :

$$s(d) = \begin{cases} \frac{(d_{max}-d)}{d^2} & \text{si } d_{min} < d < d_{max} \\ 0 & \text{si } d < d_{min} \text{ ou } d > d_{max} \end{cases}$$



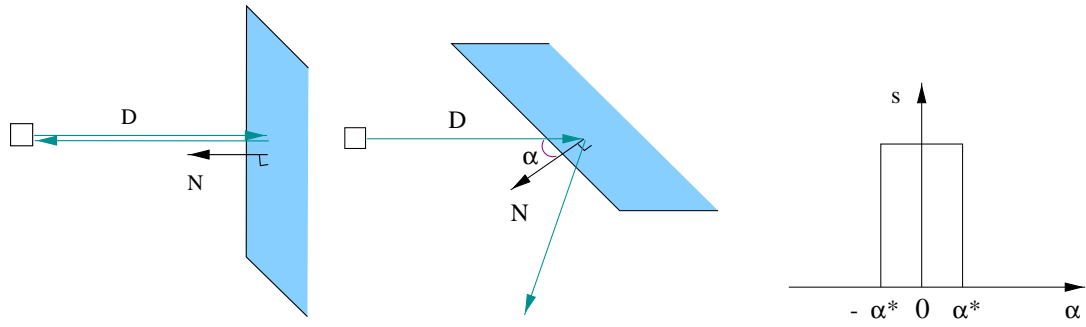
a) Signal-réponse du capteur de réflectance, inversement proportionnel à la distance. b) Seuillage du signal aux distances  $d_{min}$  et  $d_{max}$ .

FIG. 6.6: Atténuation du signal en fonction de la distance.

### c) Atténuation du signal selon l'angle d'incidence

Nous souhaitons prendre en compte l'angle d'incidence du rayon émis par le capteur avec l'obstacle. En effet nous avons vu que la sortie des capteurs optiques mais surtout à ultra-sons dépendent fortement de cet angle. Pour prendre en compte cette caractéristique nous devons connaître la normale de la surface au point de détection. L'angle d'incidence est alors l'angle entre cette normale et la direction du rayon. Si cet angle est suffisamment faible le rayon a une chance d'être reçu par le capteur (cf. Figure 6.7 (a)). Dans le cas contraire le rayon n'est jamais reçu (cf. Figure 6.7 (b)).

<sup>1</sup>La loi de dispersion se fait en  $\frac{1}{d^2}$  pour l'énergie, car celle-ci est une quantité conservée. L'énergie étant égale au carré de l'intensité - lumineuse ou sonore - ( $E = I^2$ ), implique que l'intensité diminue en  $\frac{1}{d}$ . Le signal étant fonction de l'intensité (généralement,  $s = I \cos \omega t$ ), le signal diminue donc en  $\frac{1}{d}$ .



a) Obstacle détecté. b) Obstacle non détecté. c) Filtre sur l'angle d'incidence.

FIG. 6.7: Importance de l'incidence dans la réflexion du rayon par l'obstacle.

### Capteur 1D

Pour le capteur 1D  $\Gamma$  en même temps que nous récupérons les coordonnées du point sélectionné  $P \Gamma$  nous récupérons la normale  $\vec{N}$  de la surface en ce point  $P$ . Nous pouvons alors calculer l'angle d'incidence  $\alpha$  du rayon de direction  $\vec{D}$  avec la normale de la surface  $\vec{N}$  de la manière suivante :

$$\alpha = \arccos(-\vec{D} \cdot \vec{N})$$

### Capteur 3D

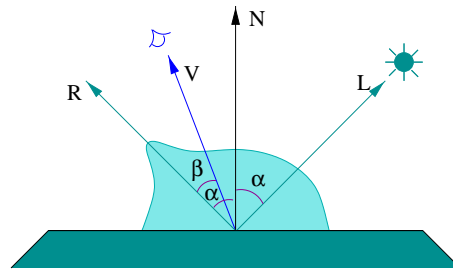
Pour le capteur 3D  $\Gamma$  en même temps que nous récupérons le tampon de profondeur correspondant à l'image détectée par la caméra  $\Gamma$  nous récupérons le tampon de couleurs. En effet  $\Gamma$  l'intensité de la lumière qui donne la couleur aux facettes des objets est fonction de l'angle d'incidence du rayon lumineux sur chaque facette  $\Gamma$  étant donné que nous utilisons comme modèle d'éclairage le modèle de Phong (cf. Annexe A). Ce modèle permet de calculer l'intensité lumineuse  $I$  en chaque point de la scène (cf. Figure 6.8)  $\Gamma$  par l'équation :

$$I = k_a I_a + I_i (k_d (\vec{L} \cdot \vec{N}) + k_s (\vec{R} \cdot \vec{V})^n) \quad (6.1)$$

$$= k_a I_a + I_i (k_d \cos \alpha + k_s \cos^n \beta) \quad (6.2)$$

où  $k_a \Gamma k_d \Gamma k_s$  sont respectivement les coefficients des termes ambiant  $\Gamma$  diffus et spéculaire  $\Gamma$   $I_a$  est l'intensité de la lumière ambiante  $\Gamma$   $I_i$  est l'intensité de la lumière incidente  $\Gamma$   $\vec{L}$  la direction de la lumière incidente  $\Gamma$   $\vec{N}$  la normale à la surface  $\Gamma$   $\alpha$  l'angle entre les deux vecteurs  $\vec{L}$  et  $\vec{N}$   $\Gamma$   $\vec{R}$  la direction de la lumière réfléchie  $\Gamma$  miroir de celle incidente par rapport à la normale de la surface  $\Gamma$   $\vec{V}$  la direction de vue  $\Gamma$   $\beta$  l'angle entre les deux vecteurs  $\vec{R}$  et  $\vec{V}$   $\Gamma$  et  $n$  un coefficient d'atténuation pour le terme spéculaire.

L'équation (6.2) décrit l'intensité d'une facette directement en fonction de l'intensité de la lumière ambiante  $\Gamma$  et de celle de la lumière incidente qui est modulée par un terme diffus ( $k_d \cos \alpha$ ) et un terme spéculaire ( $k_s \cos^n \beta$ ).

FIG. 6.8: *Modèle d'éclairage.*

Nous utilisons ce modèle d'éclairage afin de trouver la normale de chaque facette. Nous lançons un rayon lumineux depuis l'origine de la caméra  $\Gamma$  et dans sa direction  $\Gamma$  nous récupérons dans le tampon de couleurs l'intensité pour chaque point des objets détectés  $\Gamma$  et nous souhaitons en déduire la normale des surfaces des objets en chaque point. Cependant  $\Gamma$  si nous gardons la scène telle quelle  $\Gamma$  les propriétés qui ont été définies initialement pour les couleurs des objets  $\Gamma$  vont perturber les valeurs des normales que nous pourrions trouver  $\Gamma$  de par l'équation (6.2). Aussi appliquons-nous une phase de rendu monochrome sur toute la scène  $\Gamma$  phase expliquée ci-dessous. Une fois que les valeurs des intensités sont récupérées dans le tampon de couleurs  $\Gamma$  nous rétablissons le rendu normal.

Le rendu monochrome est choisi de manière à simplifier l'équation (6.2)  $\Gamma$  pour trouver le plus rapidement et le plus directement possible  $\Gamma$  la normale de chaque facette. Nous décidons alors de ne considérer que le terme diffus  $\Gamma$  en prenant les termes ambiant et spéculaire nuls pour tous les objets de la scène. L'intensité du rayon lumineux  $I_i$  est choisie à 1. Nous donnons aussi la valeur 1 au terme diffus  $k_d$  de chaque objet. L'équation (6.2) est alors réduite :

$$I = \vec{L} \cdot \vec{N} = \cos \alpha \quad (6.3)$$

Puisque le rayon ne peut rentrer dans un objet  $\Gamma$  l'angle d'incidence n'est défini que dans un demi-espace délimité par la surface de l'objet. Ainsi  $\Gamma$   $\alpha$  ne peut être que dans l'intervalle  $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$   $\Gamma$  et nous avons :

$$\alpha = \arccos I \quad (6.4)$$

Ainsi  $\Gamma$  à partir de l'intensité lumineuse  $I$  en chaque pixel  $\Gamma$  récupérée dans le tampon de couleurs  $\Gamma$  nous calculons aisément l'angle d'incidence  $\alpha$  en chaque pixel.

### **Discussion**

Une fois que nous connaissons l'angle d'incidence du rayon avec l'obstacle  $\Gamma$  nous appliquons un filtre passe-bande idéal sur cet angle (cf. Figure 6.7 (c)). De cette manière  $\Gamma$  si l'angle d'incidence est compris dans l'intervalle  $[-\alpha^*, \alpha^*]$   $\Gamma$  avec  $\alpha^*$  dépendant du type de l'onde considérée  $\Gamma$  le rayon est alors détecté. Sinon  $\Gamma$  le rayon n'est pas détecté.





## f) Discussion

Ce signal synthétisé est encore loin de la réalité. Pour qu'il soit encore plus précis il faudrait tenir compte des propriétés des surfaces des objets réfléchissants plus en détails des réflexions multiples du bruit relatif au type d'ondes au type de surfaces<sup>2</sup> etc. En outre nous ne nous sommes pas intéressés à la dynamique des capteurs c'est-à-dire à leur temps de réponse car nous avons supposé que les fréquences utilisées étaient suffisamment élevées pour avoir des réponses quasi instantanées.

Une amélioration consiste à considérer la couleur du bruit qui correspond à l'introduction d'incertitudes dans les résultats à des intervalles de temps réguliers. Pour cela il faut regarder la courbe résultante de la transformée de Fourier du signal. Si chaque fréquence a la même énergie le bruit est blanc (cf. Figure 6.9 (a)). Un autre type de bruit est le bruit coloré pour lequel une fréquence majoritaire systématique apparaît dans la courbe et des erreurs apparaissent alors à des intervalles de répétition particuliers pour cette fréquence (cf. Figure 6.9 (b)). Un bruit coloré particulier est le bruit rose qui apparaît lorsque l'on veut garder une énergie constante pour l'ensemble des fréquences. Or les hautes fréquences transportant généralement plus d'énergie on diminue ces fréquences (cf. Figure 6.9 (c)).

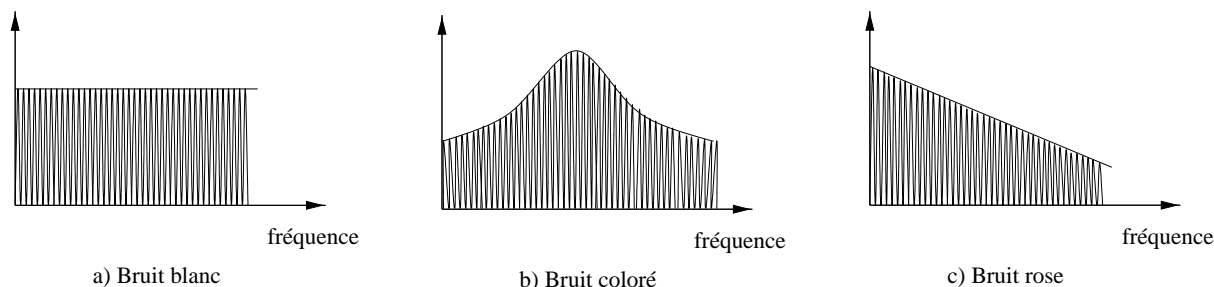


FIG. 6.9: Couleurs du bruit.

### 6.2.3 Simulation des capteurs

#### a) Interface utilisateur

Nous venons de modéliser différents types de capteurs. Nous présentons maintenant les capteurs simulés basés sur des capteurs réels. Tout d'abord nous avons développé une interface pour permettre à l'utilisateur de faire une sélection des capteurs qu'il désire placer sur le robot avec choix des caractéristiques de ces capteurs (cf. Figure 6.10). Un capteur est ajouté sur le robot après avoir choisi les valeurs des caractéristiques suivantes :

- support du capteur
- type de capteur (1D 2D 3D) d'où l'affichage de la fenêtre de dialogue du type de capteur choisi

<sup>2</sup>Plus une facette est petite, plus elle disperse le rayon. Par conséquent, le bruit augmente inversement proportionnellement à la taille de la facette de l'objet.

- taille du capteur  $\Gamma$
- position et orientation du capteur  $\Gamma$  par rapport à son support  $\Gamma$
- seuil minimal et seuil maximal  $\Gamma$
- paramètres pour l'atténuation selon la distance  $\Gamma$
- paramètres pour l'atténuation selon l'incidence  $\Gamma$
- paramètres pour le bruit.

Nous maintenons une liste des capteurs de manière à pouvoir accéder à nouveau aux modèles des capteurs  $\Gamma$  pour les modifier ou les retirer.

Des fonctionnalités supplémentaires ont été ajoutées  $\Gamma$  de manière à vérifier la détection réalisée. Dans le cas du lancer de rayon  $\Gamma$  nous affichons effectivement ce rayon  $\Gamma$  pour visualiser les moments où le rayon touche les obstacles (cf. Figure 6.11). Dans le cas de la caméra  $\Gamma$  nous affichons dans une fenêtre hors-écran le tampon de profondeur relatif à la caméra  $\Gamma$  en niveaux de gris. Nous visualisons ainsi les distances aux objets en dégradé de gris : plus le pixel est foncé  $\Gamma$  plus le point correspondant à ce pixel est proche du capteur (cf. Figure 6.12).

### b) Modélisation de capteurs réels

Nous avons simulé des télémètres laser  $\Gamma$  des capteurs à ultra-sons et des capteurs de réflectance. Nous nous servons de l'interface présentée ci-dessus pour choisir les types de capteurs à ajouter sur le robot  $\Gamma$  en précisant chacun des paramètres du capteur : sa taille  $\Gamma$  sa position et son orientation  $\Gamma$  son champ de vue  $\Gamma$  les influences de la distance et de l'angle d'incidence  $\Gamma$  et les paramètres pour le bruit. Pour chaque modèle de capteur présenté  $\Gamma$  nous donnons ses paramètres spécifiques.

#### *Télémètre laser*

C'est un capteur de type 1D avec les paramètres suivants :

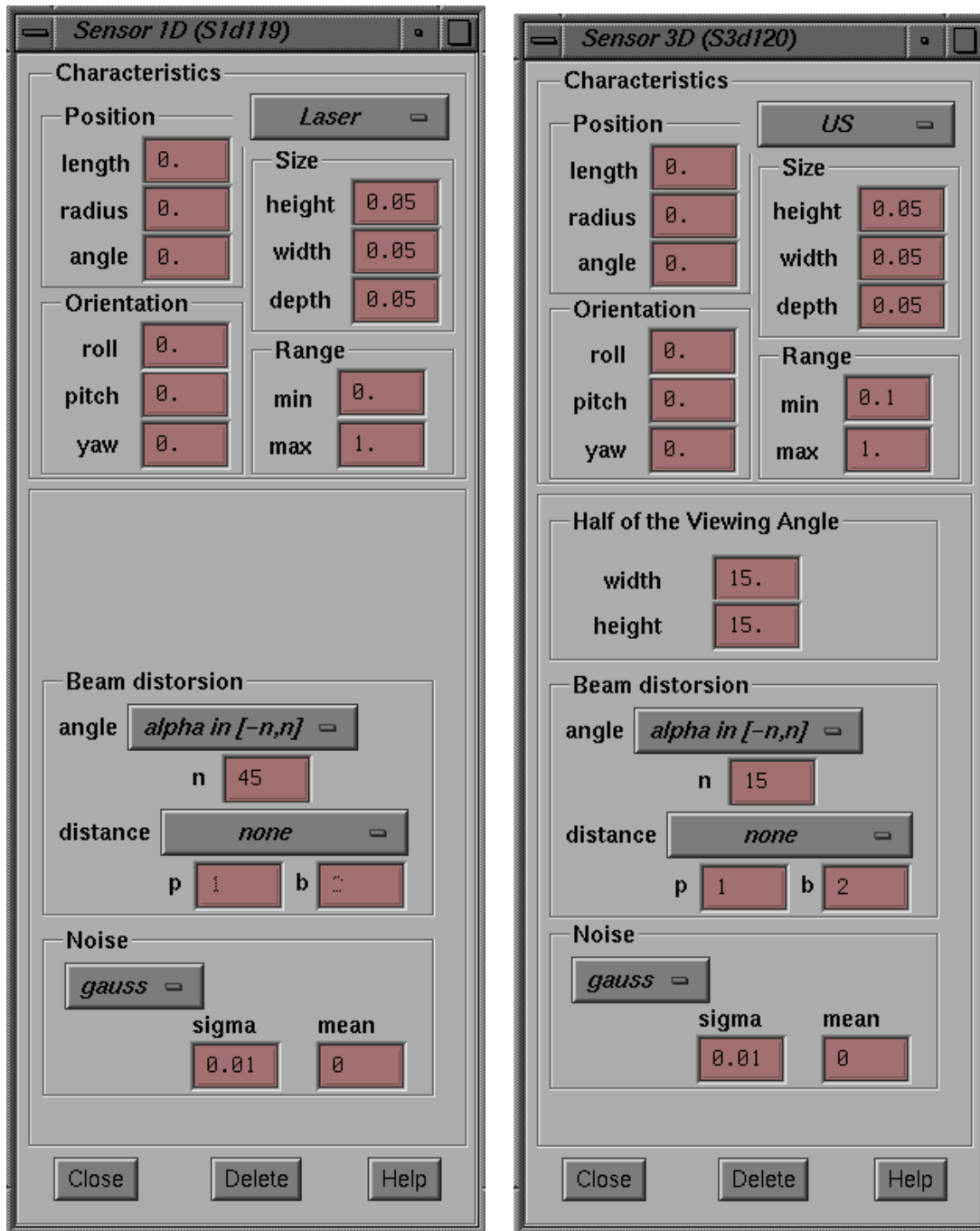
<i>taille :</i>	0.05 m x 0.05 m x 0.05 m
<i>distance min. :</i>	$d_{min} = 0.$ m
<i>distance max. :</i>	$d_{max} = 2.$ m
<i>influence de la distance :</i>	$d \in [d_{min}, d_{max}]$
<i>influence de l'angle d'incidence :</i>	$\alpha \in [-45^\circ, +45^\circ]$
<i>bruit :</i>	Gaussien $\Gamma \sigma = 0.01, m = 0.$

Une bonne approximation est de dire que la lumière est reçue lorsque l'angle d'incidence est dans l'intervalle  $[-45^\circ, +45^\circ]$ . La distance résultante  $d_r$  est alors calculée par :

```

if ( $d \in [d_{min}, d_{max}]$ ) and ( $\alpha \in [-45^\circ, +45^\circ]$ )
then  $d_r = d.(1 + B)$ 
else  $d_r = d_{max}$ 

```



a) Capteur 1D.

b) Capteur 3D.

FIG. 6.10: Éléments de l'interface, pour définir les capteurs à placer sur le robot.

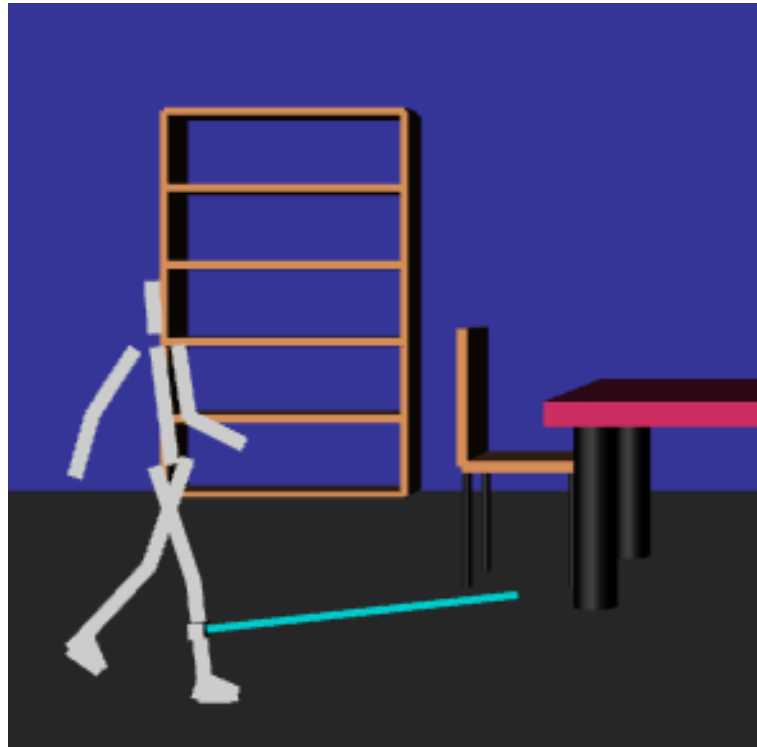
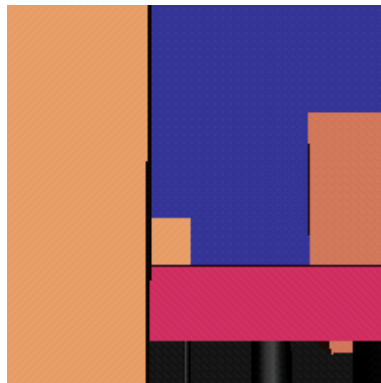
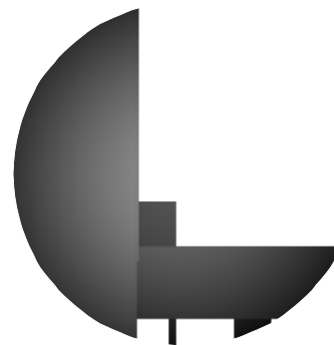


FIG. 6.11: *Visualisation du rayon modélisant l'onde émise par un capteur 1D.*



a) Visualisation directe.



b) Visualisation des profondeurs en niveaux de gris, tronquée par le lobe d'émission.

FIG. 6.12: *Visualisation de l'image de la scène vue par un capteur 3D.*

**Capteur à ultra-sons à hautes fréquences**

C'est un capteur de type 1D avec les paramètres suivants :

<i>taille :</i>	0.05 m x 0.05 m x 0.05 m
<i>distance min. :</i>	$d_{min} = 0.1$ m
<i>distance max. :</i>	$d_{max} = 1.$ m
<i>influence de la distance :</i>	$d \in [d_{min}, d_{max}]$
<i>influence de l'angle d'incidence :</i>	$\alpha \in [-15^\circ, +15^\circ]$
<i>bruit :</i>	Gaussien $\Gamma\sigma = 0.01, m = 0.$

Une bonne approximation est de dire que l'onde ultra-sonore réfléchiée de manière spéculaire est reçue lorsque l'angle d'incidence est dans l'intervalle  $[-15^\circ, +15^\circ]$ . La distance résultante  $d_r$  est alors calculée par :

```

if ( $d \in [d_{min}, d_{max}]$ ) and ( $\alpha \in [-15^\circ, +15^\circ]$ )
then  $d_r = d.(1 + B)$ 
else  $d_r = d_{max}$ 

```

**Capteur de réflectance (diode infra-rouge ou laser)**

C'est un capteur de type 1D avec les paramètres suivants :

<i>taille :</i>	0.05 m x 0.05 m x 0.05 m
<i>distance min. :</i>	$d_{min} = 0.$ m
<i>distance max. :</i>	$d_{max} = 1.$ m
<i>influence de la distance :</i>	$d \in [d_{min}, d_{max}]$
<i>influence de l'angle d'incidence :</i>	$\alpha \in [-90^\circ, +90^\circ]$
<i>bruit :</i>	Gaussien $\Gamma\sigma = 0.01, m = 0.$

La distance réelle est fonction du signal :

$$d_r = f^{-1}(s)$$

Tandis que les capteurs précédents donnent directement la distance réelle les capteurs de réflectance fournissent l'intensité de la lumière diffuse. Cette intensité est fonction de la distance  $d$  de l'angle d'incidence  $\alpha$  et du bruit  $B$  :

$$s = f(d, \alpha, B)$$

Nous utilisons l'équation suivante pour simuler le comportement du signal :

$$s = \frac{(d_{max} - d)}{d^2} \cdot \cos^2 \alpha \cdot (1 + B) \quad (6.5)$$

L'équation (6.5) indique que le signal est inversement proportionnel à la distance et tend vers zéro lorsque la distance atteint la distance maximale. Au-dessus de cette limite

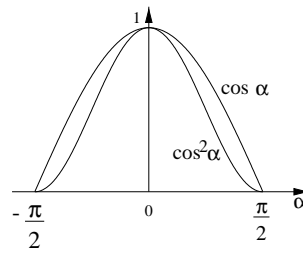


FIG. 6.13: Le signal du capteur de réflectance est fonction de l'angle d'incidence.

signal donne pour valeur zéro (cf. Figure 6.6). Le signal est aussi fonction de l'angle d'incidence : plus cet angle est petit plus l'intensité lumineuse est grande. Nous choisissons de modéliser cet aspect en considérant le signal comme fonction du carré du cosinus de l'angle d'incidence (cf. Figure 6.13).

Lorsqu'un signal est reçu par un tel capteur de par l'équation (6.5) nous ne pouvons déduire directement la distance détectée puisque dans la réalité aucune information sur l'angle d'incidence et sur le bruit n'est fournie. Nous faisons donc l'approximation que  $\alpha$  et  $B$  sont nuls de manière à obtenir quand même une valeur approximative de la distance détectée  $d$  à partir du signal modélisé  $s$ . De l'équation (6.5) nous obtenons un polynôme de degré 2 en  $d$  :

$$s.d^2 + d - d_{max} = 0$$

Les deux racines de ce polynôme sont :

$$f^{-1}(s) = \frac{-1 \pm \sqrt{1 + 4.s.d_{max}}}{2s} \quad (6.6)$$

La seconde racine est toujours négative nous prenons donc la première racine qui est toujours positive pour calculer la distance approximée  $d_r$  :

$$\begin{aligned} &\mathbf{if} (d \in [d_{min}, d_{max}]) \mathbf{and} (\alpha \in [-90^\circ, +90^\circ]) \\ &\mathbf{then} s = \frac{(d_{max} - d)}{d^2} \cdot \cos^2 \alpha \cdot (1 + B) \\ &\quad d_r = \frac{-1 + \sqrt{1 + 4.s.d_{max}}}{2.s} \\ &\mathbf{else} d_r = d_{max} \end{aligned}$$

### Capteur à ultra-sons à basses fréquences

C'est un capteur de type 3D avec les paramètres suivants :

<i>taille :</i>	0.05 m x 0.05 m x 0.05 m
<i>distance min. :</i>	$d_{min} = 0.1$ m
<i>distance max. :</i>	$d_{max} = 1.$ m
<i>influence de la distance :</i>	$d \in [d_{min}, d_{max}]$
<i>influence de l'angle d'incidence :</i>	$\alpha \in [-15^\circ, +15^\circ]$
<i>bruit :</i>	Gaussien $\Gamma \sigma = 0.01, m = 0.$
<i>angle de largeur :</i>	$l = 30^\circ$
<i>angle de hauteur :</i>	$h = 30^\circ$

Une bonne approximation est de dire que l'onde ultra-sonore réfléchi de manière spéculaire est reçue lorsque l'angle d'incidence est dans l'intervalle  $[-15^\circ, +15^\circ]$ . La distance résultante  $d_r$  est alors calculée par :

```

 $d_r = d_{max}$ 
for each pixel  $p$  in the image
  if ( $d(p) \in [d_{min}, d_{max}]$ ) and ( $\alpha(p) \in [-15^\circ, +15^\circ]$ )
    then  $d_r(p) = d(p) \cdot (1 + B)$ 
      if  $d_r(p) < d_r$ 
        then  $d_r = d_r(p)$ 

```

#### 6.2.4 Discussion

Nous avons défini deux types de base de capteurs  $\Gamma$  les capteurs 1D et les capteurs 3D  $\Gamma$  correspondant à deux types d'ondes. Les premières étant très directives  $\Gamma$  nous les avons modélisées par un lancer de rayon. Les secondes étant moins directives et par conséquent plus volumiques que les précédentes  $\Gamma$  nous les avons modélisées à partir du concept de caméra dans *Open Inventor*<sup>TM</sup>. Pour parfaire ces capteurs de base réalisés au moyen d'outils graphiques  $\Gamma$  nous appliquons un post-traitement pour prendre en compte les atténuations du signal causées par la distance et l'angle d'incidence  $\Gamma$  ainsi que le bruit du capteur.

À partir de ces capteurs de base  $\Gamma$  nous avons mis en œuvre différents modèles de capteurs proximétriques  $\Gamma$  tels qu'un télémètre laser  $\Gamma$  un capteur de réflectance  $\Gamma$  des capteurs à ultra-sons hautes fréquences et basses fréquences. Nous pouvons alors placer ces capteurs sur le robot. Ainsi  $\Gamma$  lorsque le robot marche  $\Gamma$  il peut explorer son environnement grâce aux capteurs fixés sur lui. En effet  $\Gamma$  les capteurs délivrent à chaque instant des informations sur l'environnement local du robot. Toutefois  $\Gamma$  ces capteurs proximétriques ne délivrent que des distances aux obstacles. Or  $\Gamma$  ce type de données ne permet pas  $\Gamma$  de manière directe  $\Gamma$  de renseigner sur la nature des obstacles rencontrés. Avec les distances pour seul type d'information sur son environnement  $\Gamma$  l'unique action que le robot peut réaliser est de s'arrêter. Ce comportement est assez limité  $\Gamma$  et nous souhaitons améliorer ses capacités  $\Gamma$  en lui fournissant une connaissance un peu plus conséquente. Nous proposons donc divers outils pour analyser les distances délivrées par les capteurs et répondre à cette attente.

## 6.3 Analyse des données capteurs

Nous souhaitons analyser les données fournies par les capteurs placés sur le robot. Tout d'abord nous supposons que toutes les données des capteurs sont cohérentes entre elles. Cette hypothèse simplificatrice permet d'éviter les inconvénients engendrés par des informations contradictoires tels que la détection du même objet par deux capteurs mais en des endroits différents. En général ce problème est traité lors de la fusion multi-capteurs (cf. Annexe B).

Nous avons vu que les capteurs modélisés jusqu'à présent ne fournissent qu'une information ponctuelle de distance aux objets rencontrés. Nous proposons dans un premier temps de rassembler les distances détectées au cours du mouvement pour en tracer l'évolution. Ensuite nous choisissons de calculer le profil de l'environnement détecté. Il s'agit de déterminer les points touchés dans l'environnement à partir des distances détectées de la géométrie du robot et de la position et l'orientation du capteur sur le robot. Enfin nous proposons un algorithme qui permet de reconnaître certains types d'obstacles ceux qui présentent un intérêt particulier pour notre étude.

### 6.3.1 Courbes de distances

Nous pouvons placer les capteurs modélisés en divers endroits du robot virtuel. Lorsque nous simulons le mouvement de marche du robot à chaque instant de l'animation les capteurs modélisés envoient un nouveau signal correspondant à la nouvelle position du robot dans l'environnement. En effet les capteurs attachés au robot se déplacent avec lui. Par conséquent les signaux capteurs fournissent continuellement les distances aux obstacles les plus proches détectés. Nous pouvons tracer la courbe des distances mesurées par les capteurs en fonction du temps (cf. Figures 6.14 et 6.15).

Toutefois ces courbes que nous appelons *courbes de distances* sont difficilement interprétables de manière directe de par le fait que le mouvement des capteurs dépend de leur position et de leur orientation sur le robot et que les distances dépendent de la nature et de la géométrie des objets rencontrés. Nous avons donc besoin d'extraire des informations pertinentes de ces données brutes de distances.

### 6.3.2 Profil de l'environnement

Une des tâches du robot consiste à détecter les obstacles pouvant gêner son mouvement afin de éviter les collisions. La décision à prendre quant à la réaction du robot est directement liée au type d'obstacle rencontré. En effet la décision est différente selon que le robot détecte un mur un pilier une marche un trou etc. Aussi la simple détection de la distance aux obstacles n'est-elle pas suffisante il faut de surcroît reconnaître le type de l'obstacle détecté. Pour reconnaître les objets dans l'environnement il faut tout d'abord reconstruire cet environnement à partir des informations données par les capteurs.

Les capteurs de type 3D n'offrent pas une information exacte puisque tout un volume est détecté et qu'une seule réponse de distance et non de direction est transmise.



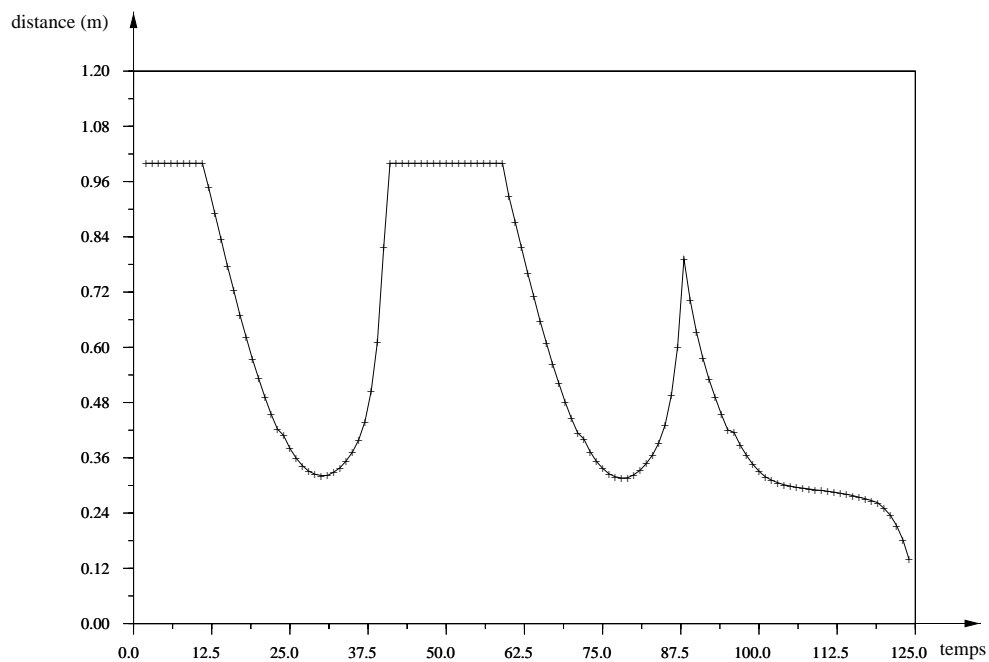
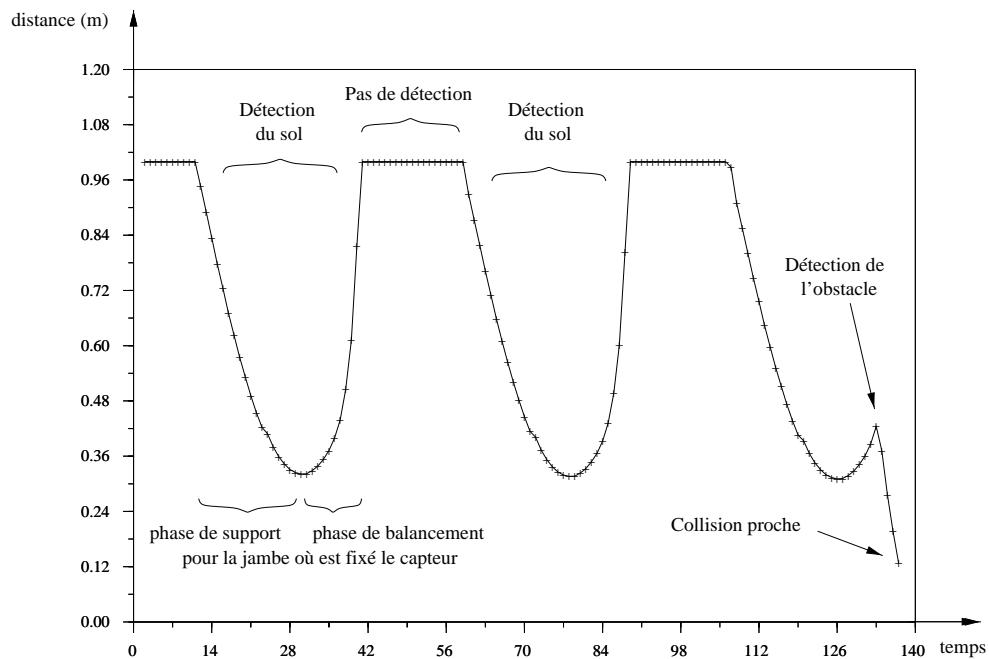
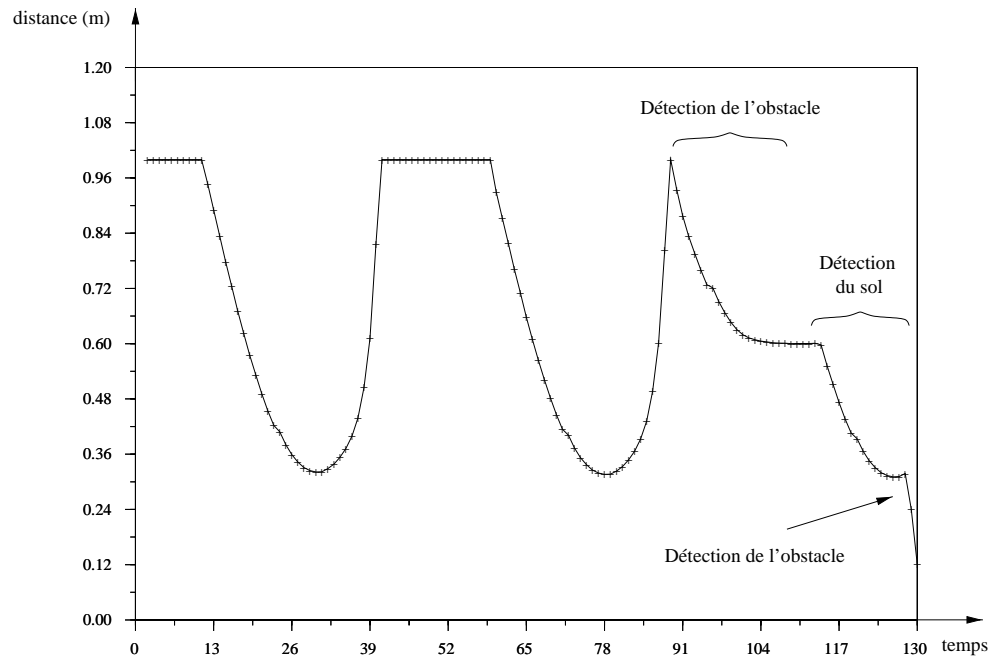
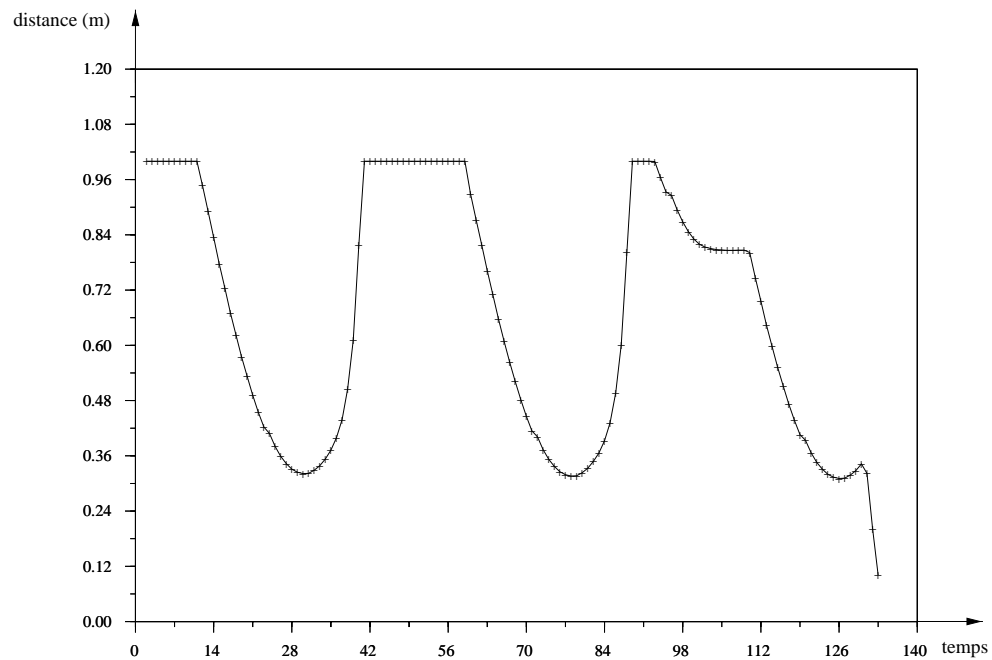


FIG. 6.14: Exemples de courbes de distances idéales : un capteur 1D est placé sur la jambe droite du robot, il est orienté dans la direction de marche, et il a une portée de 1m. Le robot marche, puis le capteur détecte quelque chose (un obstacle posé au sol, de hauteur 30cm).



c) Initialement, le robot est à 3.5m de l'obstacle.



d) Initialement, le robot est à 3.7m de l'obstacle.

FIG. 6.15: *Autres exemples de courbes de distances idéales : un capteur 1D est placé sur la jambe droite du robot, il est orienté dans la direction de marche, et il a une portée de 1m. Le robot marche, puis le capteur détecte quelque chose (un obstacle posé au sol, de hauteur 30cm).*

Ces capteurs sont utilisés en tant que détecteurs de présence plutôt que de détecteurs de distance. Les capteurs de type 1D quant à eux permettent de détecter la distance aux obstacles dans une direction précise. Il est alors possible de les utiliser pour localiser les points touchés par les rayons et ainsi reconstruire le profil de l'environnement détecté.

Puisque le robot ne fait pour l'instant qu'avancer en ligne droite nous choisissons de reconstruire l'environnement de manière simplifiée en calculant seulement le profil 2D de l'environnement détecté par chaque capteur. L'acquisition des données et donc la reconstruction de l'environnement se fait au fur et à mesure de l'avancement du robot. Chacun des capteurs balaye plus ou moins l'espace verticalement selon le mouvement de balancement du segment du robot sur lequel il est fixé ce qui permet de déterminer la hauteur et la longueur des obstacles. Par ailleurs nous avons vu que généralement plusieurs capteurs sont répartis sur le robot. Certains d'entre eux peuvent faire un angle avec la direction de la marche ce qui balaye des "tranches" de l'environnement.

Pour dessiner le profil nous calculons la position et l'orientation du capteur dans le monde à partir de sa position et son orientation relative au robot de la géométrie du robot et du référentiel du robot par rapport au référentiel du monde. Une fois la position du capteur dans le monde et la distance détectée selon l'orientation du capteur dans le monde connues nous pouvons calculer le point 3D touché :

#### Variables connues

posS : position du capteur dans le repère du monde  
 orientS : orientation du capteur dans le repère du monde (quaternion)  
 distance : distance détectée par le rayon  
 dirRl : direction du rayon dans le repère du capteur  
 (dirRl = Vecteur(0π0))

#### Variables à déterminer

dirR : direction du rayon dans le repère du monde  
 posE : position de l'extrémité du rayon du capteur dans le repère du monde

#### Algorithme

dirR = orientS × dirRl ;  
 posE = posS + distance × dirR ;

Si la distance détectée est le seuil maximal du capteur cela signifie que rien n'a été détecté par le capteur. Dans le cas contraire nous ajoutons le point calculé au profil. Nous affichons le profil de l'environnement détecté au fur et à mesure de sa construction. La Figure 6.16 montre un profil calculé à partir des données fournies par un capteur idéal présentées en Figure 6.14 (b).

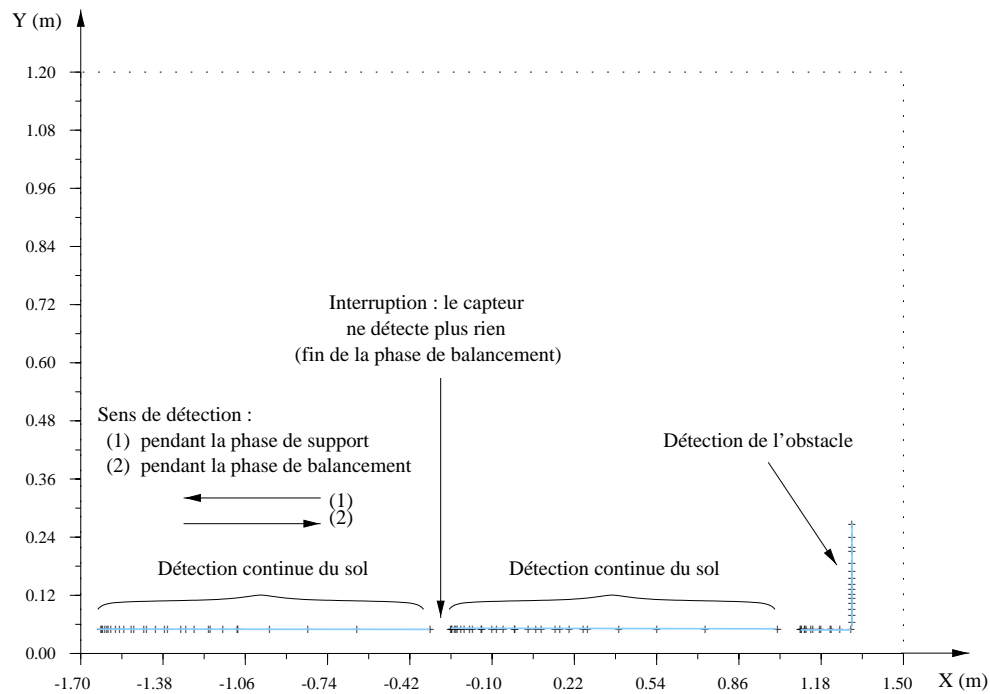


FIG. 6.16: Profil parfait calculé à partir de la courbe de distances de la Figure 6.14 (b).

### 6.3.3 Proposition d'algorithme de reconnaissance

Notre but est de développer un algorithme de reconnaissance d'objets à partir des informations capteurs afin de décider du prochain comportement à faire adopter par le robot. En effet pour pouvoir faire du contrôle de haut niveau les capteurs doivent permettre de reconnaître divers types d'objets dans l'environnement : des escaliers des obstacles à enjamber ou contourner des failles à franchir etc. Ces obstacles peuvent être de types variés il est donc nécessaire de déterminer certaines de leurs caractéristiques. Une opération de classification permet de ranger les données capteur en catégories ou classes. Une classe est une collection de formes ayant les mêmes caractéristiques après avoir réalisé une opération de pré-traitement pour filtrer les données.

Une première idée est de déterminer les courbes de distances obtenues pendant la marche et correspondant à plusieurs types de capteurs puis d'en tirer des résultats statistiques sur lesquels il est nécessaire d'appliquer des tests de vraisemblance pour vérifier qu'on ne fait pas d'erreur qu'on reconnaît bien le type de l'objet. Cependant cette méthode requiert les courbes pour chaque objet différent et chaque capteur différent. Nous avons préféré faire la reconnaissance des objets à partir de la reconstruction du profil réalisée précédemment.

Nous ne souhaitons pas utiliser des algorithmes aussi complexes que ceux développés en vision (cf. Annexe B). En effet actuellement notre reconstruction de l'environnement consiste à déterminer un profil 2D. Nous présentons donc une proposition de reconnaissance simplifiée et plus rapide de cette représentation de l'environnement. Nous choisissons

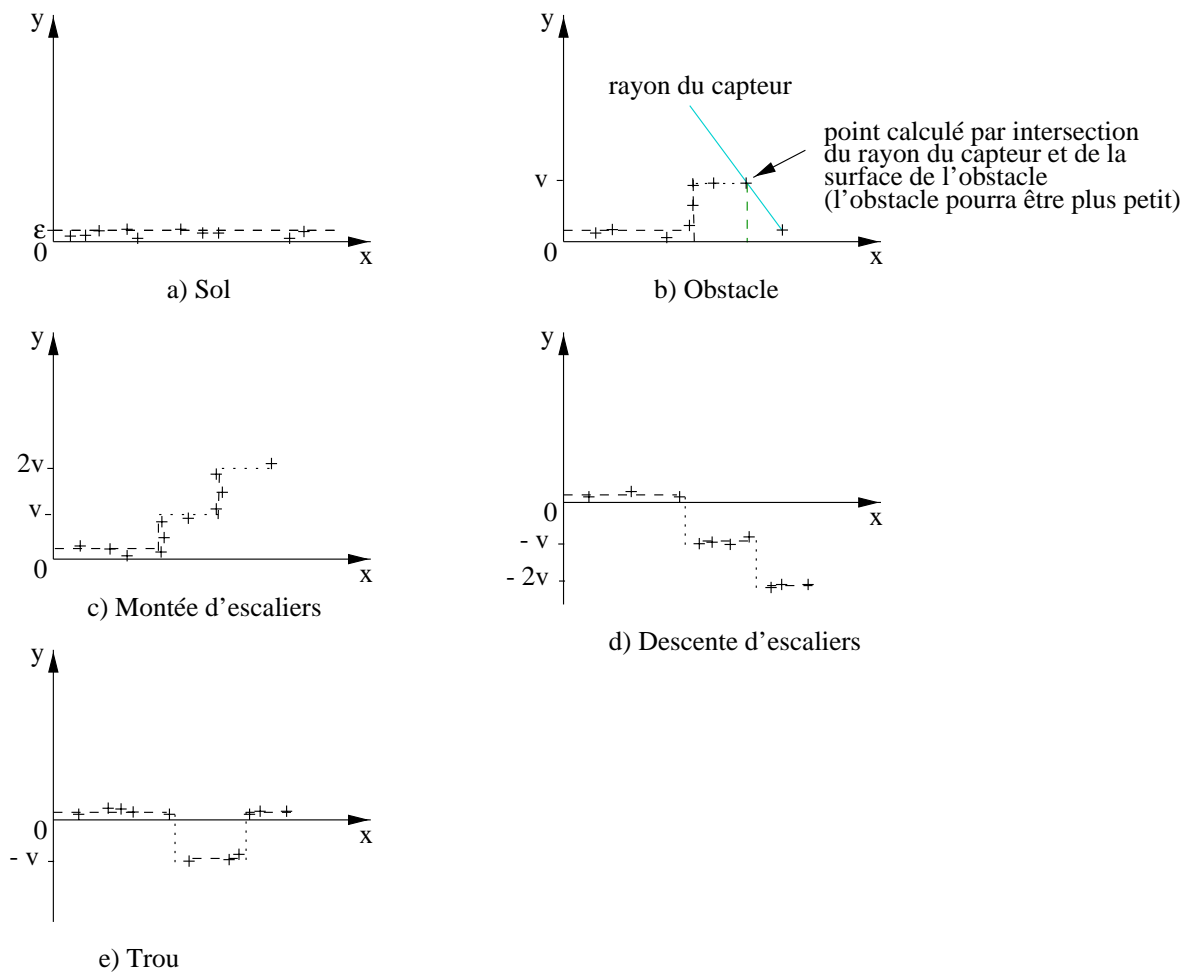
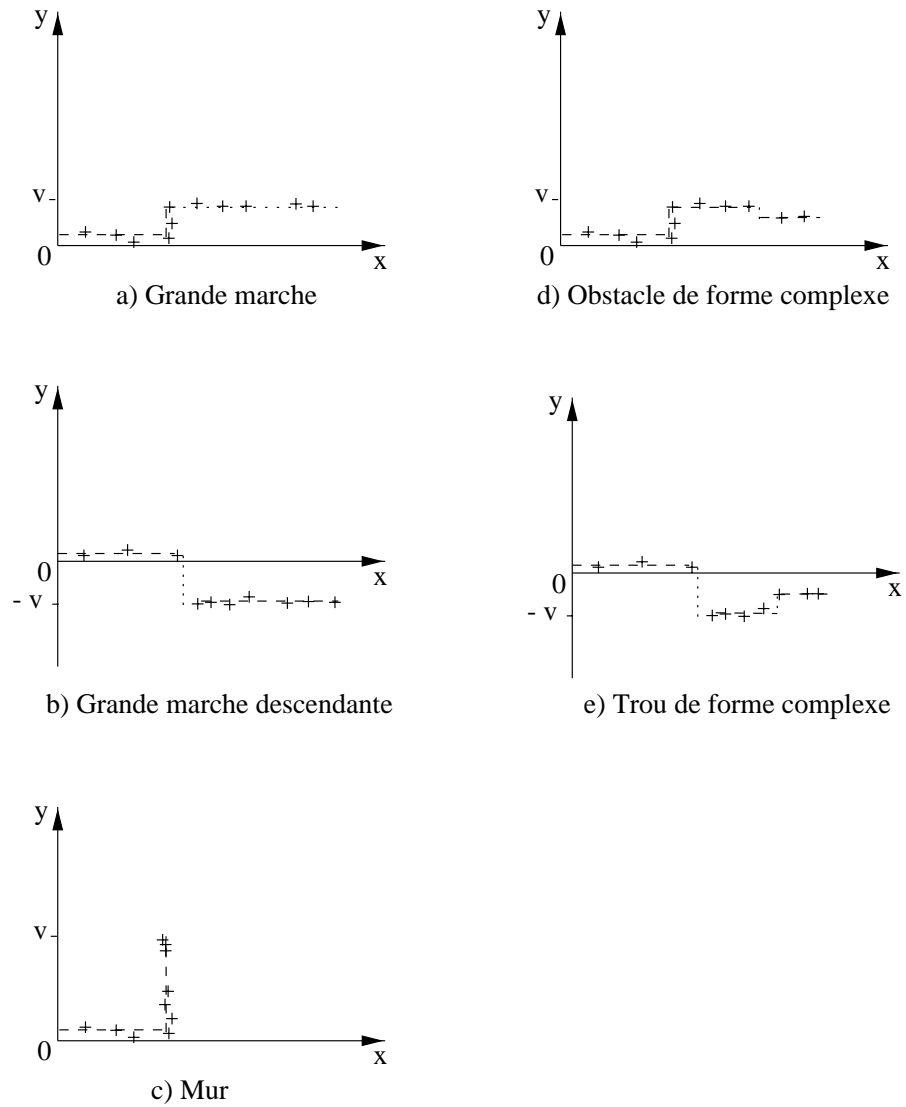


FIG. 6.17: Objets à reconnaître.

de reconnaître cinq types d'objets : le sol plan, les obstacles simples, les marches d'escaliers réguliers montants et celles d'escaliers réguliers descendants, et les trous dans le sol (cf. Figure 6.17). Le reste est considéré comme inconnu.

Nous réalisons cette proposition sous des hypothèses simplificatrices. En effet, nous utilisons uniquement des capteurs 1D, sans que le bruit des capteurs ne soit pris en compte. Si le bruit était considéré, il serait alors nécessaire de faire appel à des outils spécialisés comme par exemple les chaînes de Markov (automates d'états finis avec probabilités sur les transitions) qui permettent de tenir compte de la probabilité que l'objet rencontré soit d'un type particulier donné [DLV96]. Une autre méthode consiste à effectuer un lissage ou filtrage des données pendant la reconstruction de l'environnement [LH91, BPZ96].

Dans notre étude, tous les objets sont détectés, sans délai. De plus, les informations fournies par plusieurs capteurs 1D sont cohérentes, éliminant les problèmes habituels relatifs à la fusion multi-capteurs.

FIG. 6.18: *Objets non reconnus.*

### a) Reconstruction par segments

Tout d'abord à partir des points calculés pour le profil d'environnement l'environnement doit être reconstruit par segments au fur et à mesure de l'évolution du robot. Ainsi un segment permet de déterminer le sol un autre le profil d'une face d'un obstacle etc. Une liste de segments doit alors être maintenue les segments étant ordonnés selon leur abscisse puis leur ordonnée en insérant les nouveaux segments détectés devant le robot et en supprimant ceux qui se retrouvent derrière au fur et à mesure de la marche. Cette liste permet à tout moment de savoir quels segments sont devant le robot.

### b) Reconnaissance d'expressions régulières

Des symboles sont ensuite associés aux segments permettant de distinguer les divers types de segments :

- $h$  : segment horizontal
- $v$  : segment vertical
- $m$  : segment oblique montant
- $d$  : segment oblique descendant.

Puis une expression régulière doit être définie pour chaque type d'objets à reconnaître dans l'environnement. Une reconnaissance d'expressions régulières est alors réalisée au fur et à mesure basée sur un automate d'états fini afin de reconnaître les objets détectés dans l'environnement.

Par exemple dans le cas d'un obstacle (cf. Figure 6.19 (a)) l'expression régulière est :

$$E(\text{obstacle}) = h.v.h.v.h$$

Cette première expression régulière est valable dans le cas de détection idéale où tous les segments sont détectés de leur origine à leur extrémité. Cependant il est plus probable d'obtenir comme profils d'obstacles détectés les profils de la Figure 6.19 de (b) à (e). Cette expression est alors insuffisante pour reconnaître tous les obstacles susceptibles d'être rencontrés. Le vocabulaire des expressions régulières doit être complété par les symboles  $V_M$  et  $V_D$  afin de généraliser les formes des objets potentiellement détectables :

$$\begin{aligned} V_M &= m + m.v + v.m + m.v.m \\ V_D &= d + d.v + v.d + d.v.d \end{aligned}$$

L'expression régulière associée à un obstacle devient alors :

$$E(\text{obstacle}) = h.(V_M + v).h.(V_D + v).h$$

Toutefois dans l'absolu il se peut que cette expression régulière soit reconnue alors que l'objet détecté n'est pas un obstacle posé au sol. Par exemple il est possible de reconnaître un objet bizarre comme celui donné en Figure 6.18 (d). Cependant nous avons décidé par hypothèse d'écarter ce genre de cas de la liste des types d'objets reconnaissables.

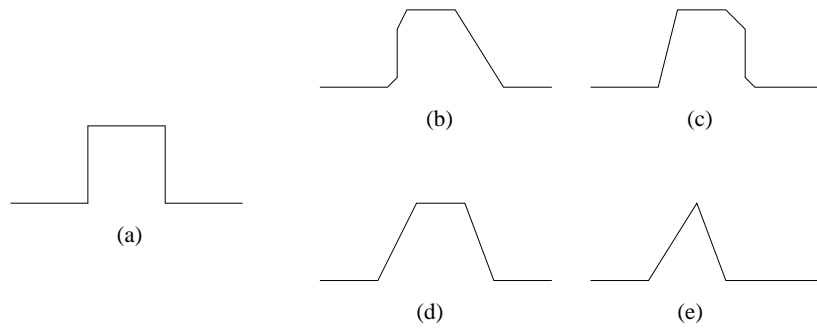


FIG. 6.19: Quelques exemples de profils possibles d'obstacles détectés : (a) Profil idéal. (b) à (e) Profils mal reconstruits.

### c) Expressions paramétrées

Des valeurs des coordonnées doivent être ajoutées aux symboles pour situer plus précisément certains segments  $\Gamma$  afin de faciliter la reconnaissance ultérieure des objets :

- les abscisses  $x$  pour les segments verticaux  $v\Gamma V_M$  et  $V_D\Gamma$
- les ordonnées  $y$  pour les segments horizontaux  $h$ .

Désormais nous ne considérons plus des expressions régulières car l'alphabet que nous utilisons est infini (par exemple  $\Gamma V_M(1) \neq V_M(2)\Gamma$  etc.). Afin de déterminer les expressions paramétrées pour les types d'objets à reconnaître nous pouvons utiliser les opérateurs définis pour les expressions régulières. L'expression suivante est enfin obtenue pour déterminer la présence d'un obstacle au sol :

$$E(\text{obstacle}) = h(0).(V_M(x) + v(x)).h(y).(V_D(x') + v(x')).h(0) \quad \text{avec} \quad \begin{cases} x' > x \\ y > 0 \end{cases}$$

Nous donnons dans la Table 6.2 les expressions associées aux cinq types d'objets que nous souhaitons reconnaître. Cet ensemble d'expressions peut facilement être complété afin de prendre en compte d'autres types d'objets (cf. Figure 6.18). Par exemple pour un objet de type :

- grande marche (cf. Figure 6.18 (a)) :  $E(\text{marche}) = h(y).(V_M + v(x)).h(y')$  avec  $y' > y$
- mur (cf. Figure 6.18 (c)) :  $E(\text{mur}) = h(y).V_M.v(x)$

### d) Reconnaissance des expressions paramétrées

Finalement l'algorithme de reconnaissance d'objets consiste à prendre la liste de symboles associée à la liste de segments présents devant le robot et à reconnaître les objets qu'ils représentent. Cet algorithme peut être représenté par l'automate reconnaisseur de la Figure 6.20 et permet de reconnaître les cinq types d'objets désirés. Par exemple concernant la détection d'un obstacle l'algorithme reconnaît le sol au début puis un mur d'une certaine hauteur puis une marche et enfin un obstacle (cf. Figure 6.21).



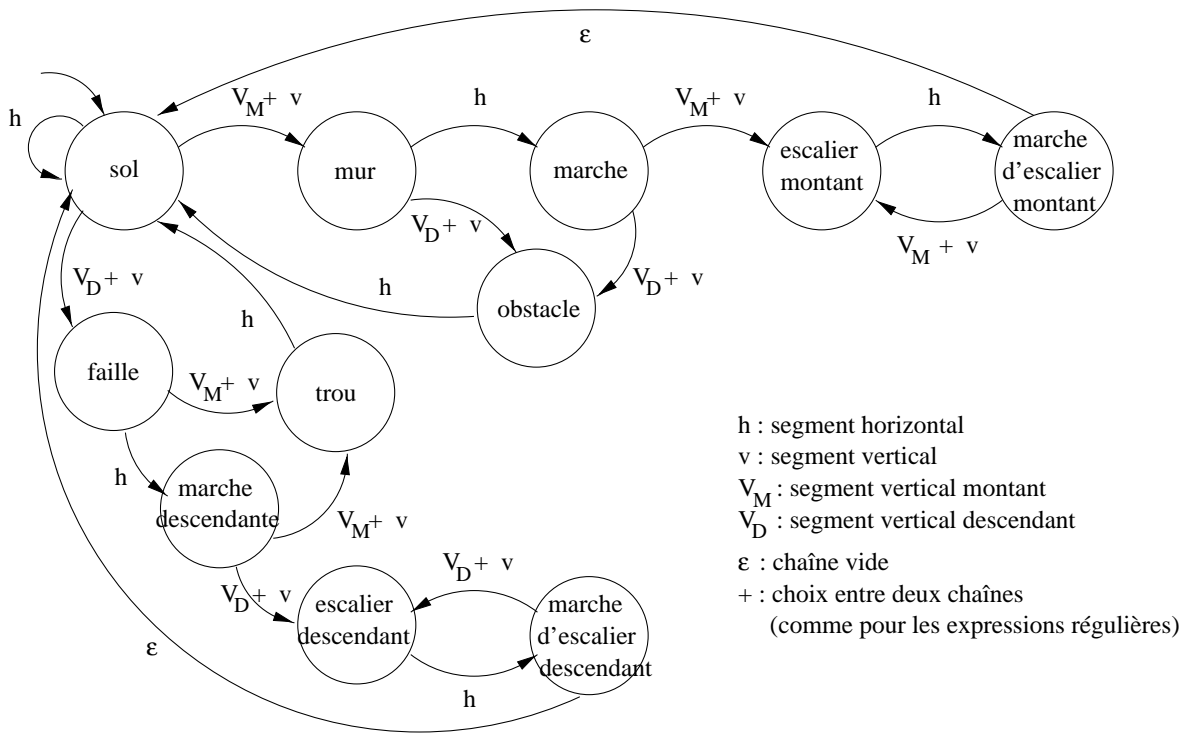


FIG. 6.20: Automate reconnaisseur des cinq types d'objets désirés.

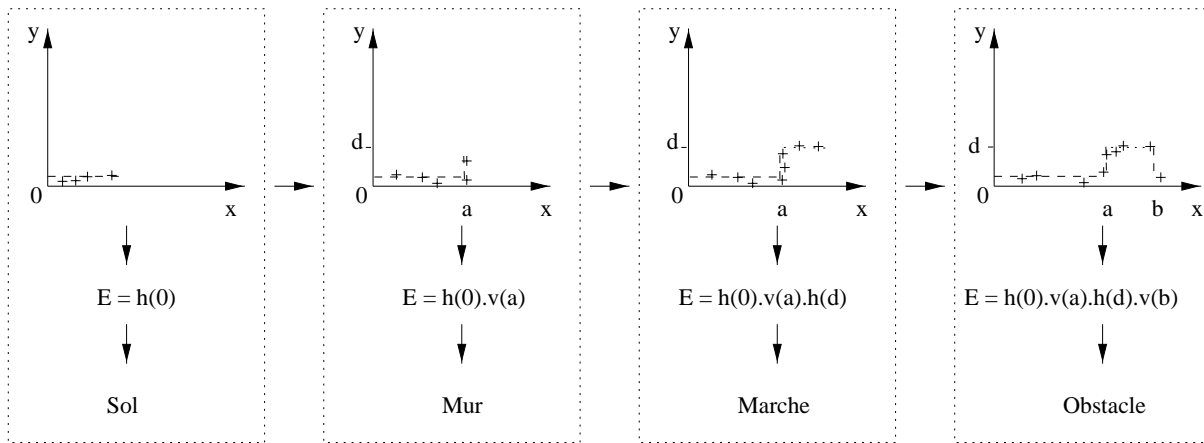


FIG. 6.21: Reconnaissance au fur et à mesure d'un obstacle.

$$\begin{aligned}
E(\text{sol}) &= h(0) \\
E(\text{obstacle}) &= h(0) \cdot (V_M(x) + v(x)) \cdot [h(y)] \cdot (V_D(x') + v(x')) \cdot h(0) \text{ avec } \begin{cases} x' > x \\ y > 0 \end{cases} \\
E(\text{montée d'escaliers}) &= h(y_0) \cdot (V_M(x_0) + v(x_0)) \cdot \left( h(y_i) \cdot (V_M(x_i) + v(x_i)) \right)^+ \cdot [h(y_{max})] \\
&\text{avec } \begin{cases} y_i - y_{i-1} = \Delta y > 0 \\ x_i - x_{i-1} = \Delta x > 0 \end{cases} \\
E(\text{trou}) &= h(0) \cdot (V_D(x) + v(x)) \cdot [h(y)] \cdot (V_M(x') + v(x')) \cdot h(0) \text{ avec } \begin{cases} x' > x \\ y < 0 \end{cases} \\
E(\text{descente d'escaliers}) &= h(y_0) \cdot (V_D(x_0) + v(x_0)) \cdot \left( h(y_i) \cdot (V_D(x_i) + v(x_i)) \right)^+ \cdot [h(y_{max})] \\
&\text{avec } \begin{cases} y_{i-1} - y_i = \Delta y > 0 \\ x_i - x_{i-1} = \Delta x > 0 \end{cases}
\end{aligned}$$

TAB. 6.2: Expressions paramétrées associées aux types d'objets à reconnaître.

Les dimensions des objets reconnus peuvent être facilement obtenues en récupérant :

- pour chaque segment horizontal : ses attributs  $(y, x_i, x_f)$  où  $y$  est l'ordonnée du segment,  $x_i$  l'abscisse de son origine et  $x_f$  l'abscisse de son extrémité
- pour chaque segment vertical : ses attributs  $(x, y_i, y_f)$  où  $x$  est l'abscisse du segment,  $y_i$  l'ordonnée de son origine et  $y_f$  l'ordonnée de son extrémité.

Nous pouvons également connaître la distance du robot à cet obstacle à partir de ces données et de la position du robot.

Nous avons ainsi proposé un algorithme qui non seulement reconnaît le type des objets à partir du profil de l'environnement mais peut également fournir les dimensions de ces objets. Cet algorithme peut être représenté par un automate qui reconnaît les objets selon leur expression paramétrée à partir des segments reconstruits pour les objets détectés. Toutefois cette proposition peut soulever des questions qui à ce stade ne peuvent avoir de réponses. En particulier dans le cas où le bruit des capteurs est considéré une étude semble être nécessaire afin de déterminer les seuils pour distinguer par exemple un segment vertical d'un segment oblique étant donné le modèle de bruit choisi pour les données des capteurs.

#### e) Cas particulier des grands trous

Dans le cas d'un trou de dimensions supérieures à la portée des capteurs 1D placés sur les jambes (cf. Figure 6.22) aucun point de l'obstacle n'apparaît sur le profil de l'environnement (cf. Figure 6.23). Le point le plus éloigné du robot à être détecté est le

bord du trou. Au niveau de la courbe de distances (cf. Figure 6.24) nous remarquons une perturbation du cycle normal de détection du sol. Le seul segment qui est reconstruit est celui du sol et donc l'algorithme de reconnaissance d'expressions paramétrées reste dans l'état initial "sol". Toutefois à partir des dimensions du segment reconstruit  $(y, x_i, x_f)$  il est possible de déterminer que le robot doit s'arrêter. Ce choix de faire arrêter le robot relève du contrôle de haut niveau que nous traitons au paragraphe 7.3.1.

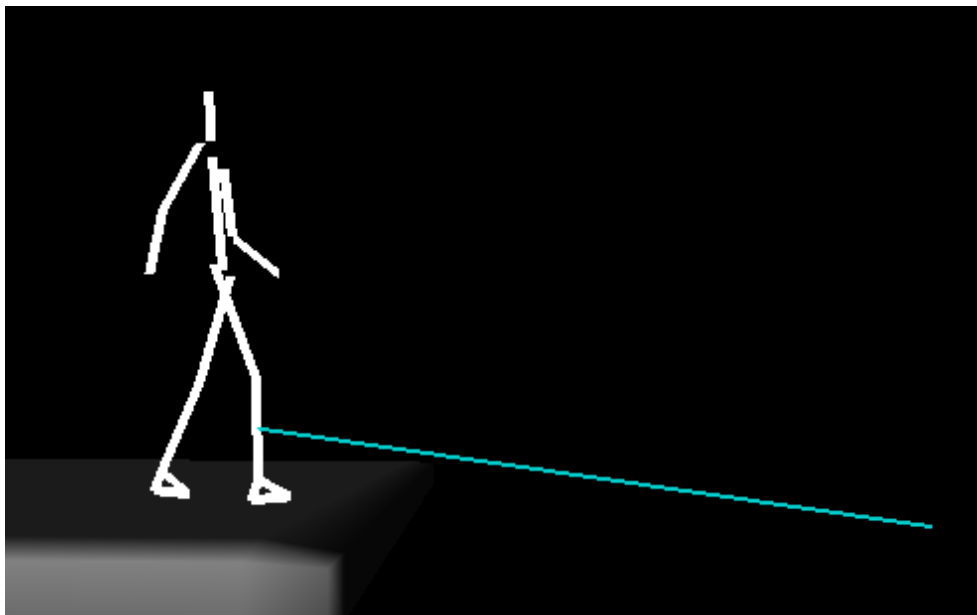


FIG. 6.22: Le robot en présence d'un grand trou.

## 6.4 Validation

Des tests avec des capteurs réels ont été réalisés par Derradji Meharzi [Meh99]. Le premier capteur testé est de type optique KM1/AKM Matsushita (cf. Figure 6.25) le second de type ultra-sons à hautes fréquences SENSOPART (cf. Figure 6.26). Ces deux capteurs ont été fixés l'un après l'autre sur une orthèse de genou (cf. Figure 6.28). Les expérimentations ont consisté à équiper le sujet de cette orthèse et à le faire marcher en ligne droite sur sol plat et dans un environnement non perturbateur pour les capteurs testés.

Les capteurs de par le dispositif de l'orthèse imposé sont positionnés approximativement au milieu du tibia à une distance d'environ 25cm du genou de 10cm environ devant le tibia et orienté d'un angle de 20° environ avec l'horizontale. Une telle configuration permet de détecter le sol pendant la marche de manière quasi continue. Les enregistrements des sorties capteurs donnant les distances détectées sont montrés en Figure 6.29 (a) Figure 6.30 (a) et Figure 6.31 (a). Nous avons essayé de reproduire au mieux les conditions précédentes de positionnement du capteur sur notre robot virtuel ainsi que de la zone de détection de façon à tester les capteurs modélisés.

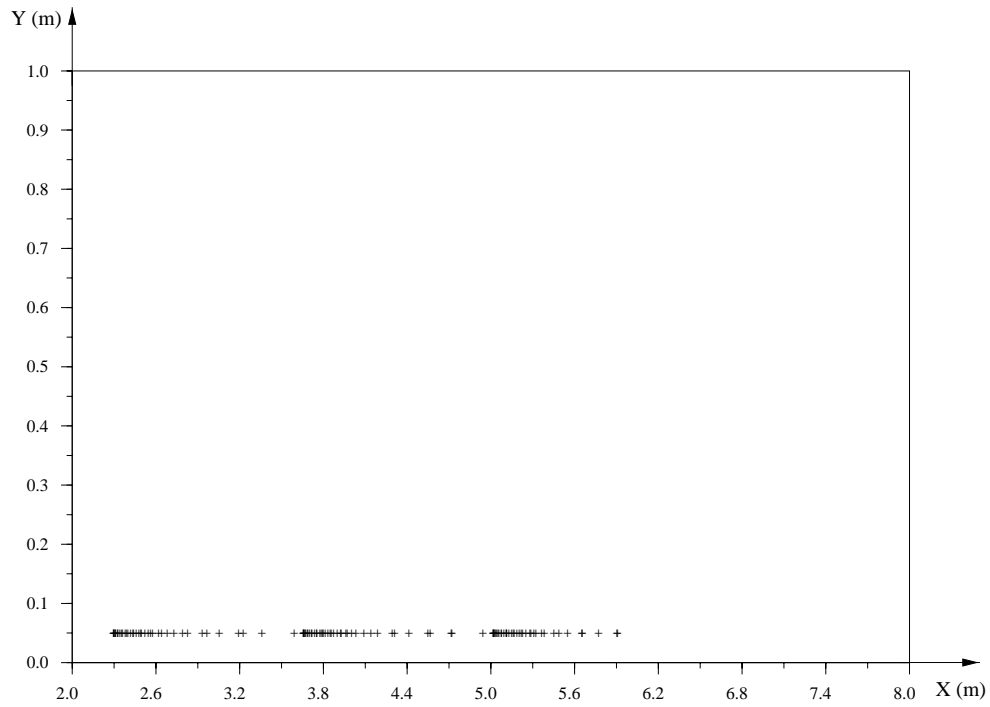


FIG. 6.23: Profil détecté lors de la simulation : aucun point ne permet de prévenir de la présence du grand trou.

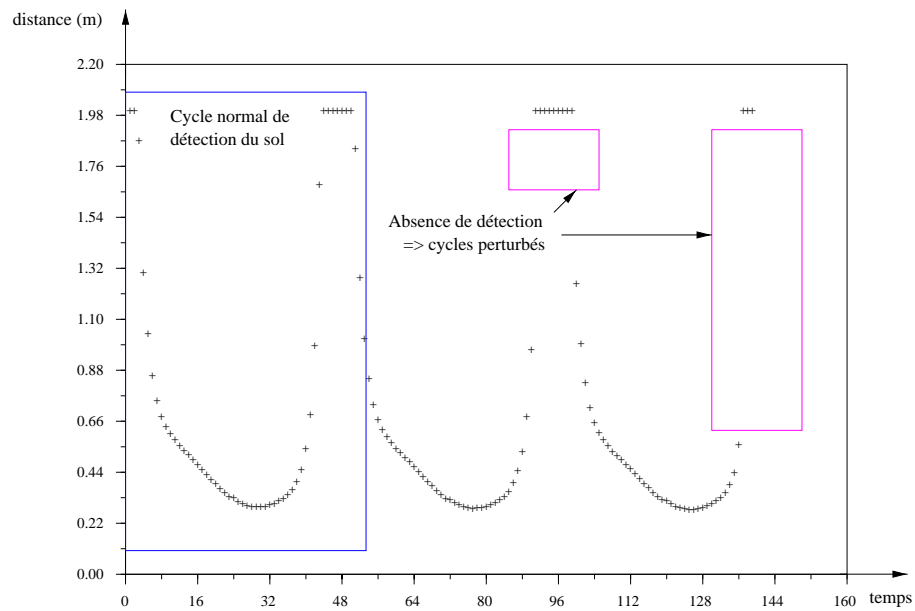


FIG. 6.24: Courbe de distances lorsque le robot est face à un grand trou : les derniers cycles de distances trouvées sont incomplets, ce qui indique la présence d'un trou.



**Capteur photoélectrique analogique  
KM1/AKM (Matsushita)**

Méthode de détection par triangulation  
optique d'aire réfléchive, avec une source  
lumineuse infra-rouge LED.

Zone de détection : 20 à 70cm.

Voltage : 24 VDC.

Temps de réponse : 5 millisecondes.

Fréquence : 200Hz.

FIG. 6.25: Capteur optique KM1/AKM (Matsushita).



**Détecteur à ultra-sons SENSOPART**

Zone de détection : 35 à 300cm (cf. Fi-  
gure 6.27).

Fréquence d'émission : 120kHz.

Résolution : 1mm.

Reproductibilité :  $\pm 3mm$ .

FIG. 6.26: Capteur à ultra-sons SENSOPART.

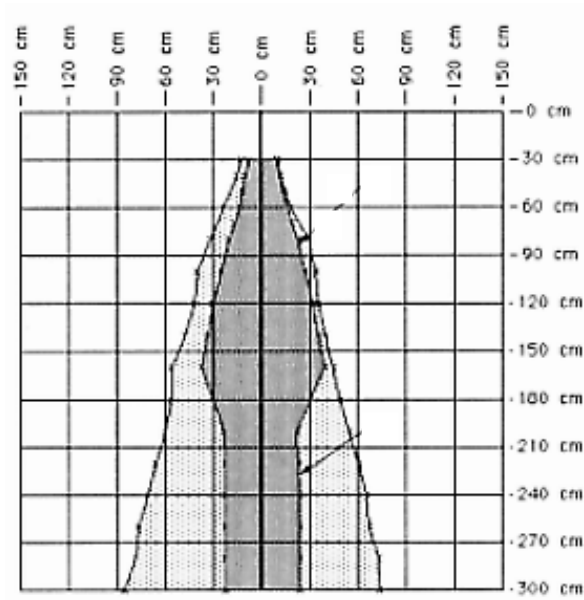


FIG. 6.27: *Cône de détection du capteur à ultra-sons réel SENSOPART.*



FIG. 6.28: *Orthèse du genou utilisée pour les tests.*

### 6.4.1 Capteur optique

Notre modèle de capteur laser s'est avéré moins performant que le capteur réel. En effet, quel que soit l'angle d'incidence du rayon, le capteur réel détecte toujours le sol. Nous avons donc employé un modèle de capteur plus simple que le capteur 1D, en considérant la distance et le bruit, mais sans prendre en compte l'angle d'incidence du rayon. Nous appelons ce capteur "capteur SAI" (sans angle d'incidence). Le résultat alors obtenu s'approche de manière convaincante de la réalité (cf. Figure 6.29 (b)). La différence notable entre les deux courbes de la Figure 6.29 concerne la largeur des pics. Toutefois, cela peut facilement être imputé aux différences entre la marche du sujet et la marche du robot. Il reste à vérifier cette hypothèse dès que le robot réel marchera. Pour cela, il faudra calculer une marche identique pour le robot virtuel et le robot réel, et poser sur le robot virtuel un capteur virtuel au même endroit que le capteur réel sur le robot réel. Ensuite, il s'agira de comparer les différents résultats obtenus qui devraient, au vu des résultats actuels, coïncider encore mieux.

### 6.4.2 Capteur à ultra-sons

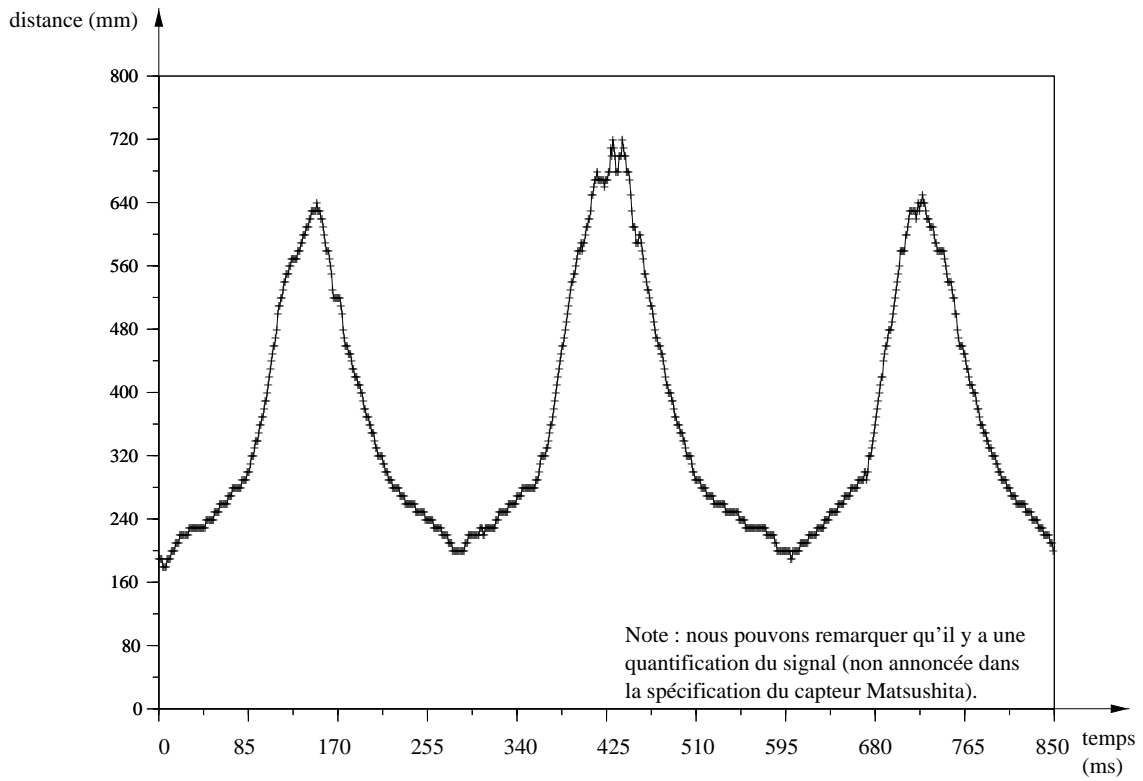
De la même manière que pour le capteur optique, notre modèle de capteur à ultra-sons s'est avéré moins performant que le capteur réel. Nous avons également employé le capteur "SAI" à la place, et le résultat obtenu est assez convaincant, que ce soit pour la simple détection du sol (cf. Figure 6.30 (b)) ou celle de l'obstacle (cf. Figure 6.31 (b)). La différence entre les courbes, imputée à la différence de marche, est plus accentuée.

### 6.4.3 Discussion

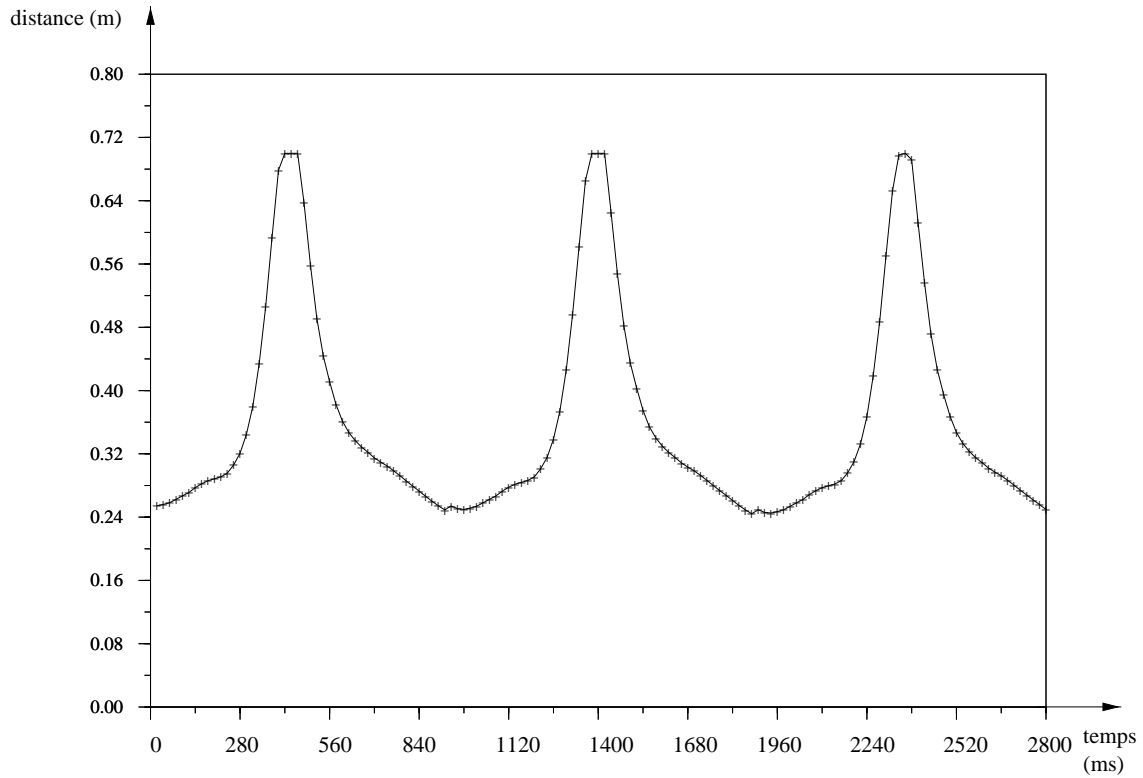
Nous en déduisons que les tests réalisés avec les capteurs réels pour détecter le sol ou des obstacles simples, tendent à infirmer nos hypothèses trop fortes pour modéliser les capteurs. Il apparaît que le capteur "SAI" est suffisant dans ces cas. Cependant, ces expérimentations ont été réalisées dans des conditions assez favorables, au sens où l'environnement n'était pas bruyant. Il peut en être tout autrement dans d'autres environnements. Aussi, des tests plus complets sont nécessaires pour vérifier le comportement des capteurs dans toutes les conditions d'utilisation possibles. En attendant, nous pouvons considérer qu'un bon modèle de capteurs optiques et à ultra-sons est notre modèle de capteur "SAI".

## 6.5 Conclusion

Nous avons présenté les capteurs que nous avons modélisés au moyen d'outils graphiques et de post-traitements analytiques. Il s'agit de capteurs proximétriques de type 1D et de type 3D, modélisant des capteurs réels de type optiques et à ultra-sons. Ces capteurs renvoient la distance aux obstacles détectés, qui doit servir à la reconnaissance de l'environnement par le robot. À cette fin, nous avons développé des outils pour analyser ces distances détectées. Tout d'abord, la courbe de distances permet de rassembler les données. Les distances sont utilisées au fur et à mesure de la simulation pour calculer le profil de l'environnement détecté. Enfin, l'algorithme de reconnaissance, à partir du profil de l'environnement, permet de reconnaître la nature des obstacles rencontrés.



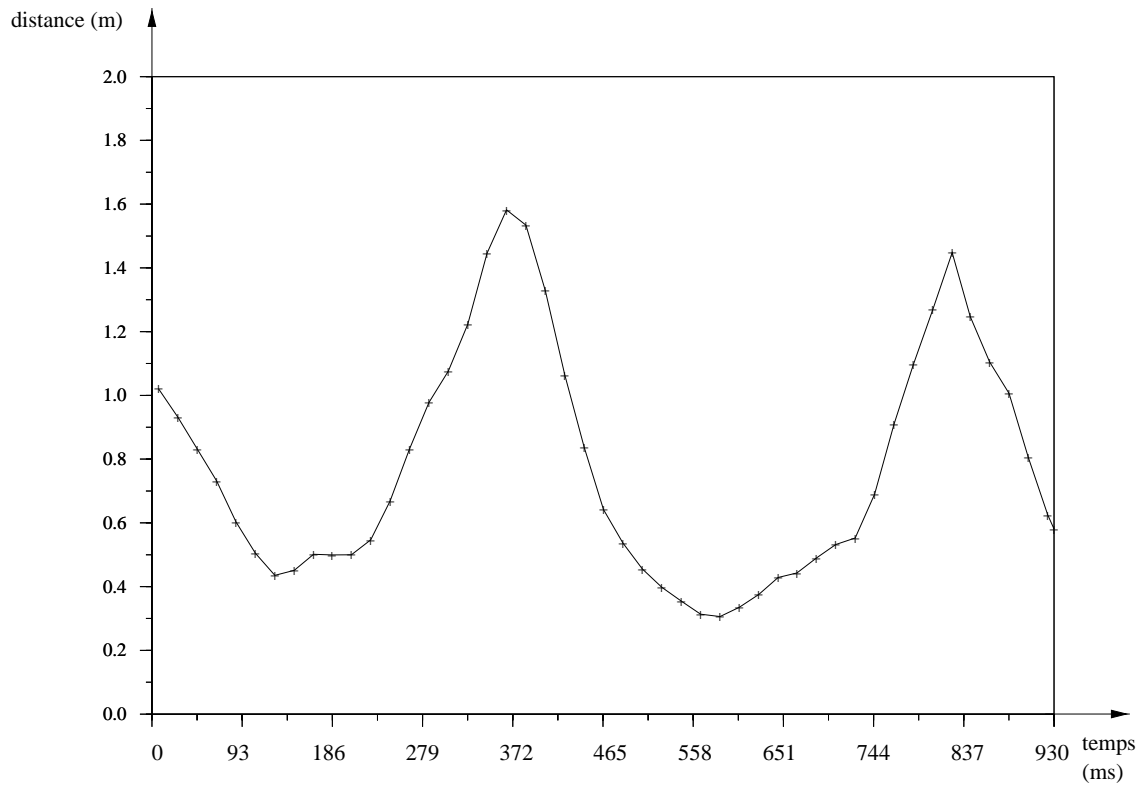
a) Capteur réel sur sujet humain.



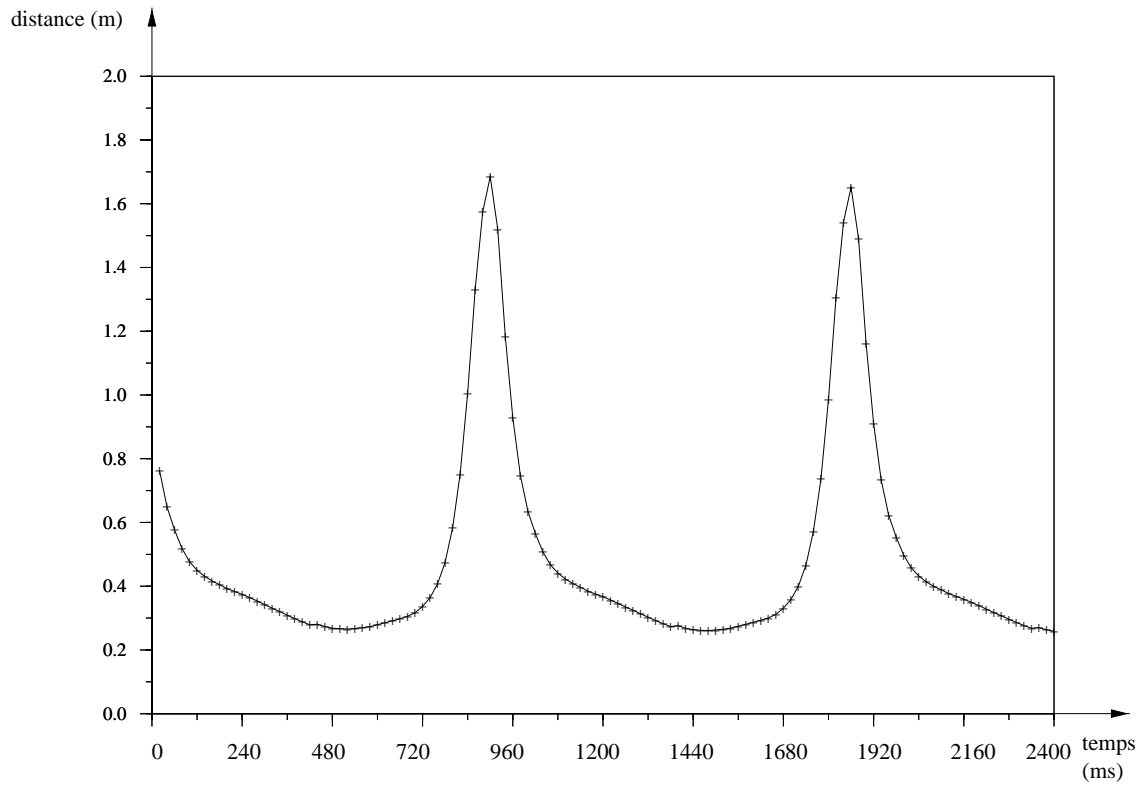
b) Capteur simulé sur robot virtuel.

FIG. 6.29: Enregistrements des sorties capteurs optiques réalisés lors de la détection du sol pendant la marche.



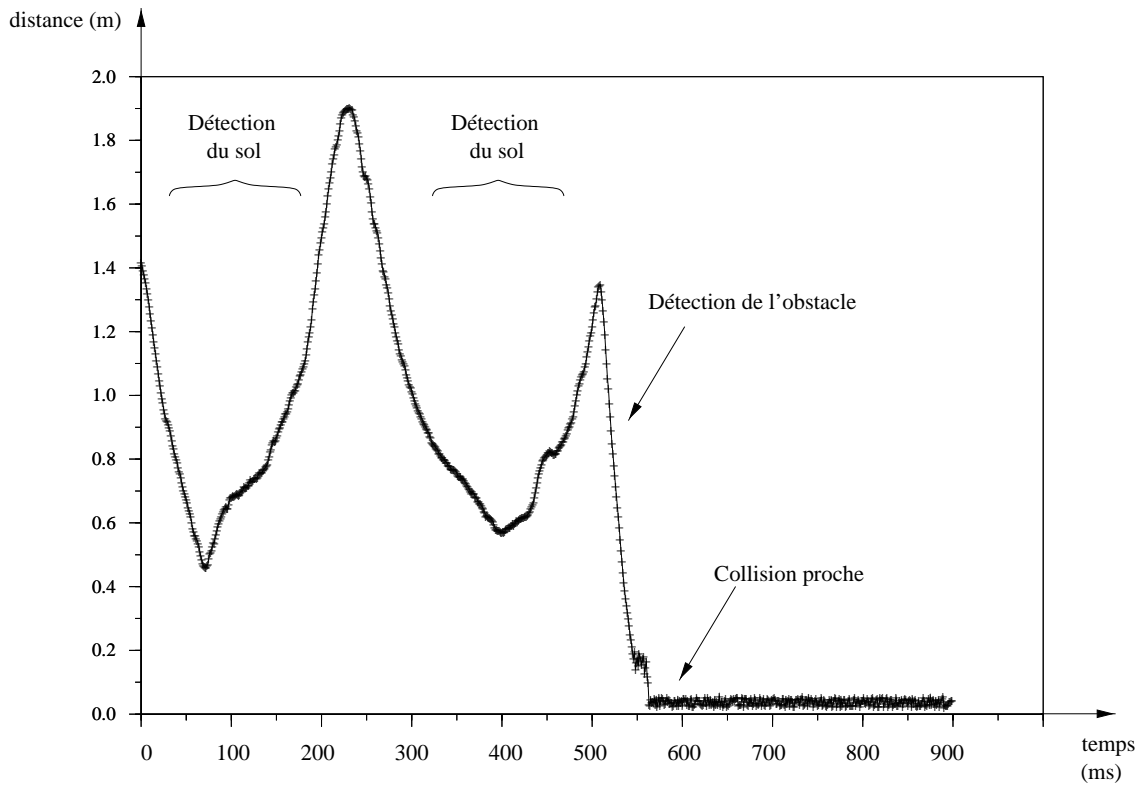


a) Capteur réel sur sujet humain.

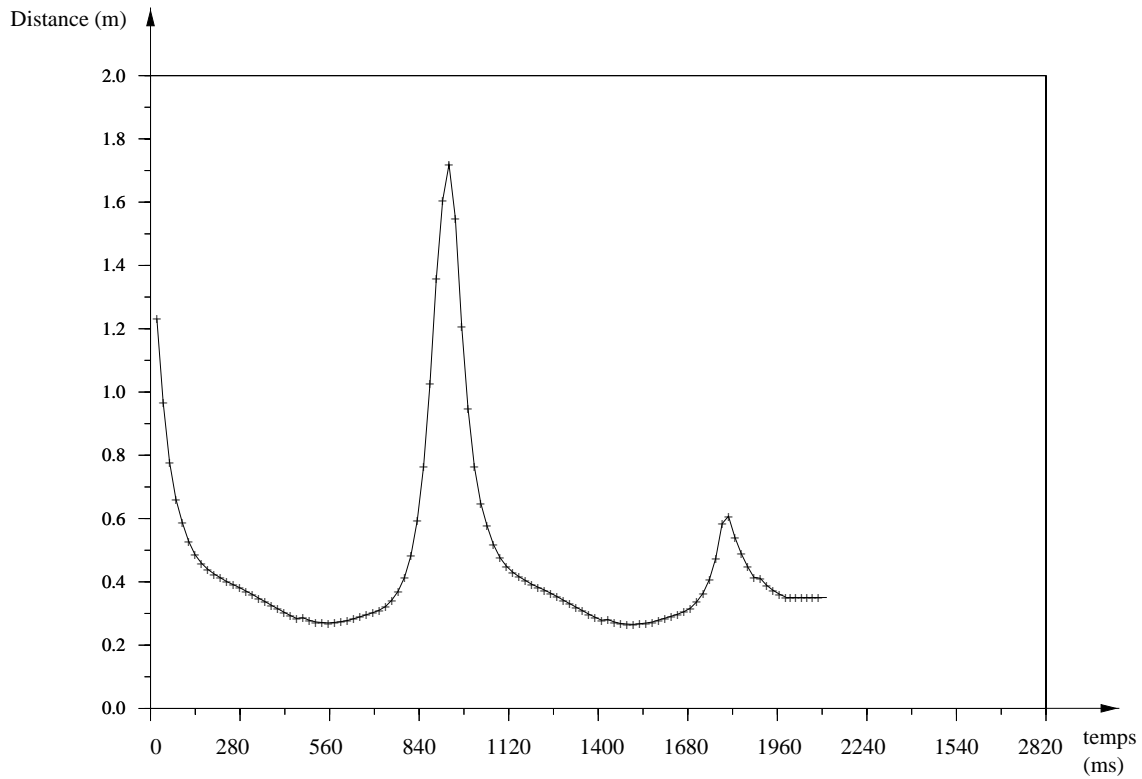


b) Capteur simulé sur robot virtuel.

FIG. 6.30: Enregistrements des sorties capteurs à ultra-sons réalisés lors de la détection du sol pendant la marche.



a) Capteur réel sur sujet humain.



b) Capteur simulé sur robot virtuel.

FIG. 6.31: *Enregistrements des sorties capteurs à ultra-sons réalisés lors de la détection d'un obstacle (carton de 40cm).*

Nous avons testé des capteurs proximétriques réels afin de valider nos modèles de capteurs. L'outil qui nous a permis de comparer les résultats obtenus par les capteurs réels et ceux obtenus par simulation a été la courbe de distances. Il est apparu que les modèles de capteurs réalisés étaient trop restrictifs et que de meilleures correspondances étaient obtenues avec des capteurs "SAI" (sans angle d'incidence). Cependant les tests effectués ne sont en aucun cas exhaustifs et de plus amples expérimentations seraient nécessaires pour valider parfaitement notre approche de modélisation de capteurs.

D'autre part ces modèles de capteurs peuvent être améliorés de plusieurs manières. La première est de changer les hypothèses de bruit pour mieux correspondre au bruit produit par les capteurs réels. En effet au vu des résultats expérimentaux la quantification du signal semble être constante quelque soit l'amplitude du signal et correspond plutôt à une erreur additive que multiplicative. Une deuxième manière consiste à d'ajouter une quantification sur le signal en espace et en temps : en espace pour renvoyer des mesures de distance qui soient des multiples de  $3mm$  par exemple comme c'est le cas pour le capteur SENSOPART et en temps pour prendre en compte la fréquence de mesure ce qui crénèle les valeurs des signaux (cf. Figure 6.29 (a)) pouvant ainsi affecter le contrôleur de mouvement du robot. Une troisième manière concerne la prise en compte les problèmes d'interférence d'ondes liés à l'utilisation conjointe de plusieurs capteurs. Enfin une dernière manière est de considérer la nature photométrique des objets constituant l'environnement pour les diverses longueurs d'ondes utilisées par nos modèles de capteurs.

Une importante amélioration peut être apportée en reconstruisant non pas uniquement le profil mais tout l'environnement dans ses trois dimensions. Nous obtenons ainsi une meilleure représentation de cet environnement. Nous pouvons alors reconnaître une plus grande variété d'objets. Il peut alors s'avérer intéressant d'utiliser des techniques de reconnaissance d'objets de vision plus performantes que notre algorithme partiel de reconnaissance.

Dans le prochain chapitre nous nous intéressons à des applications complètes avec les capteurs. Tout d'abord nous avons vu que le robot doit percevoir son environnement pour qu'il puisse évoluer à l'intérieur en toute sécurité. Par conséquent il faut qu'il détecte tous les objets susceptibles d'entraver et/ou perturber son chemin. Nous pouvons dans cette optique couvrir le robot de capteurs mais ce n'est pas une attitude raisonnable. Nous nous intéressons donc à l'étude du placement de capteurs sur le robot pour déterminer une configuration optimale selon un critère à définir.

Une seconde application est la simulation complète du robot dans son environnement. Le robot est capable de marcher ainsi que de percevoir et reconnaître son environnement. Il nous a semblé intéressant de commencer à expérimenter la réactivité du robot face à son environnement. En particulier nous nous intéressons à la détection d'obstacles pendant la marche pour déclencher un changement dans le comportement du robot par exemple en réalisant le mouvement transitoire d'arrêt ou en adoptant un mode de locomotion plus approprié tel que la montée d'escaliers en présence d'un escalier.

# Chapitre 7

## Applications

### 7.1 Introduction

Nous nous intéressons maintenant aux applications directes des capteurs modélisés ainsi que des outils d'analyse développés dans le chapitre précédent. Nous avons vu que le robot a besoin de la perception et de la reconnaissance de formes. Notre but est d'avoir une connaissance globale de haut niveau des éléments constituant l'environnement à partir des diverses informations locales de bas niveau acquises par plusieurs capteurs placés sur le robot.

Une première application des capteurs est donc d'étudier leur placement sur le robot bipède. En effet nous avons le choix du nombre de capteurs, du type de capteurs, de leur position et de leur orientation sur le robot. Comme les capteurs ont des caractéristiques différentes, notamment le type de détection 1D ou 3D, il nous semble intéressant de varier les types de capteurs à placer sur le robot afin d'unir les qualités de chacun. Toutefois le nombre de ces capteurs doit être limité à cause du coût et du poids additionnels. La difficulté de cette étude est d'ordre méthodologique, puisqu'il s'agit de sélectionner une configuration appropriée de capteurs devant équiper le robot selon l'environnement et les tâches à exécuter.

Une seconde application consiste à valider l'ensemble de ce qui a été réalisé. D'une part nous avons calculé un cycle de marche ainsi que les mouvements transitoires de départ et d'arrêt afin d'obtenir un mouvement de référence complet pour la marche du robot. D'autre part nous avons modélisé des capteurs proximétriques et nous avons développé des outils d'analyse pour les données capteur dans le but de réaliser les tâches de perception et de reconnaissance du robot. Nous souhaitons maintenant connecter ces deux travaux en réalisant une simulation complète du robot dans son environnement.

Nous présentons dans un premier temps l'étude que nous avons réalisée pour déterminer le meilleur placement des capteurs sur le robot selon certains critères et dans un second temps une simulation complète de la marche du robot dans l'environnement et qui considère et traite au niveau du contrôle du mouvement les informations des capteurs.

## 7.2 Détermination du placement des capteurs

Nous souhaitons étudier le placement des capteurs sur le robot afin de déterminer la configuration permettant de détecter le mieux l'environnement. Toutefois la détection de l'environnement dépend de divers paramètres. Comme nous ne souhaitons pas étudier le placement des capteurs dans un cadre général nous définissons des hypothèses de travail pour ces paramètres. Ensuite nous proposons un outil d'analyse des configurations de capteurs sur le robot. Enfin nous présentons plusieurs placements de capteurs que nous analysons selon diverses situations.

### 7.2.1 Hypothèses de travail

La détection de l'environnement dépend de plusieurs paramètres. Tout d'abord elle dépend du type de l'environnement s'il est constitué d'objets variés ou similaires en dimensions en propriétés photométriques etc. Ensuite la détection dépend aussi du mouvement du robot et des différentes parties le constituant. Par exemple lors de la marche un capteur sur le tronc du robot aura un mouvement plutôt rectiligne avec une faible amplitude de balayage vertical tandis qu'un capteur placé sur un membre inférieur aura un mouvement non rectiligne et avec une grande amplitude de balayage vertical. Enfin elle dépend également du type des capteurs utilisés notamment concernant leur portée. Nous restreignons notre étude du placement des capteurs en définissant plusieurs hypothèses de travail que nous détaillons ci-dessous.

#### a) Environnement

Concernant l'environnement plus les objets qui le constituent sont petits moins ils ont de chance d'être détectés même dans la réalité. Nous faisons donc l'hypothèse que les objets de l'environnement ont des dimensions supérieures à  $20\text{cm}$ . Par ailleurs étant donné que nos capteurs actuellement modélisés ne tiennent pas compte des propriétés photométriques des objets nous faisons l'hypothèse qu'un objet sera détecté quelque soit sa nature photométrique.

D'autre part nous supposons que l'environnement n'est pas dynamique. Ainsi aucun obstacle n'évolue dans l'environnement et ne peut par conséquent entrer de lui-même en collision avec le robot en particulier par derrière. Aussi les obstacles avec lesquels le robot est susceptible d'entrer en collision se trouvent devant lui et doivent être détectés au cours du déplacement du robot.

#### b) Mouvement du robot

Un avantage majeur des robots bipèdes sur les robots mobiles à roues est qu'un capteur placé sur une jambe balaye verticalement l'environnement selon les mouvements de balancement de cette jambe pendant la marche. Ce mouvement de balayage vers le haut et vers le bas fournit des informations supplémentaires. Par exemple lorsque le capteur

est positionné sur le tibia le capteur balaye aussi le sol pendant la phase de balancement. Avec le robot bipède nous pouvons ainsi détecter des trous éventuels dans le sol un sol accidenté ou la présence d'escaliers.

### c) Capteurs

Les capteurs proximétriques ayant une portée de 3 mètres nous semblent suffisants puisqu'au-delà de 3 mètres nous pouvons estimer que l'environnement n'est pas local au robot. Ainsi l'information concernant la détection locale d'obstacles n'est pas encore opportune pour faire changer le comportement du robot. De plus le traitement de cette information alors inutile peut faire perdre du temps au traitement des autres informations provoquant un temps de réponse plus long pour l'adaptation du mouvement du robot.

Par ailleurs pour des raisons de sécurité liées au temps de réaction et d'adaptation à l'environnement le robot doit avoir connaissance de la présence d'un obstacle suffisamment tôt temps qui dépend de la vitesse du robot et de la distance de sécurité nécessaire au robot pour s'arrêter. L'arrêt dépend de la position courante du robot. Dans un premier cas le pied de balancement est derrière le pied de support et le mouvement transitoire d'arrêt consiste à amener le pied de balancement près de l'autre pied. Dans un second cas le pied de balancement est devant le pied de support et le mouvement transitoire d'arrêt consiste à poser le pied de balancement normalement puis à amener le nouveau pied de balancement près de l'autre pied. Ainsi dans le pire des cas nous pouvons prendre comme distance de sécurité nécessaire à l'arrêt du robot un peu plus de la longueur d'un demi-pas. Selon les données de notre mouvement de référence de marche un demi-pas fait  $50\text{cm}$ .

D'autre part nous souhaitons voir le robot évoluer sans risque dans un environnement humain. Idéalement le robot doit tout particulièrement percevoir les obstacles et les trous au niveau du sol afin d'adapter son mouvement de marche en conséquence. Par exemple tant qu'il s'agit du sol plat le robot poursuit son avancée mais s'il rencontre un obstacle il doit changer son mouvement : soit l'obstacle est suffisamment petit et alors le robot peut l'enjamber soit c'est une marche et le robot peut monter dessus pour la passer soit l'obstacle est de taille trop grande et le robot doit le contourner. S'il s'agit d'un escalier montant le robot doit alors choisir comme mode de locomotion la marche en montée d'escaliers. Si à l'inverse un trou est détecté dans le sol le robot doit également s'adapter : si le trou est suffisamment petit le robot passe par-dessus sinon il doit chercher un autre chemin. S'il s'agit d'un escalier descendant le robot change de mode de locomotion pour la descente d'escaliers. Un intérêt tout particulier est donc accordé à la détection au niveau du sol. Dans notre cas nous avons commencé à traiter l'obstacle et le trou simples.

Pour réaliser cette détection les capteurs 3D ne sont pas assez précis pour déterminer le point exact rencontré dans l'environnement. En effet nous obtenons la distance capteur-obstacle mais nous ne savons pas déterminer sa direction à cause de la faible directivité de ce type d'ondes. Par conséquent ces capteurs sont utilisés comme détecteurs de présence. Nous pouvons placer ces capteurs sur le tronc et sur les jambes du robot. Toutefois pour des raisons de problèmes de fusion de données multi-capteurs nous n'avons pas traité ce cas. Nous avons préféré nous concentrer sur les capteurs 1D qui sont au contraire

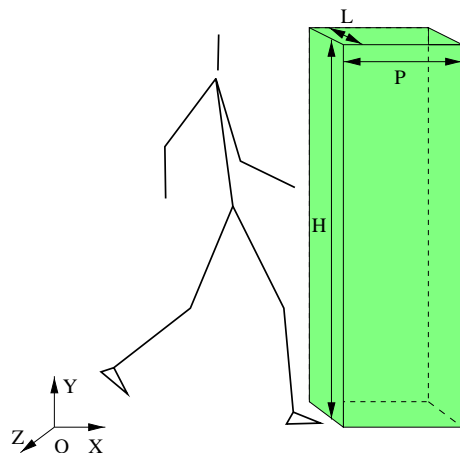


FIG. 7.1: Espace à détecter, situé devant le robot et de dimensions celles du robot.

très précis concernant la direction et la distance aux obstacles ce qui permet de détecter l'environnement de manière plus fine. De la même manière nous ne souhaitons pas avoir des données contradictoires problème traité par la fusion multi-capteurs. Par conséquent nous considérons des modèles de capteurs sans bruit. Ces capteurs sont utilisés pour reconnaître les objets de l'environnement. C'est donc le placement de ces capteurs sur les jambes que nous allons étudier en détail.

### 7.2.2 Outil d'analyse : la carte de projection

Nous souhaitons que le robot perçoive tous les obstacles susceptibles d'entraver son chemin. En effet une collision pourrait grandement endommager le robot en le faisant tomber notamment. Il est donc nécessaire de détecter la présence d'obstacles devant le robot dans un espace dont les dimensions sont celles du robot pour ce qui est de la hauteur et de la largeur (cf. Figure 7.1).

Les capteurs placés en divers endroits sur le robot doivent balayer au mieux de leur rayon cet espace à détecter. Nous choisissons donc de déterminer l'espace balayé par les capteurs au cours d'un cycle de marche. De cette manière nous avons un cycle complet de mouvement pour chaque capteur. Durant ce cycle de marche du robot qui avance selon l'axe  $(OX)$  nous calculons les coordonnées puis la projection sur le plan  $(O, \vec{Z}, \vec{Y})$  du rayon pour tout capteur 1D et de la zone détectée pour tout capteur 3D. Nous obtenons ainsi une carte de projection des endroits détectés dans l'espace par les capteurs pendant un cycle de marche.

En appliquant ce procédé à tous les capteurs positionnés sur le robot nous visualisons les zones détectées selon la configuration des capteurs sur le robot (cf. Figure 7.2). Cette carte de projection permet donc faire apparaître la densité de détection. Nous avons ainsi un outil pour choisir parmi diverses configurations de capteurs celles qui nous paraissent les plus appropriées au sens où tous les obstacles sont détectés avec un nombre restreint de capteurs selon les hypothèses de travail définis dans le paragraphe 7.2.1.

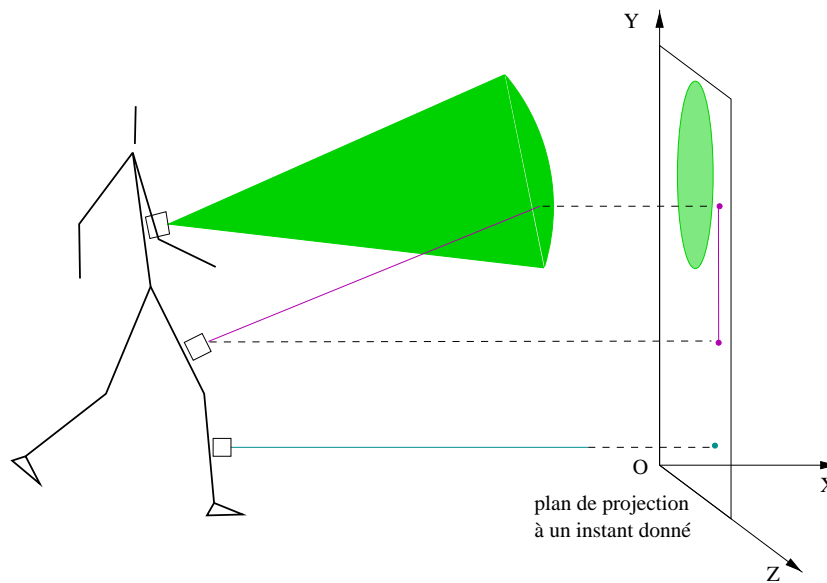


FIG. 7.2: Carte de projection.

### 7.2.3 Analyse de plusieurs placements

Nous souhaitons placer des capteurs sur le robot de manière appropriée. Tout d'abord ce placement dépend fortement du type d'environnement dans lequel le robot doit se déplacer. Puisque notre robot doit évoluer dans des environnements humains nous avons modélisé deux scènes qui représentent des types d'environnements humains possibles. La première scène est une salle à manger composée de différents meubles et d'obstacles posés sur le sol et d'une montée d'escaliers (cf. Figure 7.3). La seconde scène représente une partie de bâtiment comprenant deux pièces de type bureau à deux étages différents l'une au-dessus de l'autre. Pour passer d'une pièce à l'autre il faut emprunter des corridors étroits ainsi que des escaliers en U (cf. Figure 7.4). Nous présentons maintenant la méthodologie utilisée pour étudier le placement des capteurs sur le robot pour ces environnements.

#### a) Méthodologie

Il existe divers types de capteurs et a priori nous pouvons les placer n'importe où et n'importe comment sur le robot. Cependant certains placements sont plus judicieux que d'autres. Nous souhaitons étudier le placement des capteurs sur le robot afin de déterminer celui qui est le plus intéressant c'est-à-dire qui permette de détecter le mieux les obstacles et si possible avec le moins de capteurs pour des raisons de poids d'encombrement et de coût. Comme nous l'avons souligné au paragraphe 7.2.1 nous pouvons mettre à profit le mouvement de balancement des jambes au cours de la marche pour faire avec les capteurs 1D non seulement de la détection d'obstacles au sol mais aussi de la reconnaissance du type des obstacles et de leurs dimensions. Nous nous concentrons donc dans un premier temps sur la détection et la reconnaissance au niveau du sol avec des capteurs 1D placés et orientés dans le plan sagittal. Cela nous permet donc de faire de la reconstruction du profil de l'environnement avec les algorithmes du paragraphe 6.3.3.



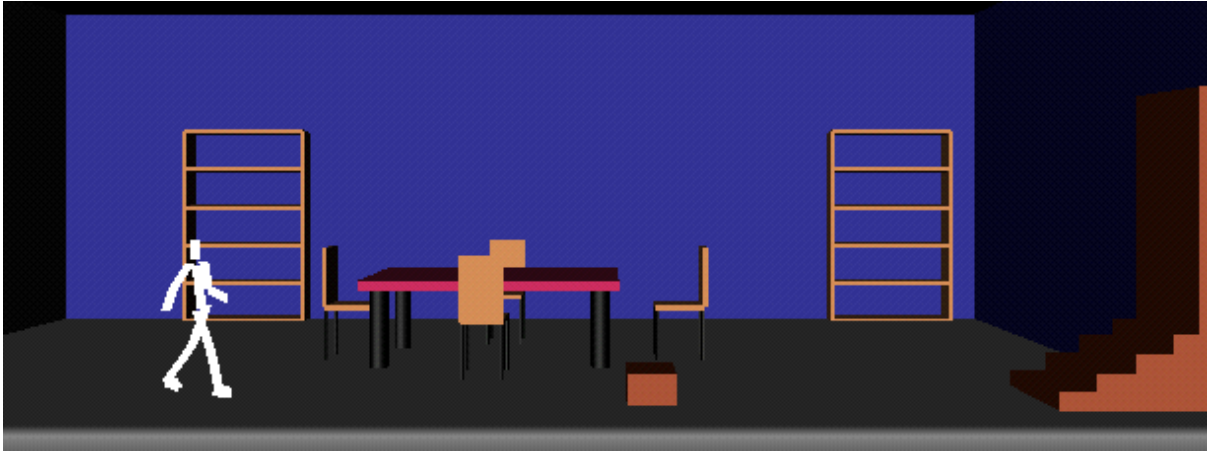


FIG. 7.3: *Première scène modélisée : salle à manger.*

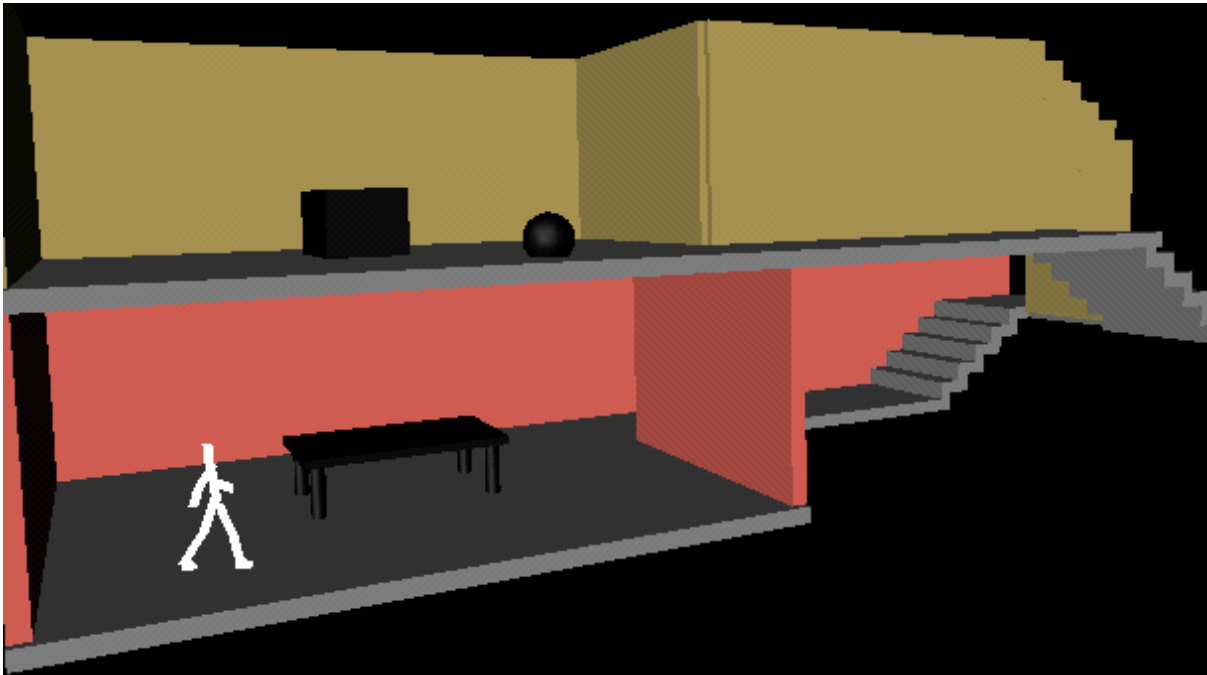


FIG. 7.4: *Deuxième scène modélisée : partie de bâtiment.*

Dans un second temps nous ajoutons d'autres capteurs éventuellement 3D ou orientés de biais par rapport au plan sagittal afin de détecter les obstacles au niveau de la partie supérieure du robot et sur les côtés.

Plusieurs paramètres interviennent dans l'étude du placement des capteurs sur le robot. Tout d'abord les capteurs peuvent être placés en divers endroits sur le robot selon diverses orientations et avec des portées variées. Ensuite le nombre de capteurs en chaque position peut apporter des informations supplémentaires pour la détection de l'environnement "en largeur". Le critère à satisfaire est de détecter l'environnement le mieux possible et suffisamment tôt. En effet le robot doit avoir le temps de s'arrêter en toute sécurité et pour cela il doit avoir la place nécessaire pour finir son pas voire pour faire un pas supplémentaire. Un autre critère est d'assurer une perception continue de l'environnement c'est-à-dire qu'à chaque instant au moins un capteur détecte quelque chose. Nous nous intéressons à la détection d'un obstacle au sol et d'un trou dans le sol que nous considérons comme étant les cas de base des objets que nous pouvons rencontrer dans l'environnement. Nous recherchons la configuration de capteurs qui convient le mieux à cette détection.

Pour cela nous proposons plusieurs positionnements et plusieurs orientations des capteurs sur le robot pour différentes portées de capteurs. Ensuite nous testons les diverses configurations de capteurs obtenues sur des simulations de marche du robot pour détecter un obstacle au sol puis un trou dans le sol. Enfin nous proposons à partir de la meilleure configuration "canonique" obtenue précédemment des configurations analogues mais avec différents nombres de capteurs sur le robot afin de détecter les objets dans la largeur. Nous comparons ces configurations au moyen de la carte de projection et nous en déduisons la configuration finale qui nous paraît la meilleure.

### b) Position des capteurs

Lors du mouvement de marche les jambes du robot effectuent tour à tour des mouvements de balancement. Des capteurs placés sur les jambes offrent alors la possibilité de balayer l'espace verticalement. L'amplitude de ce balayage pour une démarche donnée dépend de la position exacte des capteurs sur le membre support. Par exemple sur la jambe plus un capteur est proche de l'articulation du genou plus son mouvement et l'amplitude de son balayage sont faibles. En revanche plus un capteur est proche de l'articulation de la cheville plus son mouvement et l'amplitude de son balayage sont importants. Nous souhaitons donc déterminer la position sur le membre support qui donne les meilleurs résultats de détection. Nous testons pour la jambe et pour la cuisse les positions suivantes :

- au milieu du support
- au quart du support et non complètement à l'origine de celui-ci car le capteur pourrait gêner l'articulation
- aux trois quarts du support et non complètement à l'extrémité de celui-ci pour la même raison que précédemment.

### c) Orientation des capteurs

Les capteurs peuvent être orientés de plusieurs manières différentes. Le plus intuitif est de les orienter à  $90^\circ$  par rapport à leur membre support dans la direction de la marche du robot. Dans le cas particulier de la détection au niveau du sol d'autres orientations peuvent être plus intéressantes. Par exemple il peut être avantageux de détecter le sol de manière continue pour manquer le moins de choses possible. Or un capteur placé au milieu de la jambe avec une orientation de  $90^\circ$  par rapport au support ne le permet pas. En effet lors de la marche ce capteur ne détecte plus rien lorsque son membre support est en fin de phase de balancement et en début de phase de support. En revanche si ce capteur est orienté à  $30^\circ$  par rapport à son membre support vers le sol il détecte en permanence le sol pendant la marche du robot. L'inconvénient est alors que le capteur détecte les pieds<sup>1</sup> ce qui ne permet pas d'assurer au robot un arrêt en toute sécurité. Il est donc nécessaire de faire un compromis pour la détection permanente du sol par rapport à la distance de sécurité nécessaire au robot que nous estimons à un peu plus de la longueur d'un pas du robot à savoir  $1.2m$  ce qui implique une orientation supérieure à  $80^\circ$  pour la jambe et supérieure à  $60^\circ$  pour la cuisse<sup>2</sup>. Nous testons donc les orientations suivantes :

- pour la jambe :  $90^\circ$  et  $80^\circ$
- pour la cuisse :  $90^\circ$ ,  $75^\circ$  et  $60^\circ$ .

### d) Portée des capteurs

Les capteurs proximétriques peuvent avoir différentes portées. Si un capteur a une trop faible portée il sera limité dans sa détection. Par exemple lors de la marche du robot lequel est initialement placé à  $4m$  d'un obstacle un capteur placé au milieu de la jambe orienté à  $90^\circ$  par rapport à la jambe et ayant une portée de  $1m$  ne détecte l'obstacle qu'à une distance de  $40cm$  ce qui est très inférieur à la distance de sécurité requise pour l'arrêt. D'un autre côté une portée du capteur trop grande n'apporte rien de plus à la détection locale de l'environnement que ne peut le faire une portée moyenne. Nous déterminons la portée minimale des capteurs pour la jambe et la cuisse en calculant la distance entre le capteur lorsque le robot est en position debout standard et le point situé sur le sol à  $1.2m$  (distance de sécurité) devant les pieds du robot<sup>3</sup>. Nous majorons cette distance et nous obtenons une portée minimale de  $1.5m$  pour la jambe et pour la cuisse. Nous testons donc pour les capteurs de la jambe et de la cuisse les portées suivantes :  $1.5m$ ,  $2m$  et  $3m$ .

### e) Analyse des placements des capteurs

Nous avons testé toutes les configurations de capteurs précédemment citées afin de déterminer la meilleure configuration qui permette de détecter les obstacles et les trous

<sup>1</sup> $d = l_1 \cdot \tan 30^\circ \simeq 10cm$ , avec la demi-longueur de la jambe  $l_1 \simeq 20cm$ , alors que la longueur du pied est  $l_0 \simeq 20cm$ .

<sup>2</sup> $\theta = \arctan(\frac{120}{d_1})$ , donc  $\theta \simeq 80.5^\circ$  pour la jambe ( $d_1 = l_1 \simeq 20cm$ ), et  $\theta \simeq 63.5^\circ$  pour la cuisse ( $d_2 = 2l_1 + l_2 \simeq 60cm$ ).

<sup>3</sup> $D = \sqrt{120^2 + d^2}$ , donc  $D \simeq 1.21m$  pour la jambe ( $d_1 = l_1 \simeq 20cm$ ), et  $D \simeq 1.34m$  pour la cuisse ( $d_2 = 2l_1 + l_2 \simeq 60cm$ ).

suffisamment à temps. Pour cela nous avons placé le robot à différentes distances de l'obstacle et du trou :  $2m$ ,  $2.25m$ ,  $2.5m$  et  $2.75m$  pour vérifier que la détection ne dépend pas de la position initiale du robot. Puis pour la jambe et la cuisse nous avons testé pour chacune des trois positions du capteur sur le support les trois orientations du capteur et pour chacune d'elles les trois portées. Le nombre de simulations s'élève donc à :

$$\begin{aligned}
 & \text{pour la jambe : } ( \quad (1 \text{ pour l'obstacle} + 1 \text{ pour le trou}) \\
 & \quad \times 4 \text{ distances} \\
 & \quad \times 3 \text{ positions} \\
 & \quad \times 2 \text{ orientations} \\
 & \quad \times 3 \text{ portées} \\
 & \quad ) \\
 & \quad + \\
 & \text{pour la cuisse : } ( \quad (1 \text{ pour l'obstacle} + 1 \text{ pour le trou}) \\
 & \quad \times 4 \text{ distances} \\
 & \quad \times 3 \text{ positions} \\
 & \quad \times 3 \text{ orientations} \\
 & \quad \times 3 \text{ portées} \\
 & \quad ) \\
 & = 360 \text{ simulations.}
 \end{aligned}$$

Nous donnons les courbes de distances et les profils d'environnement représentatifs obtenus lors des simulations en Annexe C. En étudiant les courbes de distances nous avons déterminé les configurations qui ne satisfont pas le critère lié à la distance de sécurité pour l'arrêt du robot c'est-à-dire celles pour lesquelles l'obstacle ou le trou sont détectés pour la première fois à une distance inférieure à cette distance de sécurité. En comparant ensuite les différents profils nous déterminons les meilleurs résultats obtenus pour la détection de l'obstacle et du trou. Nous en déduisons que la meilleure configuration canonique de capteurs sur le robot est celle où le capteur sur la jambe est placé au milieu orienté de  $90^\circ$  par rapport à la jambe avec une portée de  $2m$  et le capteur sur la cuisse est placé au milieu orienté de  $60^\circ$  par rapport à la cuisse avec une portée de  $3m$  (cf. Figure 7.5).

#### f) Nombre de capteurs et analyse

Nous avons jusqu'à présent étudié les capteurs orientés dans la direction de marche du robot. Il peut s'avérer intéressant d'étudier également le résultat donné par des capteurs orientés légèrement de biais par rapport aux capteurs initiaux de manière à détecter un peu sur les côtés. En effet en supposant que les capteurs des deux jambes sont distants de  $30cm$  tout obstacle de  $20cm$  situé entre les deux pieds ne sera jamais détecté. De plus si le rayon du capteur est fini il n'en va pas de même pour le pied. Celui-ci mesure environ  $18cm$  de large et peut très bien entrer en collision avec un obstacle décalé de  $5cm$  par rapport au capteur. Pour éviter ce genre de situations nous proposons de mettre plusieurs capteurs sur les emplacements de la configuration trouvée dans le paragraphe précédent. Nous testons plusieurs nombres de capteurs pour chaque emplacement : 1, 3 et 5 et plusieurs angles d'ouverture entre les deux rayons extrêmes :  $20^\circ$ ,  $30^\circ$  et  $40^\circ$ .

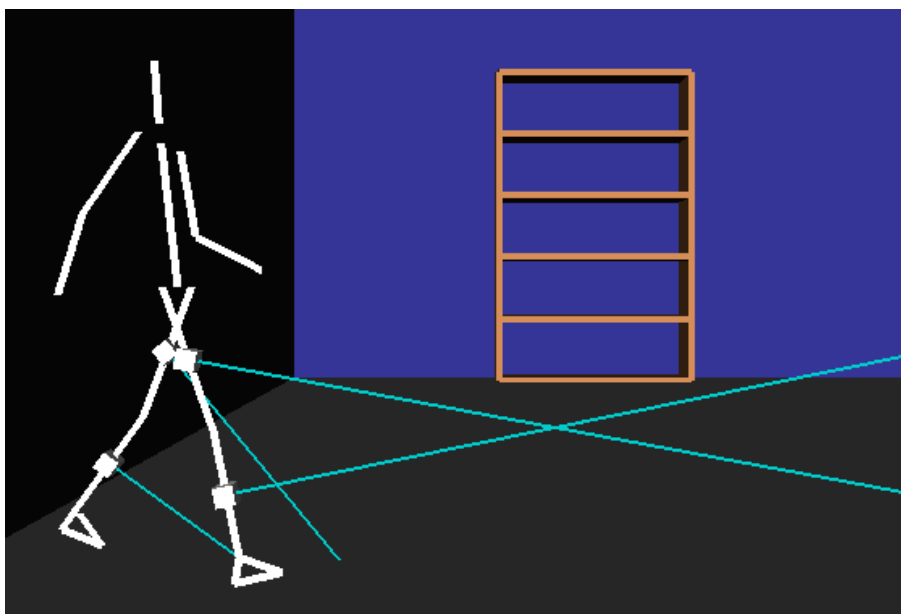
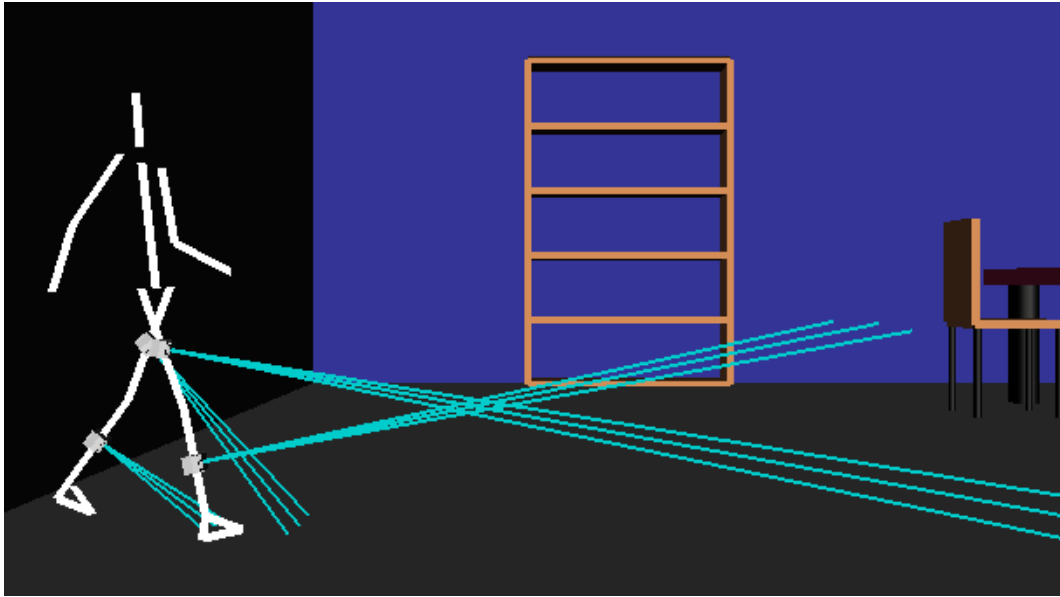


FIG. 7.5: Meilleure configuration canonique de capteurs sur le robot, selon nos hypothèses.

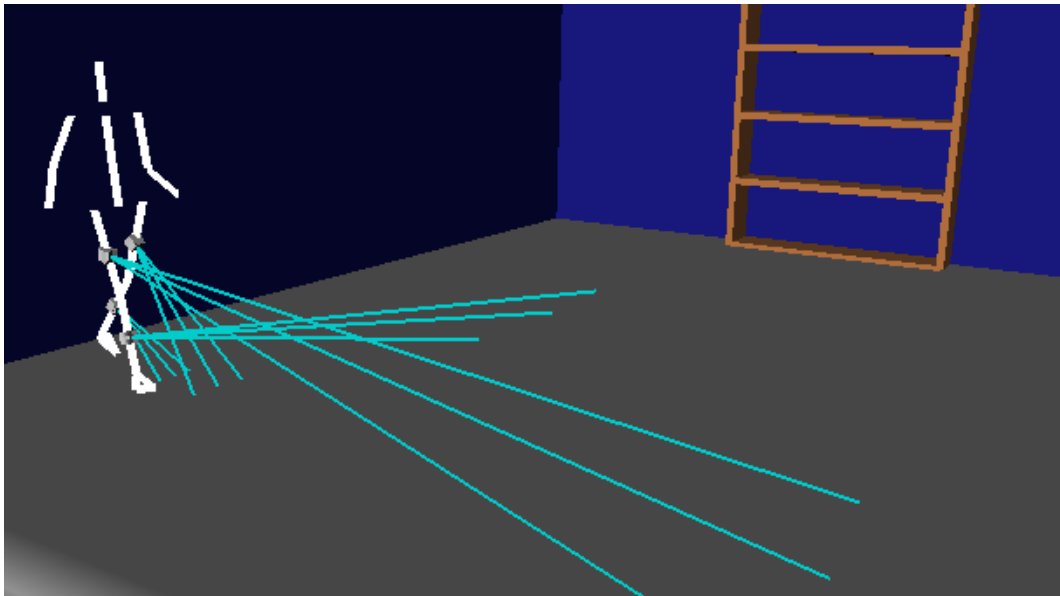
Nous calculons les cartes de projection pour chacune de ces configurations (cf. Annexe C). Nous comparons toutes ces cartes de projection pour trouver la meilleure configuration. Il s'agit de celle qui réalise le meilleur compromis entre la détection et le nombre de capteurs. Finalement nous trouvons que la meilleure configuration de capteurs sur le robot pour la détection au niveau du sol est la configuration avec 3 capteurs sur chaque support la jambe et la cuisse et avec un angle d'ouverture de  $20^\circ$  entre les deux rayons extrêmes de chaque groupe de capteurs (cf. Figure 7.6).

#### 7.2.4 Discussion

Nous avons choisi d'étudier le placement des capteurs sur le robot dans le cadre de la détection d'obstacles et de trous au niveau du sol. Trois paramètres liés aux capteurs sont à considérer : la position sur le support l'orientation par rapport à ce support et la portée de chaque capteur. Nous avons testé plusieurs valeurs pour chacun de ces paramètres lors de simulations de la marche du robot pour détecter un obstacle puis un trou à partir de différentes distances de départ pour le robot. Nous en avons déduit à partir des courbes de distances et des profils calculés une première configuration canonique de capteurs. Nous avons amélioré cette configuration en ajoutant aux capteurs initiaux des capteurs orientés légèrement de biais par rapport à ceux-ci. Nous avons testé plusieurs nombres de capteurs et plusieurs angles d'ouverture entre les rayons extrêmes de chaque groupe de capteurs. Puis nous avons comparé les différentes configurations obtenues au moyen de la carte de projection. Finalement nous avons choisi la configuration finale qui offre le meilleur compromis entre détection et nombre de capteurs. Il s'agit de la configuration de la Figure 7.6. Nous utilisons cette configuration de capteurs dans le paragraphe suivant pour réaliser une simulation complète du robot dans l'environnement.



a) Vue de profil.



b) Vue décalée.

FIG. 7.6: Meilleure configuration finale de capteurs sur le robot, selon nos hypothèses.

## 7.3 Déclenchement des trajectoires transitoires

Nous souhaitons réaliser une simulation complète du robot dans son environnement. Il s'agit de faire marcher le robot dans l'environnement tout en lui faisant détecter cet environnement. Lorsque les capteurs délivrent des informations relatives à un changement dans l'environnement le robot doit adapter son mode de locomotion en conséquence. Normalement ce type de simulation doit être réglé par le contrôle de mouvement du robot qui sera développé par les roboticiens. Cependant pour valider nos travaux nous avons développé un contrôle simplifié adapté à notre étude.

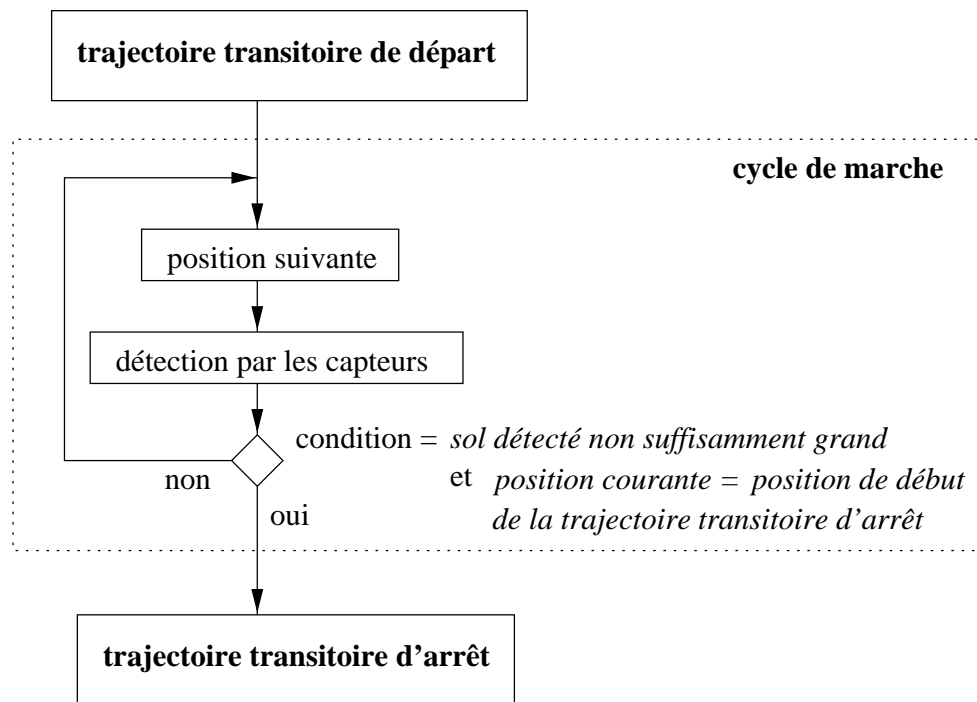
### 7.3.1 Contrôle de haut niveau

Au niveau du contrôle du robot il s'agit de prendre en compte l'information délivrée par les capteurs et ainsi d'adapter le comportement du robot à l'environnement. Selon l'information retournée par le capteur le contrôle change de type de démarche. Par exemple le robot peut passer de la marche sur sol plat à la montée ou la descente d'escaliers. Ou encore il peut adapter la longueur et la hauteur du pas pour enjamber un obstacle détecté s'il est suffisamment petit.

Actuellement nous n'avons comme mouvements de référence pour le robot que le cycle de marche à plat et les mouvements transitoires de départ et d'arrêt. La simulation consiste alors à faire démarrer le robot puis à le faire avancer en marchant jusqu'à ce que les capteurs placés sur le robot selon la configuration de la Figure 7.6 détectent et reconnaissent un obstacle le contrôle ordonnant alors l'arrêt du robot. Nous avons donc réalisé un contrôle de haut niveau simplifié (cf. Figure 7.7). Ce contrôle peut être aisément enrichi avec d'autres types de démarches (la montée et la descente d'escaliers le demi-tour la course etc.) et d'objets à détecter (la détection de portes des fenêtres de plans inclinés etc.).

Concrètement le contrôle actuel comprend trois étapes. La première étape réalise la trajectoire transitoire de départ de marche du robot. Cela consiste à calculer les positions à prendre par le robot depuis la position debout standard jusqu'à la posture à atteindre dans le cycle de marche. La deuxième étape correspond au cycle de marche. Une première sous-étape calcule la position suivante du robot puis une seconde sous-étape effectue la détection par les capteurs. À partir de cette détection de l'environnement nous comparons la distance entre le robot et le dernier point du segment du sol horizontal avec la longueur du demi-pas du robot. Si c'est supérieur le robot fait un demi-pas supplémentaire sinon la trajectoire transitoire d'arrêt est déclenchée. Enfin la dernière étape réalise la trajectoire transitoire d'arrêt du robot pour que celui-ci se retrouve dans la position debout standard.

Nous remarquons que cet algorithme de contrôle fonctionne aussi bien dans le cas d'obstacles que de trous y compris dans le cas de trous de dimensions supérieures à la portée des capteurs.

FIG. 7.7: *Algorithme de contrôle du robot.*

### 7.3.2 Simulation

Nous avons placé le robot en position debout standard dans la salle à manger virtuelle à  $5m$  d'un obstacle à détecter. Nous avons ensuite placé les capteurs sur le robot selon la configuration de la Figure 7.6. Au niveau contrôle nous avons choisi le mode de locomotion initial comme étant la marche et nous avons lancé la simulation.

Le robot effectue le mouvement transitoire de départ et commence un cycle de marche. Les capteurs placés sur les jambes et orientés vers le sol délivrent les distances détectées (cf. Figure 7.8 (a)). Ces distances sont utilisées en continu pour calculer le profil d'environnement que l'algorithme de reconnaissance interprète comme étant le sol. Nous pouvons remarquer sur les courbes de la Figure 7.8 (a) qu'à tout instant tous les capteurs ne donnent pas la distance maximale c'est-à-dire qu'il y a toujours au moins un capteur qui perçoit quelque chose. Ainsi il n'y a pas d'interruption dans la détection de l'environnement au cours du temps. Par conséquent non seulement il est intéressant de placer les capteurs sur chaque membre du robot pour détecter les obstacles à leur niveau mais cela permet aussi d'assurer une détection sans interruption du sol. Nous retrouvons ce résultat sur le profil d'environnement (cf. Figure 7.8 (b)).

Au bout de deux cycles de marche du robot les capteurs du tibia droit détectent les premiers un objet à une distance de  $1.5m$  environ. Le profil d'environnement permet d'obtenir la hauteur du point détecté qui est de  $20cm$ . L'algorithme de reconnaissance en déduit qu'il ne s'agit plus du sol mais d'un obstacle qui est en train d'être détecté de  $20cm$  de hauteur au moins. Le contrôle du robot en est informé et modifie le mode de



locomotion du robot pour effectuer le mouvement transitoire d'arrêt. Et le robot s'arrête retrouvant sa position standard debout à  $70\text{cm}$  de l'obstacle. Le profil d'environnement final permet d'obtenir la hauteur totale de l'obstacle qui est de  $25\text{cm}$ .

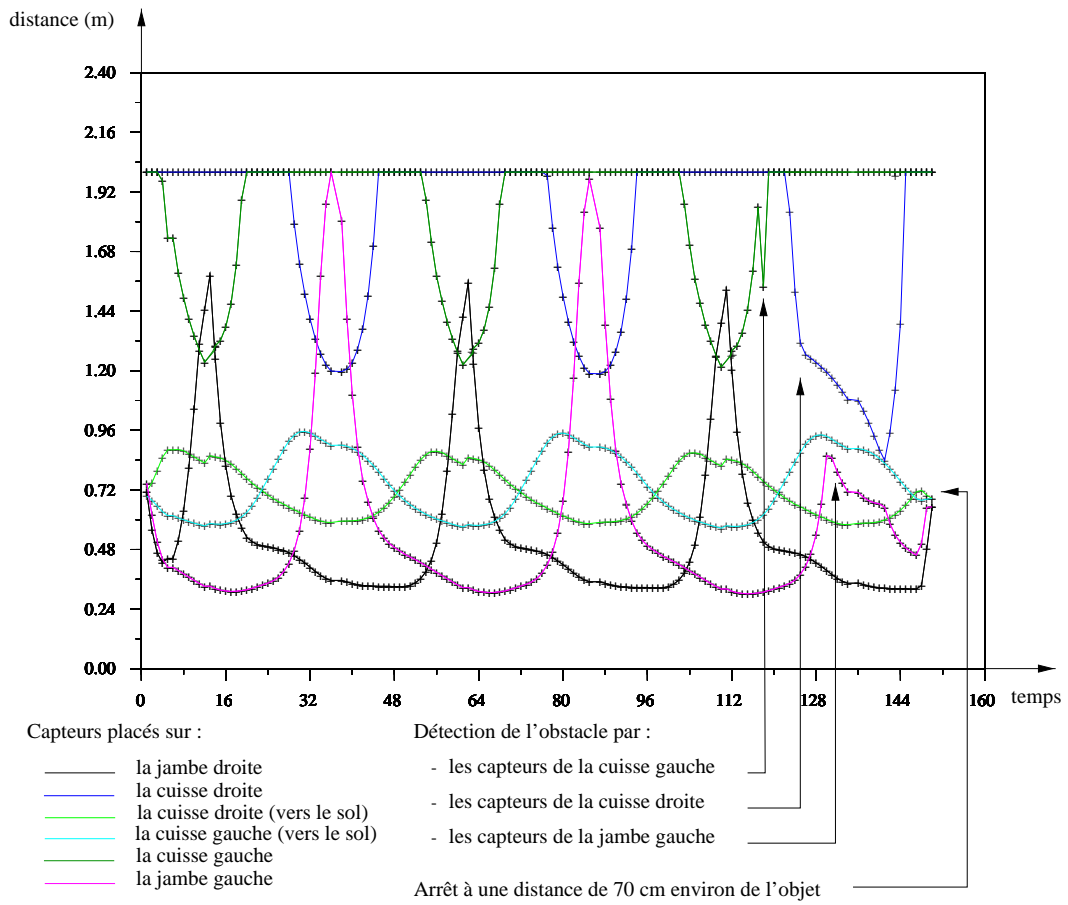
## 7.4 Conclusion

Nous avons proposé des outils pour étudier le placement des capteurs sur le robot nécessaire à une bonne perception de l'environnement. Ces outils la carte de projection mais aussi le profil de l'environnement et l'algorithme de reconnaissance au vu des tests réalisés et des résultats obtenus se révèlent être satisfaisants et nous permettent de définir une configuration des capteurs sur le robot convenable au sens de la détection maximale et du nombre minimal de capteurs pour un environnement particulier. Nous avons obtenu la configuration de la Figure 7.6 pour la détection d'obstacles et de trous au niveau du sol.

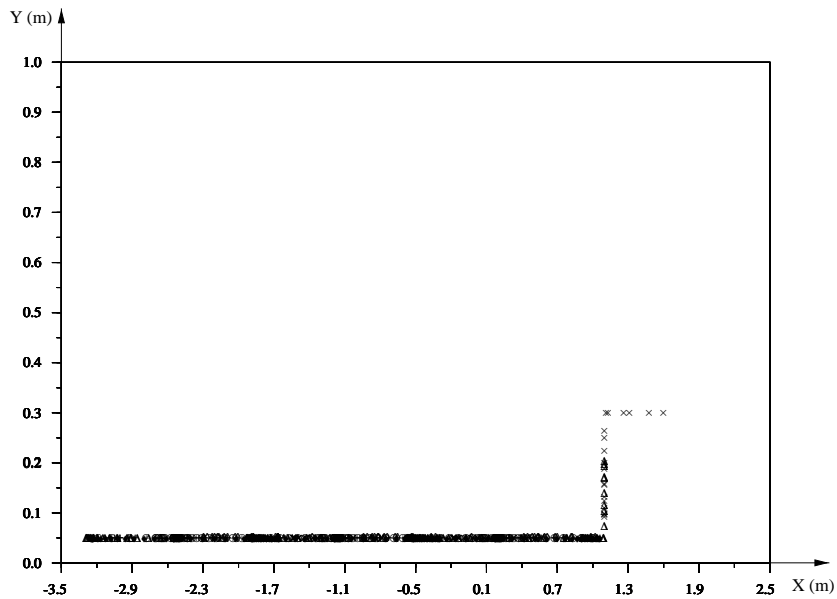
Cette configuration est dépendante du type d'environnement du robot. Pour des environnements homogènes c'est-à-dire de même nature géométrique pour le type d'objets à détecter et de même nature photométrique pour les ondes des capteurs le nombre de capteurs nécessaires est moins important que pour des environnements hétérogènes. De plus le type des capteurs utilisés dépend de la nature de l'environnement. Par exemple s'il s'agit de pièces composées en grande partie d'objets métalliques hautement spéculaires telles que des usines métallurgiques les capteurs optiques ne sont pas d'une grande utilité. En effet l'effet de diffusion étant absent les rayons émis ne sont jamais détectés à l'exception de ceux réfléchis par des obstacles perpendiculaires à la direction des rayons incidents. Il faudrait donc choisir entre les divers types de capteurs en fonction de la spécularité et autres propriétés photométriques des objets composant l'environnement.

Ainsi cette étude doit largement être étendue à d'autres environnements. Elle doit en outre être étendue à la reconnaissance de classes d'objets supplémentaires. Elle peut aussi être affinée pour reconnaître notamment le sol non plat qui engendrerait des perturbations sur le mouvement du robot et sur les détections des capteurs pouvant très bien rendre ces résultats inapplicables. Cette étude ainsi complétée permettra d'obtenir une meilleure connaissance de l'environnement et ainsi d'assurer un contrôle du robot mieux adapté et plus performant.

Nous avons ensuite utilisé la configuration de capteurs de la Figure 7.6 pour réaliser une simulation complète du robot dans l'environnement. En effet le robot doit s'adapter à l'environnement en fonction des informations fournies par les capteurs. Pour cela nous avons défini un contrôle simplifié du robot permettant de réaliser un mouvement transitoire de départ puis un ou plusieurs cycles de marche et enfin un mouvement transitoire d'arrêt en fonction des informations délivrées par les capteurs et analysées par les outils développés dans le chapitre 6. Nous avons ainsi obtenu une simulation où le robot démarre puis marche pendant un certain temps et enfin s'arrête lorsque ses capteurs ont détecté un obstacle. Ce contrôle peut être complété par d'autres démarches et d'autres types d'objets détectables.



a) Courbe de distances.



b) Profil d'environnement.

FIG. 7.8: Simulation complète du robot dans l'environnement.

Concernant les démarchesΓ étant donnés les mouvements de référence actuellement définisΓ nous ne pouvons les moduler. OrΓ il peut s'avérer intéressant de modifier la démarche (longueur de pasΓ hauteur de pasΓ etc.). Si par exempleΓ le robot arrive à 50cm environ d'une marche d'escaliersΓ il ne peut faire un pas de plusΓ son pas étant plus long que l'espace disponibleΓ et il ne peut commencer la marche dans l'escalier sans risque de poser le pied à côté de la marcheΓ entraînant un déséquilibre et la chute du robot. Une amélioration importante consiste à savoir adapter la longueur et la hauteur du pas du robot et à réaliser cette adaptation par le contrôle de haut niveau.

# Chapitre 8

## Conclusion

### 8.1 Bilan

Notre étude s'est portée sur la réalisation d'un simulateur d'un robot bipède. En effet le projet BIP dans lequel s'inscrit cette thèse s'intéresse à la construction et au contrôle d'un robot bipède anthropomorphe (cf. Figure 8.1). Notre simulateur doit permettre aux roboticiens du projet de tester et de valider les lois de contrôle développées sur le robot virtuel grâce à une interface entre l'intégrateur numérique des roboticiens et notre simulateur avant de les appliquer au robot réel. Il faut donc que le simulateur corresponde au maximum à la réalité. Nous avons utilisé la synthèse d'images qui permet d'avoir un retour visuel de qualité et nous avons développé des fonctions particulières pour ce simulateur.

#### 8.1.1 Robot virtuel

Nous avons modélisé le robot sous forme de structure articulée grâce à la notation Denavit-Hartenberg en représentant seulement le squelette. Concernant le mouvement de ce robot nous avons abordé son contrôle sous son aspect robotique. Les roboticiens devraient réaliser ce contrôle par fonction de tâches. Cette technique tente de contrôler le mouvement du robot en réduisant l'erreur entre ce mouvement et un mouvement de référence. Toutefois aucun contrôle n'est disponible pour la marche actuellement. Dans le but de tester le simulateur nous avons donc choisi de fournir à notre robot directement un mouvement de référence. Ce mouvement pourra en outre être utilisé par les roboticiens pour développer leur contrôle par fonction de tâches.

Nous avons cherché à déterminer ce mouvement de référence. Une étude biomécanique de la marche humaine nous a montré à quel point il est délicat de trouver des paramètres intuitifs pour réaliser ce mouvement. En effet la nature biomécanique des jambes est très compliquée rendant le processus de marche complexe et difficilement réalisable à partir de peu de donnée. Nous nous sommes donc intéressés à l'obtention de ce mouvement en étudiant les méthodes utilisées en synthèse d'images pour animer des personnages virtuels. Parmi les principales méthodes basées sur la cinématique la dynamique ou la capture de mouvements ce sont ces dernières qui nous ont paru le mieux adaptées à notre cadre. Elles permettent en effet d'avoir directement un mouvement de nature réaliste de par la technique employée qui consiste à enregistrer le mouvement d'un sujet humain.

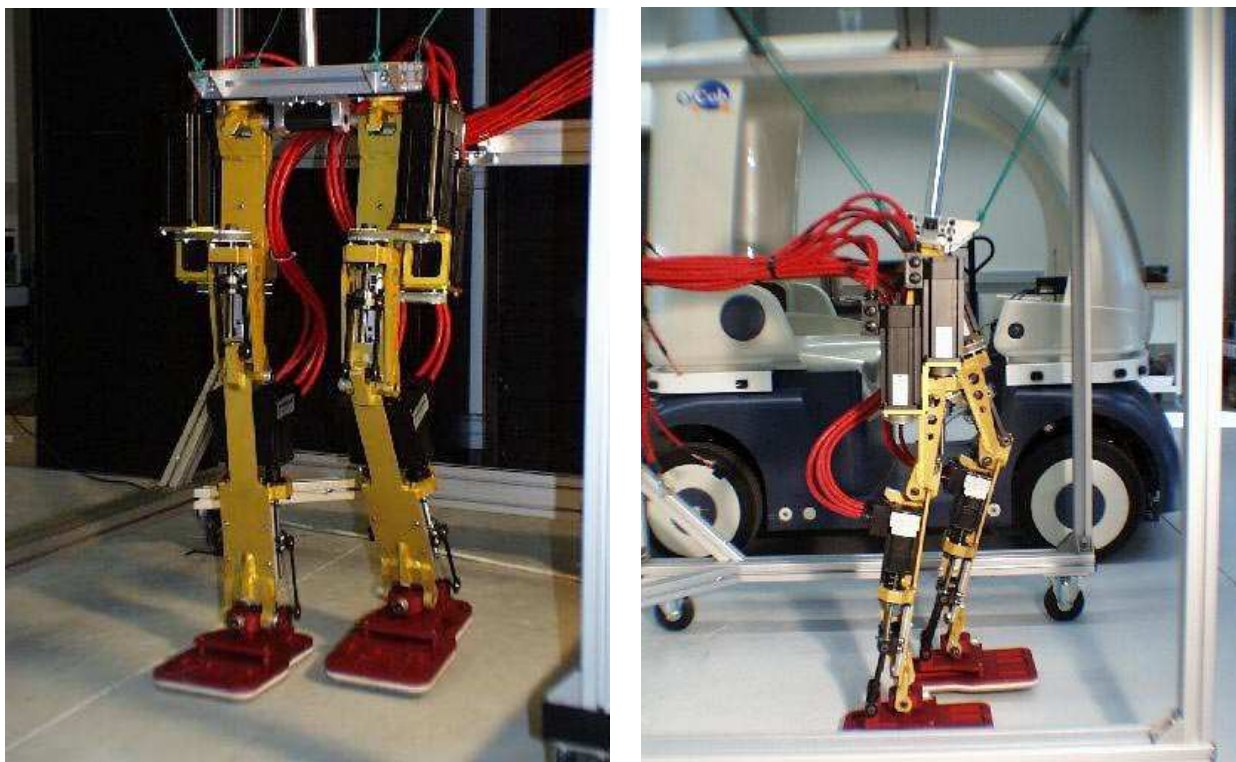


FIG. 8.1: Photographies du robot anthropomorphe réel.

Nous avons donc obtenu des données de cycles de marche enregistrés. Nous les avons adaptées pour obtenir un cycle de marche correct pour notre robot. Cependant par souci d'application future de ces mouvements au robot réel par l'intermédiaire de la fonction de tâches ce cycle de marche apparaît insuffisant. En effet les postures prises par le robot au cours d'un cycle de marche sont toutes en équilibre instable. Or le robot doit débuter son mouvement à partir d'une posture où il est en équilibre stable statique typiquement en position debout standard. Nous avons eu besoin de calculer des mouvements transitoires de départ et de manière symétrique d'arrêt ces mouvements ne nous ayant pas été fournis. Nous avons proposé une méthode basée sur l'interpolation de polynômes cubiques afin d'obtenir des mouvements transitoires physiquement réalistes. Nous avons ainsi défini un mouvement de référence complet pour la marche du robot.

### 8.1.2 Perception virtuelle

Pour qu'il puisse évoluer dans un environnement en toute sécurité le robot doit percevoir son environnement et reconnaître les objets qui le constituent afin d'être le plus autonome possible. Nous avons donc étudié du point de vue robotique comment les robots réels réalisent ces deux fonctions indispensables. Concernant la perception de l'environnement les robots peuvent être équipés de caméras et/ou de capteurs. La caméra fournit un ensemble complet mais complexe d'informations tandis que les capteurs renseignent sur un type particulier d'information plus facile et rapide à traiter. Ainsi la caméra est plutôt

utilisée pour les tâches de haut niveau telles que la navigation dans un environnement et la planification de trajectoire alors que les capteurs sont utilisés pour des tâches de bas niveau en général pour simplifier la tâche de la caméra en fournissant directement les informations nécessaires sans être obligé de traiter une image complexe pour les obtenir. Nous nous sommes intéressés tout particulièrement à la détection locale d'obstacles pouvant gêner le chemin du robot. Les capteurs assignés à cette tâche sont les capteurs proximétriques. Les plus employés en robotique sont les capteurs optiques et à ultra-sons qui sont basés sur la propagation d'onde et fournissent la distance aux obstacles détectés. Cette distance permet au robot de ne pas rentrer en collision avec les objets mais ne lui fournit aucune indication quant à la nature des objets. Il ne peut par conséquent adapter son mode de locomotion en fonction de l'environnement. Des méthodes de reconnaissance de formes sont alors appliquées pour répondre à cette attente. Elles permettent de fournir des informations de plus haut niveau que les simples distances ou points détectés au moyen d'analyses effectuées sur ces données de base. En particulier elles tentent de reconstruire puis de reconnaître la forme géométrique des objets détectés. Le robot peut alors s'adapter à l'environnement en fonction des objets reconnus.

Dans un premier temps nous avons modélisé des capteurs proximétriques au moyen d'outils d'infographie. Nous avons utilisé le lancer de rayon et la caméra perspective pour déterminer la distance du capteur aux objets détectés ainsi que l'angle d'incidence. Nous avons calculé la distance résultat en fonction de cette distance calculée de l'angle d'incidence et d'un bruit ajouté pour prendre en considération les perturbations des dispositifs électroniques réels. Nous avons ainsi développé des capteurs 1D modélisant des capteurs optiques et à ultra-sons à haute fréquence ainsi que des capteurs 3D modélisant des capteurs à ultra-sons à basse fréquence.

Dans un deuxième temps nous avons développé des outils d'analyse pour déterminer la nature des objets à partir des informations brutes des capteurs modélisés. Le premier outil est la courbe de distances qui permet d'avoir une représentation graphique des distances détectées pendant la simulation au cours du temps. Elle permet en outre de connaître les périodes pendant lesquelles un capteur ne détecte rien par exemple pour ajouter un autre capteur dans le but d'avoir une détection continue de l'environnement dans le temps. Le deuxième outil est le calcul du profil d'environnement à partir des distances détectées. Cet outil permet de retrouver les points touchés dans l'environnement et ainsi de visualiser ce qui a été perçu. Le troisième outil est l'algorithme de reconnaissance. Cet algorithme permet de reconnaître les objets détectés à partir des points du profil d'environnement. Dans un troisième temps nous avons présenté des tests réalisés avec des capteurs réels dans le but de valider nos modèles de capteurs. Au vu des résultats comparatifs mais non exhaustifs il est apparu que nos modèles étaient trop pessimistes et qu'un capteur pour lequel l'angle d'incidence n'est pas pris en compte donne de meilleures correspondances avec les enregistrements obtenus par les capteurs réels du moins pour les situations choisies.

### 8.1.3 Applications

Une fois les capteurs modélisés et simulés, nous les avons utilisés pour réaliser deux applications. La première application a consisté à étudier le placement des capteurs sur le robot réel. En effet, une étude est nécessaire pour pouvoir définir la meilleure configuration de capteurs sur le robot, au sens du nombre de capteurs et de la détection maximale des obstacles dans l'environnement. Nous avons, à cet effet, développé un outil supplémentaire qui est la carte de projection, afin de visualiser la zone devant le robot, balayée par les capteurs placés sur le robot pendant un cycle de marche. Nous avons alors fait une proposition de configuration de capteurs qui nous paraisse convenir le mieux. Toutefois, la qualité de cette configuration dépend de l'environnement et la proposition que nous avons effectuée peut nécessiter une adaptation en cas d'environnement différent.

La seconde application concerne la réalisation d'une simulation complète du robot dans l'environnement. Nous avons réalisé un contrôleur simplifié de manière à changer le mode de locomotion pour le robot, selon les informations retournées par les capteurs et concernant l'environnement local perçu. Nous avons ainsi obtenu une simulation du robot dans l'environnement. À partir de la position debout standard, le robot a réalisé le mouvement transitoire de départ, puis a poursuivi son mouvement par un cycle de marche. Les capteurs fournissent des informations indiquant que le sol est détecté. À un instant donné de la simulation, un capteur détecte un objet, qui est reconnu par l'algorithme de reconnaissance comme étant un obstacle de hauteur  $25\text{cm}$  au moins. Le contrôle déclenche alors le mouvement transitoire d'arrêt, et le robot s'arrête.

### 8.1.4 Discussion

Finalement, notre simulateur n'est pas entièrement fini, mais les fonctionnalités principales ont été réalisées. En effet, il est possible de fournir un mouvement à faire exécuter par le robot, ce mouvement pouvant ultérieurement être calculé par le contrôle par fonction de tâches. D'autre part, les fonctions de perception et de reconnaissance ont été réalisées et permettent au robot de connaître l'environnement dans lequel il évolue. Ces fonctions ont aussi permis la possibilité d'étudier le placement des capteurs sur le robot selon l'environnement à considérer. Une idée de contrôle de haut niveau a également été proposée, afin de réaliser un test complet de notre simulateur de robot. Notre simulateur peut être, cependant, amélioré à plusieurs points de vue. Nous présentons à cet effet plusieurs perspectives.

## 8.2 Perspectives

### 8.2.1 Mouvements de référence divers et variés

Concernant les mouvements de référence du robot, il serait très avantageux de récupérer d'autres démarches, ainsi que de calculer les mouvements transitoires correspondants, de manière à mieux adapter le robot à son environnement. De plus, le robot a tout intérêt à pouvoir adapter la longueur de son pas, par exemple pour avancer jusqu'à la bonne

distance à une marche afin d'enchaîner tout de suite avec le mouvement de montée d'escaliers. Il a aussi intérêt à savoir adapter la hauteur de son pas pour s'adapter aux marches de différents escaliers qui peuvent être de tailles non identiques.

Des travaux ont été réalisés sur l'adaptation de mouvements en synthèse d'images. [WF97] animent des structures articulées par évolution angulaire mais dans le but d'animer la croissance plutôt que le mouvement à proprement parler. [HP97] font de la simulation dynamique à l'aide de contrôleurs proportionnels-dérivés et adaptent le mouvement en changeant l'échelle de certains paramètres concernant l'état du système les contrôleurs et le pas de temps d'intégration. Ils règlent également les paramètres manuellement pour changer la morphologie des personnages. [Gle98] a réalisé un travail plus proche de ce que nous recherchons car il permet d'adapter un mouvement calculé sur un personnage à un autre personnage. Pour cela il identifie les contraintes permettant de réaliser le mouvement initial puis choisit une représentation du mouvement et résout le problème contraint non-linéaire. Nous pourrions adapter cette méthode pour calculer des longueurs et des hauteurs de pas différentes de celles de la démarche de référence originale et ainsi obtenir un mouvement à pas variable permettant au robot de mieux s'adapter à son environnement.

### 8.2.2 Vue améliorée

Concernant la perception locale et les capteurs modélisés de plus amples tests avec des capteurs réels sont nécessaires pour avoir une meilleure connaissance de leurs propriétés. En particulier le bruit lié au dispositif pourrait être mieux modélisé de manière à élaborer le contrôle en fonction de ces perturbations qui se retrouveront pour le robot réel. D'autre part les capteurs actuellement modélisés ne tiennent pas compte de la spécularité des objets qui intervient pourtant de manière non négligeable dans la réalité. Une amélioration consisterait donc à définir aux objets de l'environnement des propriétés supplémentaires pour considérer la spécularité de l'objet en fonction des longueurs d'ondes qui seront considérées pour les capteurs.

### 8.2.3 Vers l'autonomie du robot

Dans le but d'améliorer la reconnaissance des objets dans l'environnement il serait nécessaire de considérer un plus large éventail d'objets à déterminer selon les environnements dans lesquels le robot réel devra évoluer. De plus le profil de l'environnement n'est pas suffisant il faudrait aussi pouvoir considérer l'environnement dans ses trois dimensions. Nous pourrions utiliser la méthode de [RTJ89] pour reconstruire l'environnement 3D perçu basée sur l'information de profondeur. Nous pourrions également mettre au point un meilleur algorithme de reconnaissance.

Le but final étant de rendre le robot le plus autonome possible une des tâches principales que nous pourrions réaliser est la navigation dans un environnement. Cette tâche nécessite d'avoir des informations plus importantes et de haut niveau concernant l'environnement. La caméra est alors utilisée pour offrir une perception globale de l'environnement. Il devient alors nécessaire de traiter les informations complexes des images captées par la



caméra qui requiert d'avoir recours à des techniques plus élaborées comme l'analyse d'images techniques bien connues de vision par ordinateur.

Une autre fonction permettant d'améliorer la tâche de navigation est la mémorisation de l'environnement. Il s'agit de cartographier l'environnement à partir des informations fournies par les divers types de perception. Une méthode qui pourrait être utilisée consiste à définir une grille d'occupation des objets qui est mise à jour au fur et à mesure de l'évolution du robot dans son environnement. Ainsi lorsque celui-ci repasse par un endroit il peut utiliser la grille d'occupation comme dans [Elf90] pour savoir plus vite où se trouvent les objets dans les environs et ainsi planifier et agir plus rapidement.

#### 8.2.4 Exploitation du simulateur

Il serait intéressant de faire le lien entre le simulateur et la simulation dynamique en testant des lois de commande de marche du robot sur le simulateur. Il peut être en effet avantageux et aisé de vérifier si des lois de commande simples aboutissent à des résultats pour des mouvements du robot analogues à ceux de lois plus complexes mais en des temps réduits. D'autre part les études réalisées dans cette thèse peuvent être utilisées sur le robot réel en particulier en ce qui concerne le placement des capteurs sur ce robot et l'algorithme de reconstruction de l'environnement pour aider à la navigation du robot. Et dans l'hypothèse où d'autres robots bipèdes seraient construits le simulateur peut s'avérer utile pour définir au préalable les choix technologiques concernant la conception de ces robots.

# Annexe A

## Modèle d'éclairage

### A.1 Interaction lumière/objet en synthèse d'images

En synthèse d'images on considère les objets qui réfléchissent, transmettent et émettent la lumière [WW92]. Les six principaux phénomènes résultant de l'interaction lumière/objet c'est-à-dire ceux qui donnent lieu à des couleurs en modifiant ou filtrant la distribution d'énergie de la lumière incidente sont la *réflexion*, la *transmission*, l'*absorption*, la *diffraction*, la *réfraction* et l'*interférence*.

La radiométrie donne :

- la loi de réflectance :  $\theta_1 = \theta'_1$   
avec  $\theta_1$  l'angle de la lumière incidente par rapport à l'horizontale et  $\theta'_1$  celui de la lumière réfléchie.
- la loi de réfraction :  $n_1 \sin \theta_1 = n_2 \sin \theta_2$   
avec  $\theta_2$  l'angle de la lumière transmise dans le second milieu par rapport à l'horizontale,  $n_1$  et  $n_2$  les indices de réfraction des deux milieux. L'indice de réfraction est :  $n_i = \frac{c_0}{c_i}$  avec  $c_0$  la vitesse de la lumière dans le vide et  $c_i$  celle du milieu considéré. On peut aussi définir  $n$  par :  $n = \sqrt{\epsilon_r}$  où  $\epsilon_r$  est la constante diélectrique du milieu.
- le coefficient de réflexion :  $\rho = \frac{\Phi_r}{\Phi_0} = \left( \frac{n_1 - n_2}{n_1 + n_2} \right)^2$ .
- le coefficient d'absorption :  $\alpha$
- le coefficient de transmittance :  $\gamma = 1 - \rho_2 \approx \frac{2n}{n^2 + 1}$

Bien que ces phénomènes aient été modélisés comme extensions en synthèse d'images un intérêt particulier a été porté sur la réflexion. Les modèles de réflexion en synthèse d'images sont en effet motivés pas tant pour imiter la réalité que pour simuler certains aspects de la réalité qui peuvent être calculables en considérant les contraintes des applications et du matériel disponibles. Le modèle de Phong est une imitation empirique de la réflexion spéculaire et diffuse qui approxime plus que les modèles théoriques précis mais qui produit des résultats visuels acceptables pour un bon nombre d'applications.

Les modèles de réflexion de premier ordre (ou locale) sont des outils utilisés en synthèse d'images pour simuler l'interaction lumière/objet. Ces modèles appartiennent à la catégo-

rie de ceux pour lesquels seulement la réflexion directe ou première de la lumière par une surface due à une source lumineuse est considérée. Cependant ces modèles de premier ordre sont limités en soi dans leur aptitude à simuler l'illumination dans un environnement complet. En effet ils traitent les objets comme s'ils volaient librement dans l'espace et interagissaient seulement avec une source de lumière. Par la suite des modèles dits globaux ont permis de mieux simuler la réflexion grâce aux techniques de lancer de rayons et de radiosité. Le lancer de rayons sélectionne un aspect particulier de l'interaction objet/lumière qui est la réflexion spéculaire et la transmission l'approxime et exclut les autres considérations. La méthode de radiosité qui est en quelque sorte l'inverse du lancer de rayons favorise l'interaction de surfaces réfléchissantes diffuses à l'exclusion de la réflexion spéculaire.

Pour comprendre les limitations et les approximations inhérentes aux modèles de réflexion en synthèse d'images [SH81] un fondement théorique est maintenant présenté.

## A.2 Théorie : Intensité et Énergie

L'effet relatif à la couleur d'un objet est décrit par une intensité directionnelle. En effet la radiation qui émane dans une direction donnée depuis une surface est définie par la somme des intensités réfléchie et émise. Dans des modèles simples de synthèse d'images la réflexion est traitée séparément de l'émission. Les objets sont des entités auxquelles sont appliqués des modèles de réflexion et les sources lumineuses généralement approximées à des sources lumineuses ponctuelles sont des entités séparées qui fournissent l'énergie initiale. Cependant dans la méthode de radiosité toutes les surfaces qu'elles soient émettrices ou réfléchissantes sont intégrées dans le même modèle. De telles surfaces obéissent à l'équation générale de l'intensité réfléchie et émise :

$$I(\theta_r, \phi_r) = I_{\text{émise}}(\theta_r, \phi_r) + I_{\text{réfléchie}}(\theta_r, \phi_r)$$

où  $(\theta_r, \phi_r)$  est une direction sortante spécifiée par ces deux angles (cf. Figure A.1).

On s'intéresse à la réflexion et on définit la réflectivité comme le nombre de passages des rayons lumineux. On considère le rapport qui lie l'énergie entrante à l'intensité sortante :

$$\rho''(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{I_r(\theta_r, \phi_r, \theta_i, \phi_i)}{E_i(\theta_i, \phi_i)}$$

où  $\rho''$  est le coefficient de réflectivité spectrale (le double '' indique qu'il dépend à la fois des directions entrante et sortante)  $I_r$  l'intensité réfléchie dans la direction sortante  $E_i$  l'énergie dans la direction entrante  $(\theta_r, \phi_r)$  la direction sortante et  $(\theta_i, \phi_i)$  la direction entrante (cf. Figure A.1). En synthèse d'images ces directions sont habituellement données par des vecteurs unités  $L$  et  $V$   $L$  étant le vecteur de direction de la lumière  $(\theta_i, \phi_i)$   $V$  le vecteur de vue ou vecteur de direction sortante digne d'intérêt  $(\theta_r, \phi_r)$ .

On remarque que  $(\theta_r, \phi_r)$  n'est pas la direction miroir. L'utilisation du mot "réfléchie" est quelque peu déroutante puisqu'il arrive que ce mot soit employé pour indiquer la

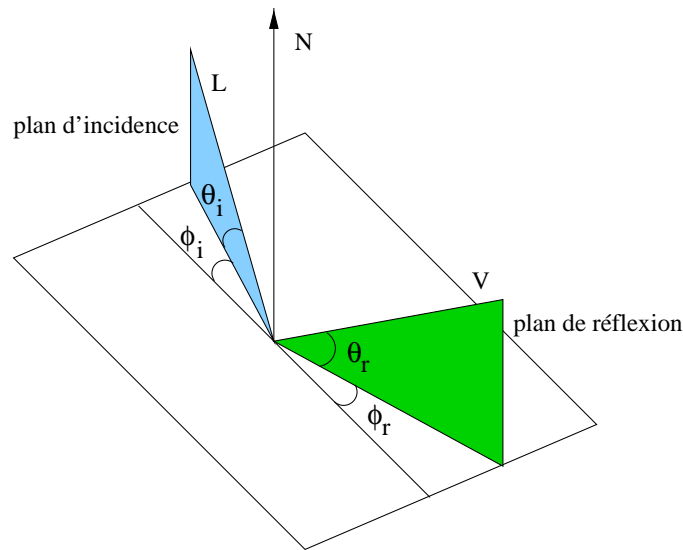


FIG. A.1: Conventions utilisées en réflexion.

direction miroir. Nous emploierons ce mot “réfléchi” pour désigner la direction de vue  $\Gamma$  et le mot “miroir” pour désigner celle dans laquelle la lumière est réfléchie de manière spéculaire.

L'énergie entrante  $E_i(\theta_i, \phi_i)$  est liée à l'intensité entrante  $I_i(\theta_i, \phi_i)$  (cf. Figure A.2) par l'équation :

$$E_i(\theta_i, \phi_i) = I_i(\theta_i, \phi_i) \cos \theta_i d\omega_i$$

avec  $d\omega_i$  l'angle solide dans lequel l'énergie est contenue. Un angle solide  $\omega$  d'un élément de surface  $\Gamma$  pour un point de vue donné est défini par l'aire de l'élément de surface divisée par le carré de la distance avec le point.

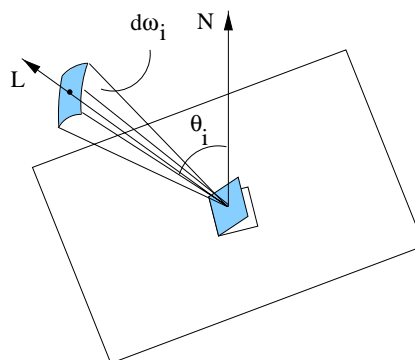


FIG. A.2: Intensité mesurée par rapport à la normale de la surface projetée dans sa direction.

### A.2.1 Réflectivité spectrale bidirectionnelle

L'énergie réfléchie depuis un point sur une surface dépend de l'angle de l'énergie incidente et de l'angle de vue pour l'énergie réfléchie. Ainsi on peut définir une réflectivité spectrale bidirectionnelle. La réflectivité est dite spectrale car on reconnaît que dans tous les cas pratiques cette réflectivité est fonction de la longueur d'onde.

On considère la radiation spectrale incidente dans une direction entrante particulière  $(\theta_i, \phi_i)$ . Une partie de cette radiation est réfléchie dans la direction  $(\theta_r, \phi_r)$  et on peut se référer à une intensité réfléchie par :

$$I_{\lambda,r}(\lambda, \theta_r, \phi_r, \theta_i, \phi_i)$$

qui est une fonction de la longueur d'onde  $\lambda$  et des directions réfléchie et incidente.

Il a été montré dans [SH81] que la réflectivité spectrale bidirectionnelle est réciproque c'est-à-dire que  $\rho''$  est symétrique par rapport aux angles de réflexion et d'incidence :

$$\rho''(\theta_i, \phi_i, \theta_r, \phi_r) = \rho''(\theta_r, \phi_r, \theta_i, \phi_i)$$

Ainsi on peut placer la source de lumière au point de vue et le point de vue à la source de lumière et obtenir le même résultat.

Les relations de réciprocity sont utilisées dans de nombreux algorithmes de synthèse d'images. Par exemple en lancer de rayons les rayons sont lancés du point de vue dans la scène ce qui correspond à la direction inverse de propagation de la lumière (cf. Figure A.3). Ce schéma généralement mis en œuvre pour la réflexion et la transmission spéculaires suppose que le coefficient de réflexion soit symétrique par rapport aux angles incident et réfléchi.

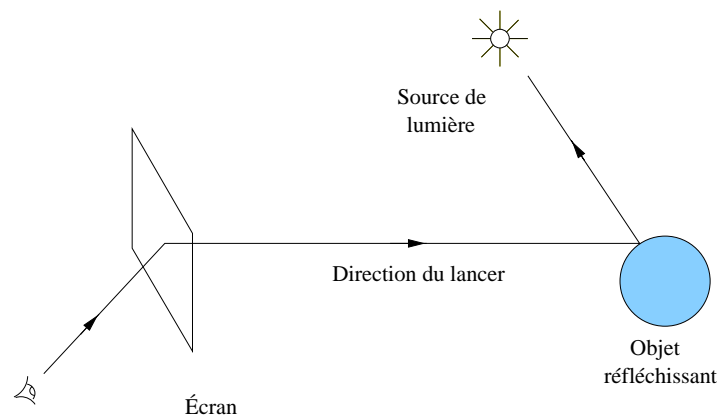


FIG. A.3: Réciprocité en lancer de rayons : les rayons sont tracés dans le sens opposé au sens de propagation de la lumière.

### A.2.2 Surfaces diffuses et spéculaires

Les surfaces parfaitement diffuses et parfaitement spéculaires sont deux cas particuliers des versions simplifiées de surfaces. Pour une surface réfléchissante diffuse une intensité entrante est réfléchi de manière égale dans toutes les directions (cf. Figure A.4 (a)). Une surface qui exhibe des propriétés de miroir ou spéculaires restitue l'intensité entrante exclusivement dans la direction "miroir" (cf. Figure A.4 (b)). Ces extrêmes sont particulièrement intéressants en synthèse d'images car les modèles de réflexion empiriques les plus communs représentent une surface comme une combinaison d'une surface parfaitement diffuse et d'une surface parfaitement spéculaire (cf. paragraphe A.3).

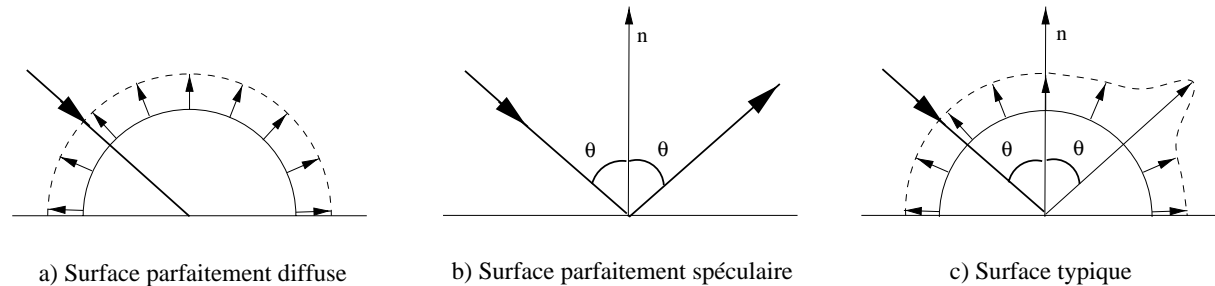


FIG. A.4: (a) Une surface parfaitement diffuse propage l'intensité réfléchi de manière égale dans toutes les directions. (b) Une surface ayant une réflectivité spéculaire ou de type miroir tend à produire une composante plus grande dans la direction "miroir".

Une surface parfaitement diffuse réfléchit la radiation de manière égale dans toutes les directions et apparaît de brillance égale depuis n'importe quel point de vue. Cependant la quantité d'énergie réfléchi peut varier en fonction de l'angle d'incidence  $(\theta_i, \phi_i)$ . Ainsi la réflectivité spectrale bidirectionnelle est indépendante de  $(\theta_r, \phi_r)$  elle ne dépend plus que de  $\lambda$  et  $(\theta_i, \phi_i)$ .

Une surface parfaitement spéculaire (un miroir) réfléchit la lumière dans une seule direction si la radiation incidente est confinée dans une seule direction. L'angle que fait la radiation réfléchi avec la normale de la surface est égal à l'angle de la radiation incidente par rapport à la normale de la surface et les deux rayons sont dans le même plan :

$$\theta_r = \theta_i \text{ et } \phi_r = \phi_i + \pi$$

Si le miroir est parfait il n'y a aucune radiation sortante dans les autres directions et la réflectivité spectrale bidirectionnelle est seulement fonction de la direction de la radiation entrante et de la longueur d'onde.

## A.3 Application : Le modèle de réflexion de Phong

Ce modèle de réflexion classique le plus courant en synthèse d'images divise la réflectivité en une composante diffuse et une composante spéculaire. La réflectivité spec-

trale bidirectionnelle est approximée par :

$$\rho''_{\lambda}(\lambda, \theta_r, \phi_r, \theta_i, \phi_i) = k_{\text{diffus}} + k_{\text{spéculaire}} \cos^n \Phi$$

avec  $k_{\text{diffus}}$  la fraction d'énergie réfléchie de manière diffuse  $k_{\text{spéculaire}}$  la fraction d'énergie réfléchie de manière spéculaire et  $\Phi$  l'angle entre la direction miroir  $R$  et la direction de vue  $V$  (cf. Figure A.5).

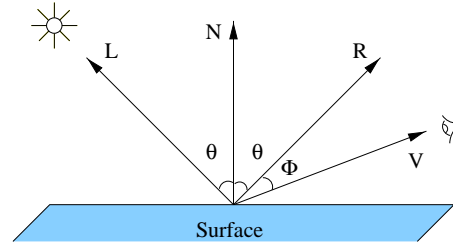


FIG. A.5: *Modèle de réflexion de Phong.*

La plupart des mises en œuvre réduisent la réflectivité bidirectionnelle à une réflectivité qui dépend seulement de l'angle de vue :

$$\rho''_{\lambda, \text{phong}}(\lambda, \theta_r, \phi_r, \theta_i, \phi_i) = \rho'_{\lambda}(\lambda, \Phi)$$

où le matériau est supposé isotrope c'est-à-dire dont les propriétés sont identiques dans toutes les directions. Bien qu'il n'y ait rien pour empêcher une dépendance anisotrope de la réflectivité sur les deux angles sortants le modèle de Phong est le plus souvent mis en œuvre comme décrit ci-dessus. La propagation empirique du reflet dans la direction miroir était l'innovation importante apportée par Phong donnant ainsi une méthode peu coûteuse et efficace pour calculer la géométrie du reflet spéculaire.

La mise en œuvre usuelle du modèle de Phong est donnée en termes de vecteurs unitaires associés à la géométrie du point de considération. Les termes ambiant diffus et spéculaire étant abrégés à leurs initiales respectivement  $a$  et  $s$  on a :

$$I_{\lambda, r}(\lambda, \Phi) = I_{\lambda, a} \cdot k_a(\lambda) + I_{\lambda, i} \cdot (k_d(\lambda)(L \cdot N) + k_s(R \cdot V)^n) \quad (\text{A.1})$$

avec  $R$  le vecteur réflexion unitaire  $V$  le vecteur unitaire du point de la surface au point de vue  $n$  un indice contrôlant l'étroitesse du reflet spéculaire.

On remarque que  $k_s$  est indépendant de la longueur d'onde donnant au reflet la couleur de la source. La Figure A.6 montre la variation de l'intensité lumineuse en un point d'une surface calculée avec cette équation. La variation d'intensité est présentée comme un profil qui est fonction du vecteur de vue  $V$  alors qu'il tourne dans toutes les orientations possibles dans le plan de la feuille autour du point d'intérêt. Le demi-cercle représente la contribution constante diffuse et ambiante et la contribution spéculaire est donnée pour différentes valeurs de  $n$ .

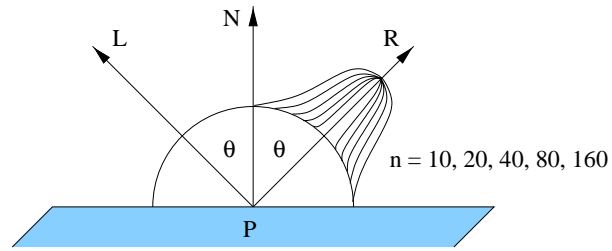


FIG. A.6: Intensité lumineuse en un point  $P$ , fonction de l'orientation du vecteur de vue  $V$ .

Si plusieurs sources sont utilisées alors l'équation (A.1) devient :

$$I_{\lambda,r}(\lambda, \Phi) = I_{\lambda,a}k_a(\lambda) + \sum_{m=1}^M I_{\lambda,i,m}(\lambda)(k_d(\lambda)(L_m \cdot N) + k_s(R_m V)^n)$$

avec  $M$  le nombre de sources lumineuses séparées  $L_m$  le vecteur du point à la  $m^{\text{ième}}$  source lumineuse  $R_m$  le vecteur réflexion associé à la  $m^{\text{ième}}$  source lumineuse.





## Annexe B

# Perception et reconnaissance de formes

### B.1 Introduction

La perception et la reconnaissance de formes sont deux domaines de recherche liés à la robotique (mais pas uniquement) et font l'objet de nombreuses publications dans des conférences et des journaux tels que *International Conference on Advanced Robotics*, *IEEE Transactions on Robotics and Automation*, *International Journal of Robotics Research* etc. La compréhension qu'un robot peut avoir de son environnement est généralement le résultat de deux étapes distinctes : l'extraction grâce aux capteurs de l'information sur un objet puis la manipulation de ces informations qui conduit à une décision par exemple la localisation ou la reconnaissance d'objets [BFM88]. De nombreux capteurs fournissent directement une information géométrique comme les coordonnées de points du bord de l'objet. Afin de traiter les problèmes de décision des algorithmes géométriques sont utilisés pour reconnaître les formes : enveloppe convexe, diamètre d'un ensemble, squelette d'une forme, décomposition en parties convexes, recherche des plus proches voisins [Tou80].

Nous abordons trois problèmes directement liés à la robotique : la *fusion multi-capteurs*, la *reconstruction de formes* à partir d'informations partielles fournies par les capteurs et la *reconnaissance d'objets*. Nous présentons ces problèmes ainsi que quelques méthodes proposées en robotique pour les traiter. Nous n'utilisons certes pas ces méthodes car notre travail présenté dans le chapitre 6 est simplifié. Nous nous sommes intéressés en effet uniquement au profil de l'environnement (cf. paragraphe 6.3.2). Cependant il nous est apparu intéressant de savoir comment la fonction de reconnaissance pouvait être traitée afin d'avoir des idées concernant sa réalisation. De plus ces méthodes peuvent être utilisées pour les travaux futurs concernant une reconnaissance plus complète de l'environnement.

#### B.1.1 Fusion multi-capteurs

À partir de la connaissance de son environnement et des buts qu'il a à atteindre le robot élabore des lois de contrôle et prend des décisions d'actions. La qualité de la

connaissance de l'environnement conditionne la réussite de la mission du robot. Dans de nombreux cas, les données provenant directement des capteurs sont peu précises, peu fiables, parfois manquantes. Ceci met en péril la bonne réalisation de la mission, par un risque de collision, une réaction brutale à l'apparition de faux obstacles, un comportement erratique et des messages d'alarme incohérents.

### a) Objectif de la fusion

La fusion de données provenant de différents capteurs permet d'obtenir une information plus riche, plus précise et plus fiable. L'information résultante reflète non seulement les données fournies par chaque capteur mais également de l'information qui n'aurait pu être inférée par aucune des sources prises séparément. On utilise pour cela les propriétés de redondance et de complémentarité des capteurs. Ainsi, la fusion de données permet d'accroître la connaissance de l'état du robot dans son environnement et donc d'améliorer les performances du contrôle du robot et la qualité de son diagnostic.

### b) Systèmes d'intégration et de fusion multi-capteurs

Les dispositifs logiciels et matériels qui acquièrent, traitent et intègrent des données provenant de différentes sources d'informations sont communément appelés *systèmes d'intégration et de fusion multi-capteurs*. On distingue l'intégration de la fusion multi-capteurs de manière à séparer les aspects matériels, au point de vue architecture et contrôle de l'intégration multi-capteurs, des concepts plus mathématiques rencontrés dans l'étape de fusion effective d'informations.

L'intégration multi-capteurs concerne l'architecture du système, les relations liant les différents modules et les actions sur le système de perception. On peut citer les types d'architecture les plus répandues : les systèmes hiérarchiques, les systèmes coopératifs, la supervision centralisée et les systèmes multi-agents.

### c) Réalisation de la fusion

La fusion multi-capteurs correspond à toutes les étapes de l'intégration multi-capteurs au cours desquelles les informations provenant des différents capteurs sont effectivement combinées. L'information à combiner est issue soit de plusieurs capteurs durant une seule période d'échantillonnage, soit d'un seul capteur durant plusieurs périodes d'échantillonnage, soit enfin de plusieurs capteurs durant plusieurs périodes d'échantillonnage. Cette fusion d'informations peut être effectuée au niveau signal, primitive ou symbole :

- Au niveau signal, la fusion est la combinaison des signaux issus d'un groupe de capteurs en vue d'obtenir un signal de même nature mais d'une qualité supérieure. La procédure de fusion est souvent considérée comme une étape d'estimation. Comparée aux autres niveaux de fusion multi-capteurs, la fusion niveau signal exige le plus grand soin pour régler la concordance temporelle et spatiale entre les données sensorielles.

- Au niveau primitive (polygone  $\Gamma$  polyèdre ...)  $\Gamma$  la fusion est utilisée pour augmenter la vraisemblance qu'une primitive extraite des données fournies par un capteur corresponde réellement à un aspect important de l'environnement ou pour créer de nouvelles primitives composites. Une primitive simple est créée en attachant une signification sémantique aux résultats de traitement des données sensorielles (spatiales et/ou temporelles) tandis qu'une primitive composite est obtenue par combinaison des primitives déjà existantes.
- Au niveau symbole (sol  $\Gamma$  mur  $\Gamma$  table ...)  $\Gamma$  le niveau d'abstraction le plus élevé  $\Gamma$  la fusion permet de combiner les informations provenant de plusieurs capteurs. Elle offre un moyen de combiner des informations hétérogènes ou des informations correspondant à des régions distinctes de l'environnement. Les symboles utilisés sont soit issus du traitement des informations fournies par les capteurs  $\Gamma$  soit le résultat d'un processus de raisonnement symbolique. Ce processus utilise des connaissances a priori sur l'environnement  $\Gamma$  ou alors des sources d'information extérieures au système  $\Gamma$  comme l'avis d'expert informant sur la présence d'un certain type de cible dans l'environnement. Un symbole représente une prise de décision sur un aspect particulier de l'environnement. Cette décision provient généralement de la mise en correspondance des primitives dérivées des informations sensorielles avec un modèle.

### B.1.2 Reconstruction de formes

Le but de la reconstruction est de restituer une structure topologique globale de l'objet à partir des informations partielles délivrées par les systèmes de fusion multi-capteurs. Typiquement  $\Gamma$  l'information sur un objet ou sur un environnement n'est que partielle. En effet  $\Gamma$  un capteur mesure un ensemble de points  $\Gamma$  non obligatoirement ordonnés  $\Gamma$  et qui peuvent appartenir à un ou plusieurs objets. Ainsi  $\Gamma$  pour pouvoir construire un modèle complet de l'objet  $\Gamma$  il faut le regarder depuis différents points de vue ou avec différents capteurs  $\Gamma$  et combiner les informations partielles ainsi recueillies en une structure cohérente représentant l'objet (cf. Figure B.1).

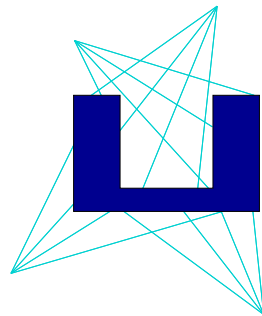


FIG. B.1: *Reconstruction de formes à partir de différents points de vue.*

Dans les cas les plus simples  $\Gamma$  il est possible d'être maître de la façon dont les points sont mesurés. Aussi  $\Gamma$  par un choix judicieux des mesures  $\Gamma$  on peut reconstruire sans ambiguïté la forme de l'objet [CY87].

Dans le cas d'un robot mobile muni d'un télémètre laser  $\Gamma$  qui se déplace autour d'un objet et mesure des points sur le bord de cet objet  $\Gamma$  l'information mesurée est constituée de l'ensemble des coordonnées des points mesurés  $\Gamma$  ainsi que de l'ensemble des rayons. Un rayon est un segment reliant un point mesuré à la position qu'occupait le télémètre au moment de la mesure (cf. Figure B.2). Pour des raisons physiques évidentes  $\Gamma$  ces rayons ne traversent pas l'objet. Ce type d'information  $\Gamma$  implicite mais souvent ignoré dans les traitements ultérieurs  $\Gamma$  est source d'aide. Étant donné  $n$  points dans le plan  $\Gamma$  si l'on cherche à les relier pour construire une approximation polygonale de l'objet mesuré  $\Gamma$  il existe de nombreuses façons de le faire. En revanche  $\Gamma$  il n'existe qu'un seul polygone dont les sommets sont les  $n$  points et qui n'intersecte pas les rayons [ABY86].

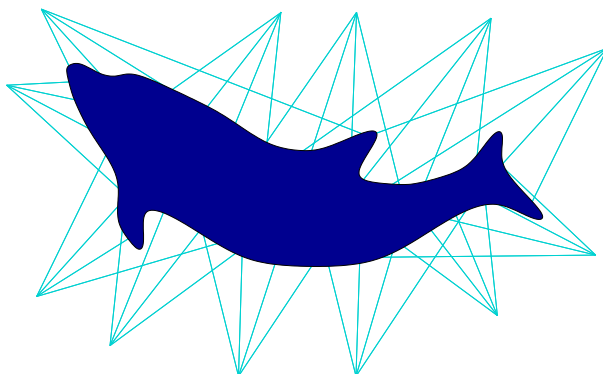


FIG. B.2: *Reconstruction de formes à partir de l'ensemble des rayons lancés.*

Dans le cas où un objet ou un ensemble d'objets est mesuré dans une succession de plans parallèles  $\Gamma$  la reconstruction peut se faire à partir de cette séquence de sections planes  $\Gamma$  chaque section étant un polygone. Une méthode consiste à relier de manière optimale les polygones de deux sections successives par une bande de triangles  $\Gamma$  en s'appuyant sur les sommets de ces polygones [FKU77] (cf. Figure B.3).

Le cas le plus défavorable est celui où aucune information autre que les coordonnées des points mesurés n'est disponible. En général  $\Gamma$  le problème de relier ces points désordonnés par un polygone simple n'admet pas de solution unique. Il faut alors contraindre le problème et/ou définir une méthode qui converge vers la bonne solution quand le nombre de points mesurés augmente [BFM88].

### B.1.3 Reconnaissance d'objets

L'étape suivant la reconstruction des formes détectées  $\Gamma$  est la reconnaissance de ces formes. Le problème de reconnaissance de formes est lié au problème d'inspection : l'objet observé que l'on cherche à reconnaître peut être identique au modèle que l'on en a  $\Gamma$  ou n'en être qu'une copie approximative. Par ailleurs  $\Gamma$  l'objet peut être entièrement ou seulement partiellement vu  $\Gamma$  si les objets sont disposés n'importe où  $\Gamma$  ou encore si les conditions d'éclairage sont mauvaises.

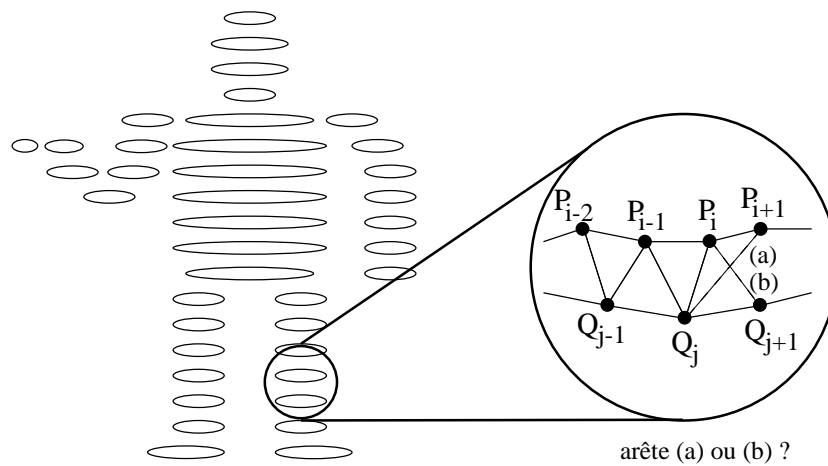


FIG. B.3: *Reconstruction de formes à partir de sections planes, avec triangulation entre les points de sections adjacentes.*

Le problème le plus simple est celui où l'objet et le modèle sont deux polygones simples  $\Gamma$  représentant parfaitement les objets correspondants. Identifier l'objet au modèle revient donc à savoir si  $\Gamma$  à une transformation affine près  $\Gamma$  c'est-à-dire à une combinaison de translation  $\Gamma$  rotation et homothétie près  $\Gamma$  l'objet et le modèle sont identiques ou non [Man76  $\Gamma$  AT78].

Le problème précédent peut être généralisé au cas des polyèdres. Les sommets et les arêtes d'un polyèdre constituent un graphe. Savoir si deux polyèdres sont congruents  $\Gamma$  c'est-à-dire identiques à une translation et une rotation près  $\Gamma$  revient à un problème d'isomorphisme de graphes. Ce problème est difficile en général  $\Gamma$  mais des algorithmes efficaces existent dans le cas de graphes planaires [Sug84].

Dans le cas où les mesures sont bruitées et/ou l'objet n'est pas exactement conforme au modèle que l'on en a  $\Gamma$  les méthodes précédentes ne s'appliquent pas. On peut alors utiliser des techniques développées comme la reconnaissance à partir de points de mesure. Soit  $P$  un polygone ou un polyèdre de  $n$  sommets et  $M$  un ensemble de  $m$  points de mesure. Le problème de reconnaissance consiste alors à trouver si une translation et une rotation peuvent amener  $M$  sur  $P$ . Le problème suivant est un problème de localisation : il faut décrire l'ensemble des transformations amenant  $M$  sur  $P$ .

## B.2 Conclusion

Le robot doit évoluer dans l'environnement en toute sécurité. Il lui faut donc une connaissance suffisante de son environnement local  $\Gamma$  au sens où tous les obstacles avec lesquels il peut entrer en collision doivent être détectés et évités. Un seul capteur ne suffit pas à remplir cette tâche de détection. En effet  $\Gamma$  la zone couverte par un seul capteur est généralement insuffisante pour garantir la détection de tout obstacle avec le robot entier. Aussi  $\Gamma$  il est très courant d'utiliser plusieurs capteurs sur les robots réels.

La fusion de données de plusieurs capteurs permet alors de combiner des informations provenant de sources diverses tout en prenant en compte le caractère changeant d'un environnement dynamique. Cette combinaison des informations obtenues par les divers capteurs à des instants différents ou non est analysée afin de reconstruire les formes des objets détectés. Une fois ces formes reconstruites il existe différents types de reconnaissance de formes. Cette reconnaissance permet de mettre à jour les modèles symboliques de l'environnement interprété. Le robot peut alors se référer à tout moment à ces modèles symboliques afin de savoir de quoi est constitué l'environnement en un endroit donné lui permettant ainsi de s'adapter au mieux à son environnement.

## Annexe C

# Tests des configurations de capteurs sur le robot

### C.1 Positions, orientations et portées des capteurs

Nous avons réalisé les 360 simulations annoncées au paragraphe 7.2.3. Cependant pour des raisons de clarté nous ne donnons que les résultats les plus représentatifs les autres résultats étant suffisamment analogues pour n'être que cités.

Ces simulations ont consisté à placer un capteur soit sur la jambe soit sur la cuisse à une des 3 positions sur le support (au quart milieu ou trois quarts) selon l'une des 2 orientations pour la jambe ( $90^\circ$  ou  $80^\circ$ ) et l'une des 3 orientations pour la cuisse ( $90^\circ$ ,  $75^\circ$  ou  $60^\circ$ ) avec une des 3 portées ( $1.5m$ ,  $2m$  ou  $3m$ ). Puis le robot était placé à une des 4 distances ( $2m$ ,  $2.25m$ ,  $2.5m$  ou  $2.75m$ ) soit d'un obstacle de  $30cm$  de hauteur soit d'un trou de  $30cm$  de hauteur. Ces simulations étaient arrêtées dès que le robot était à environ  $1m$  de l'obstacle ou du trou. Nous avons alors observé les courbes de distances et les profils de l'environnement obtenus.

Auparavant nous souhaitons apporter une simplification. Au cours de ces simulations nous nous sommes aperçus que les trois positions du capteur sur son support donnaient des résultats quasi identiques pour chaque situation particulière. La seule différence était qu'au pire le capteur aux trois quarts du support détectait un pas de temps après le capteur situé au milieu du support qui lui-même détectait un pas de temps après le capteur situé au quart du support. Nous ne présentons donc les résultats que pour les capteurs situés au milieu du support car c'est cet emplacement que nous préférons.

Nous détaillons dans le prochain paragraphe les résultats obtenus sur la jambe dans le cas de l'obstacle puis dans le cas du trou et dans le paragraphe suivant les résultats obtenus sur la cuisse dans le cas de l'obstacle puis dans le cas du trou.



### C.1.1 Jambe

#### a) Obstacle

- Le capteur orienté à  $90^\circ$  et avec une portée de  $1.5m$  détecte peu l'obstacle et assez tard selon la distance  $d$  de départ du robot (cf. Figure C.1 (b) pour  $d = 2.25m$ ).
- Le capteur orienté à  $90^\circ$  et avec une portée de  $2m$  détecte mieux l'obstacle quelque soit la distance de départ du robot (cf. Figure C.2 (b) pour  $d = 2.75m$ ).
- Un profil d'environnement équivalent est obtenu avec le capteur orienté à  $90^\circ$  et avec une portée de  $3m$  mais avec une détection de l'obstacle plus précoce.
- Les résultats obtenus avec le capteur orienté à  $80^\circ$  diffèrent très peu de ceux obtenus avec le capteur orienté à  $90^\circ$  (cf. Figure C.3 (b) pour  $d = 2.75m$ ).
- Finalement pour la détection de l'obstacle nous préférons le capteur situé au milieu du support dont la portée est  $2m$  ou  $3m$  quelque soit son orientation.

#### b) Trou

Nous donnons directement les résultats pour les capteurs de portée supérieure à  $1.5m$ .

- Le capteur orienté à  $90^\circ$  et avec une portée de  $2m$  ne détecte pas le trou (cf. Figure C.4 (b) pour  $d = 2.75m$ ).
- Le capteur orienté à  $90^\circ$  et avec une portée de  $3m$  détecte un point légèrement en dessous du sol (cf. Figure C.5 (b) pour  $d = 2.75m$ ) mais ce résultat reste insuffisant.
- Des résultats similaires sont obtenus par le capteur orienté à  $80^\circ$  (cf. Figure C.6 (b) pour  $d = 2.75m$ ).
- Finalement pour la détection du trou les capteurs sur la jambe se trouvent insuffisants. Il faudra donc faire assurer cette détection par les capteurs de la cuisse.

#### c) Choix

- Pour la jambe nous choisissons le capteur situé au milieu du support orienté de  $90^\circ$  et dont la portée est  $2m$ .

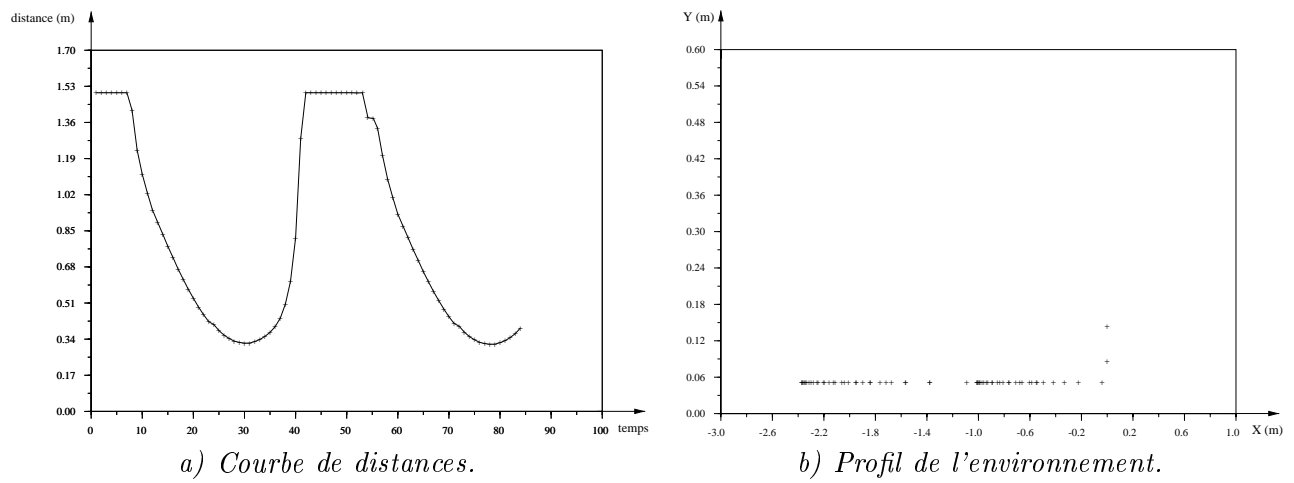


FIG. C.1: Capteur au milieu du support, avec une orientation de  $90^\circ$ , et une portée de 1.5m.

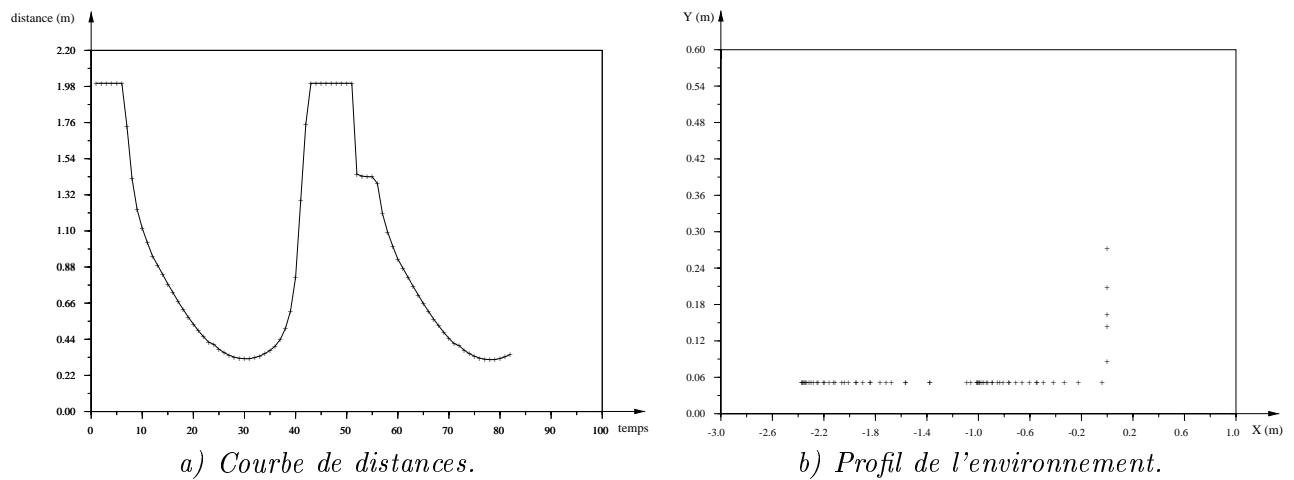


FIG. C.2: Capteur au milieu du support, avec une orientation de  $90^\circ$ , et une portée de 2m.

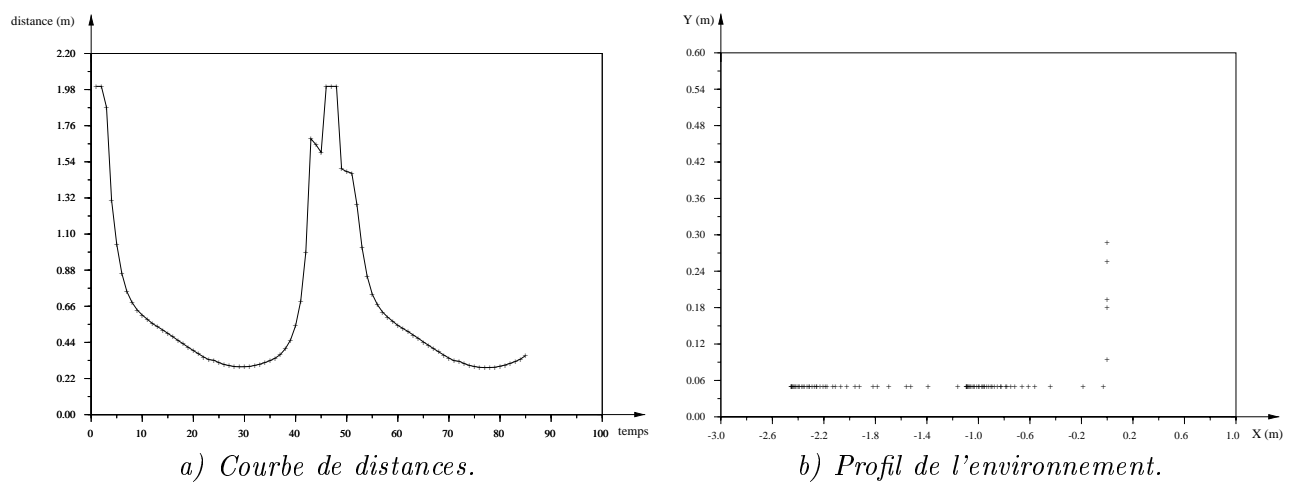


FIG. C.3: Capteur au milieu du support, avec une orientation de  $80^\circ$ , et une portée de 2m.

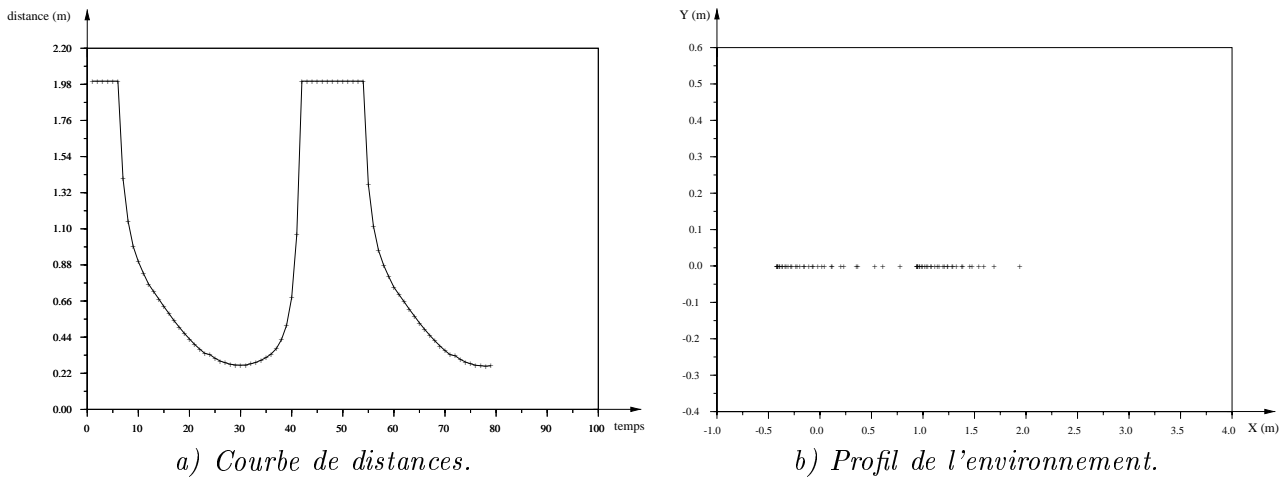


FIG. C.4: Capteur au milieu du support, avec une orientation de  $90^\circ$ , et une portée de 2m.

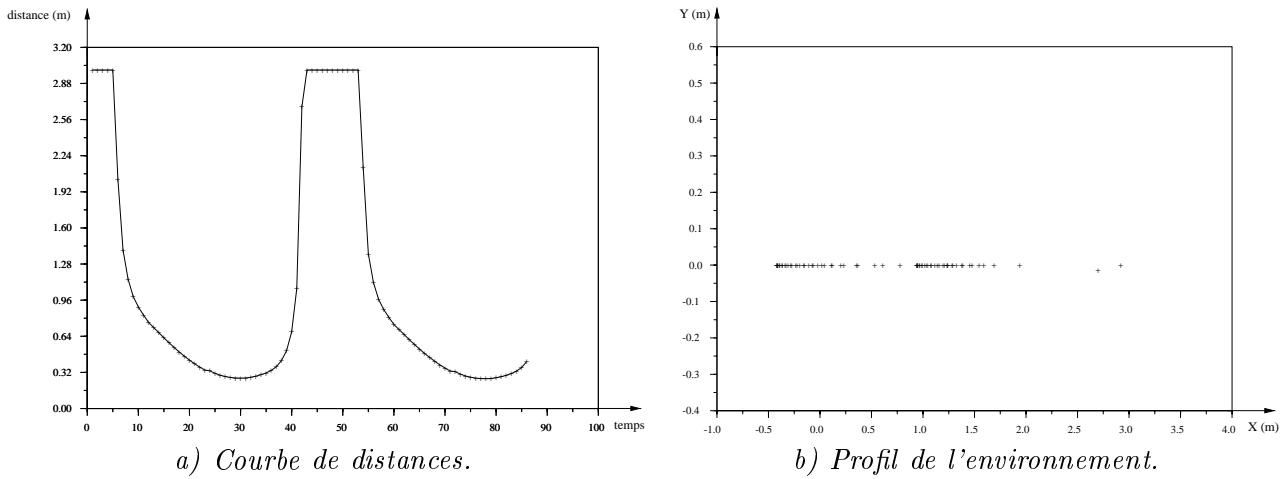


FIG. C.5: Capteur au milieu du support, avec une orientation de  $90^\circ$ , et une portée de 3m.

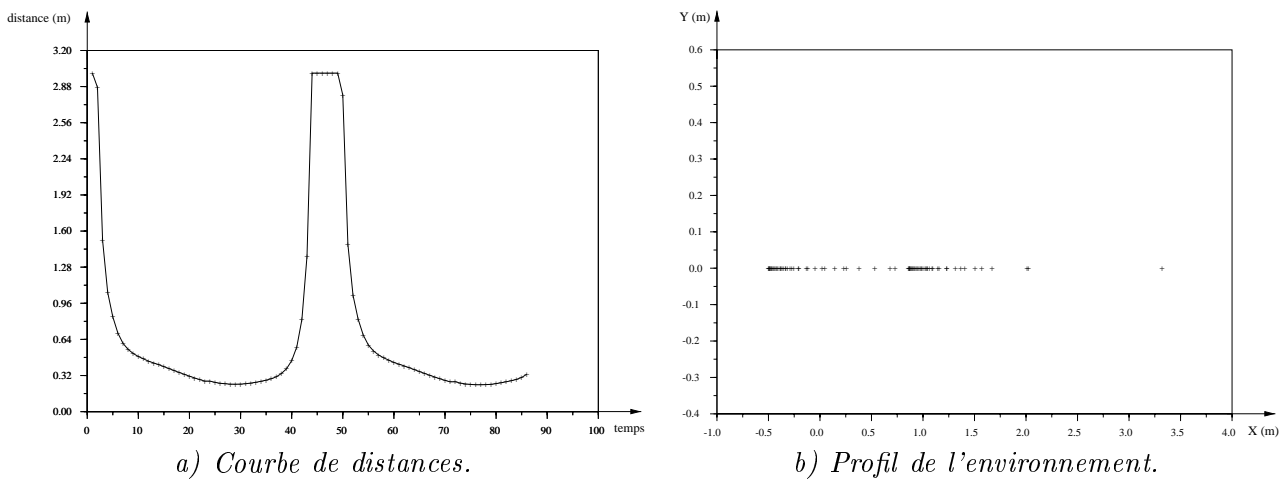


FIG. C.6: Capteur au milieu du support, avec une orientation de  $80^\circ$ , et une portée de 3m.

### C.1.2 Cuisse

#### a) Obstacle

- Le capteur orienté à  $90^\circ$  quelque soit sa portée  $\Gamma$  détecte peu le sol et l'obstacle (cf. Figure C.7 (b) pour  $d = 2.75m$ ).
- Le capteur orienté à  $75^\circ$  quelque soit sa portée  $\Gamma$  détecte mieux le sol (cf. Figure C.8 (b) pour  $d = 2.75m$ ). Cependant  $\Gamma$  avec une portée de  $1.5m$  il détecte l'obstacle un peu tard. Avec une portée de  $2m$  ou  $3m$  l'obstacle est bien détecté et suffisamment tôt.
- Les résultats sont un peu meilleurs avec le capteur orienté à  $60^\circ$  (cf. Figure C.9 (b) pour  $d = 2.75m$ ) : l'environnement est mieux détecté.
- Finalement  $\Gamma$  pour la détection de l'obstacle nous préférons le capteur situé au milieu du support  $\Gamma$  dont l'orientation est  $75^\circ$  ou  $60^\circ$  et la portée est  $2m$  ou  $3m$ .

#### b) Trou

- Le capteur orienté à  $90^\circ$  de portée supérieure ou égale à  $2m$   $\Gamma$  détecte peu le trou (cf. Figure C.10 (b) pour  $d = 2.75m$ ). Le capteur de portée  $1.5m$  ne détecte pas du tout le trou.
- Les résultats sont analogues pour le capteur orienté à  $75^\circ$  (cf. Figure C.11 (b) pour  $d = 2.75m$ ).
- Les résultats sont similaires pour le capteur orienté à  $60^\circ$  et de portée  $1.5m$  ou  $2m$  (cf. Figure C.11 (b) pour  $d = 2.75m$ ).
- Les résultats sont améliorés avec le capteur orienté à  $60^\circ$  et de portée  $3m$  (cf. Figure C.12 (b) pour  $d = 2.75m$ ).
- Finalement  $\Gamma$  pour la détection du trou les meilleurs résultats sont obtenus avec le capteur situé au milieu du support  $\Gamma$  orienté à  $60^\circ$  et de portée  $3m$ .

#### c) Choix

- Pour la cuisse  $\Gamma$  nous choisissons le capteur situé au milieu du support  $\Gamma$  orienté à  $60^\circ$  et de portée  $3m$ .

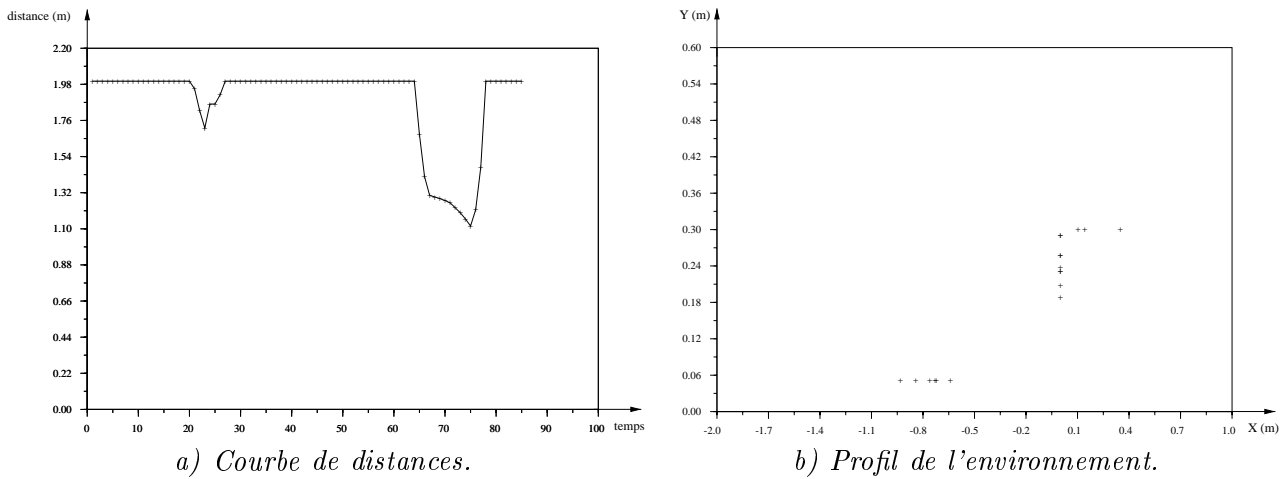


FIG. C.7: Capteur au milieu du support, avec une orientation de  $90^\circ$ , et une portée de 2m.

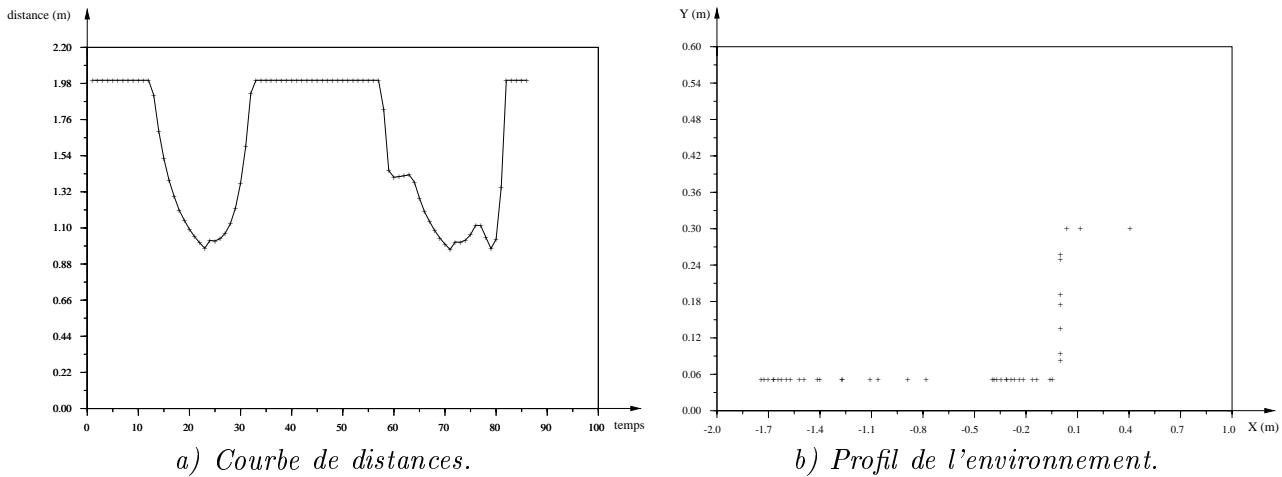


FIG. C.8: Capteur au milieu du support, avec une orientation de  $75^\circ$ , et une portée de 2m.

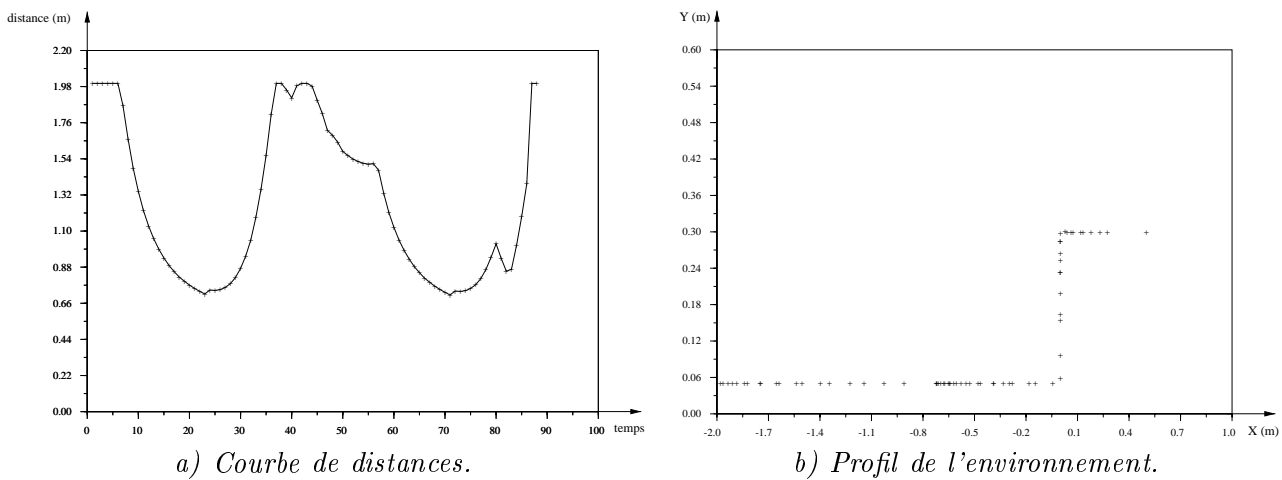


FIG. C.9: Capteur au milieu du support, avec une orientation de  $60^\circ$ , et une portée de 2m.

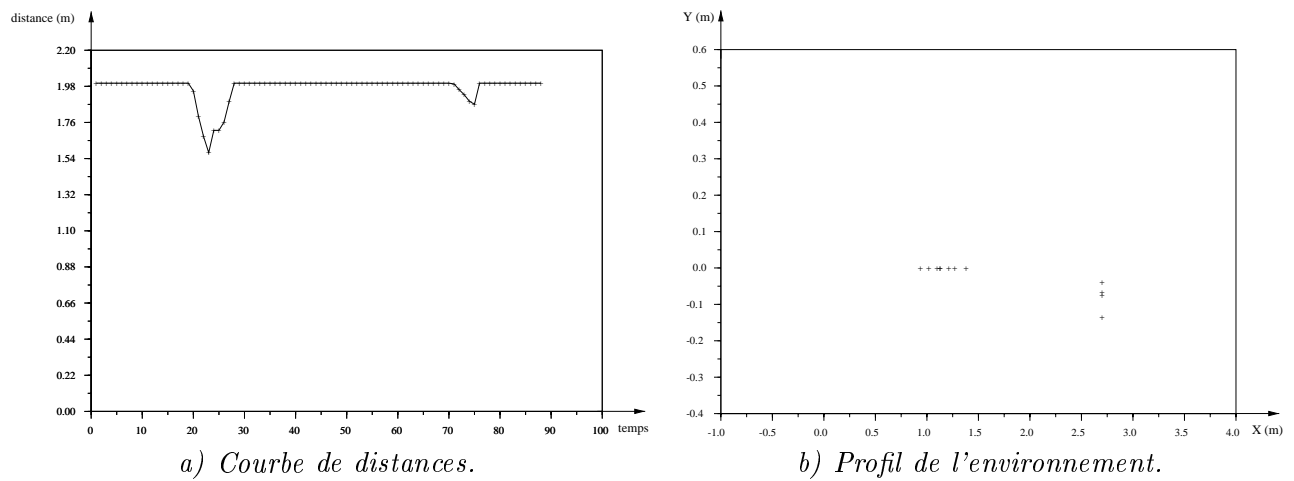


FIG. C.10: Capteur au milieu du support, avec une orientation de  $90^\circ$ , et une portée de 2m.

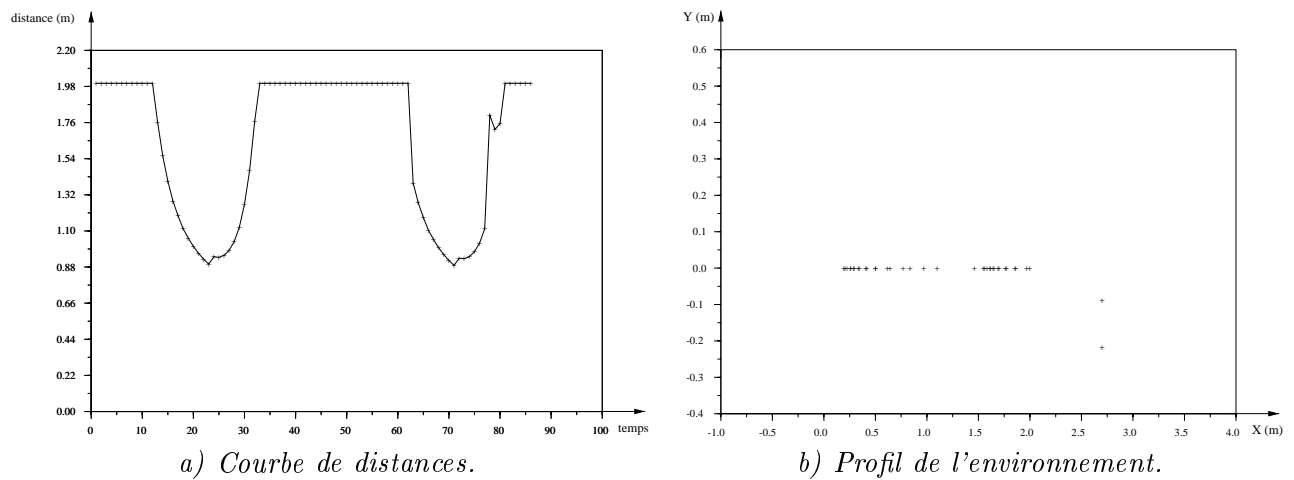


FIG. C.11: Capteur au milieu du support, avec une orientation de  $75^\circ$ , et une portée de 2m.

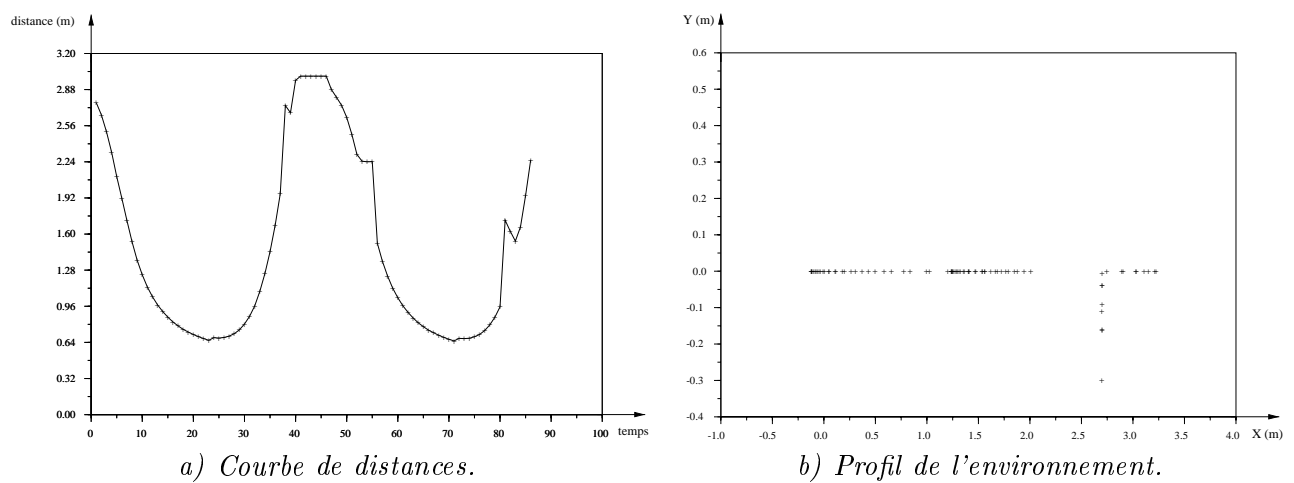


FIG. C.12: Capteur au milieu du support, avec une orientation de  $60^\circ$ , et une portée de 3m.

## C.2 Nombre de capteurs

L'étude des résultats des simulations dans le paragraphe précédent a permis de faire un choix quant à la configuration canonique des capteurs sur le robot. Nous proposons maintenant plusieurs nombres de capteurs avec différentes valeurs d'angle d'ouverture entre rayons extrêmes pour chaque emplacement de cette configuration canonique. Nous donnons les cartes de projection pour les différentes configurations de capteurs obtenues.

### C.2.1 Configuration à 1 capteur par segment

Il s'agit de la configuration canonique (cf. Figure C.13). La carte de projection associée est donnée en Figure C.16. Nous pouvons remarquer que l'espace est insuffisamment recouvert par les capteurs.

### C.2.2 Configurations à 3 capteurs par segment

Nous proposons 3 capteurs par segment pour balayer mieux l'espace devant le robot (cf. Figure C.14). Nous donnons les cartes de projections pour les valeurs d'angle d'ouverture entre rayons extrêmes suivantes :

- 20° (cf. Figure C.17) : le résultat est correct.
- 30° (cf. Figure C.18) : les rayons sont trop écartés et détectent trop au-delà de l'espace du robot et pas assez dans cet espace.
- 40° (cf. Figure C.19) : le résultat est encore aggravé.

### C.2.3 Configurations à 5 capteurs par segment

Nous proposons 5 capteurs par segment pour balayer mieux l'espace devant le robot (cf. Figure C.15). Nous donnons les cartes de projections pour les valeurs d'angle d'ouverture entre rayons extrêmes suivantes :

- 20° (cf. Figure C.20) : le résultat est correct.
- 30° (cf. Figure C.21) : les rayons sont trop écartés et détectent trop au-delà de l'espace du robot et pas assez dans cet espace de manière analogue au résultat pour la configuration à 3 capteurs.
- 40° (cf. Figure C.22) : le résultat est encore aggravé tout comme pour la configuration à 3 capteurs.

### C.2.4 Choix

Finalement les configurations permettant de bien recouvrir l'espace devant le robot sont les configurations :

- à 3 capteurs par segment et d'angle d'ouverture entre rayons extrêmes de 20°
- à 5 capteurs par segment et d'angle d'ouverture entre rayons extrêmes de 20°.

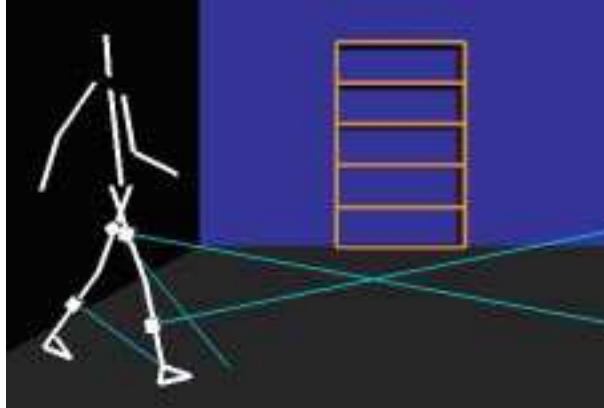


FIG. C.13: *Configuration à 1 capteur par support.*

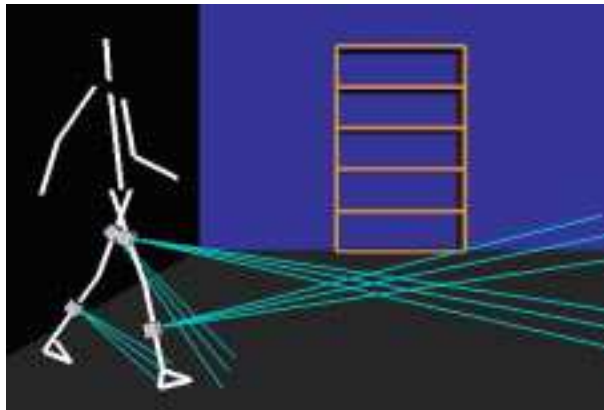


FIG. C.14: *Configuration à 3 capteurs par support.*

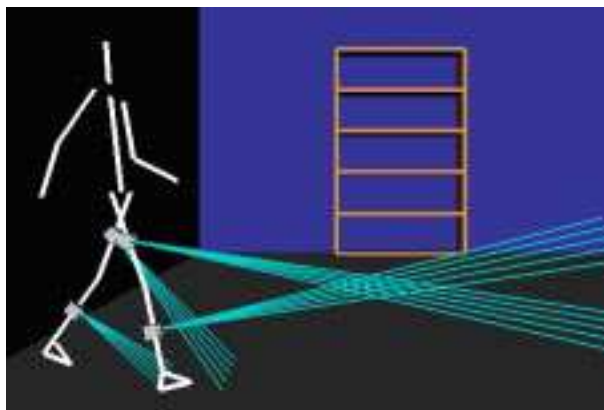


FIG. C.15: *Configuration à 5 capteurs par support.*



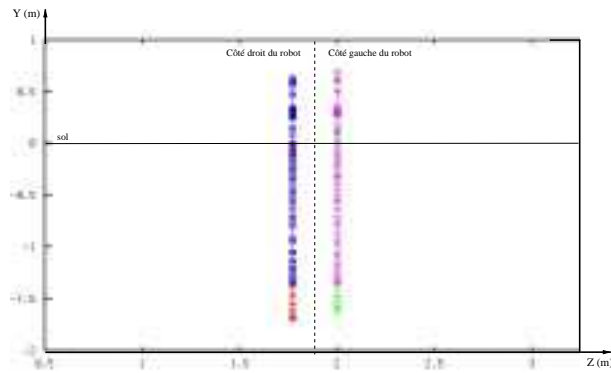


FIG. C.16: *Carte de projection de la configuration de la Figure C.13.*

Les cartes de projection obtenues pour ces deux configurations sont très similaires (cf. Figures C.17 et C.20). Aussi pour des raisons de coût et de poids nous choisissons comme configuration finale la configuration la plus petite c'est-à-dire celle à trois capteurs par segment et d'angle d'ouverture entre rayons extrêmes de  $20^\circ$ .

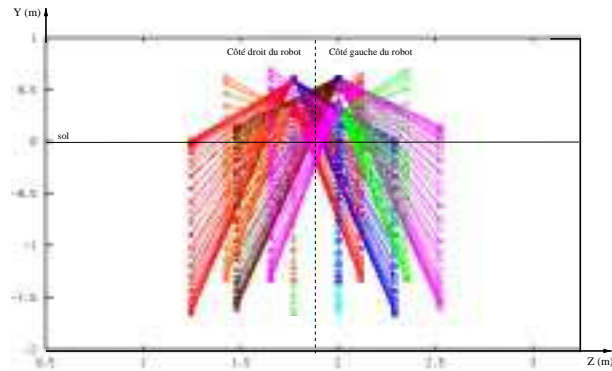


FIG. C.17: Carte de projection de la configuration de la Figure C.14, avec un angle d'ouverture de  $20^\circ$  entre les rayons extrêmes.

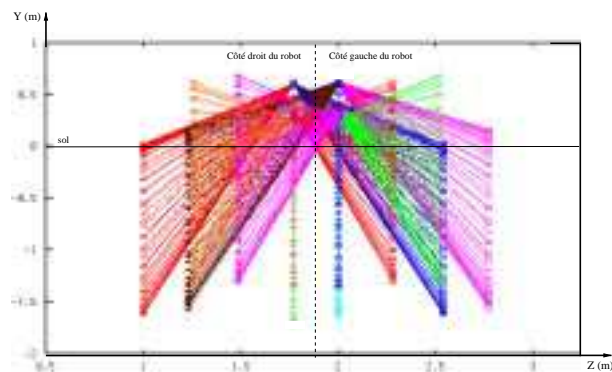


FIG. C.18: Carte de projection de la configuration de la Figure C.14, avec un angle d'ouverture de  $30^\circ$  entre les rayons extrêmes.

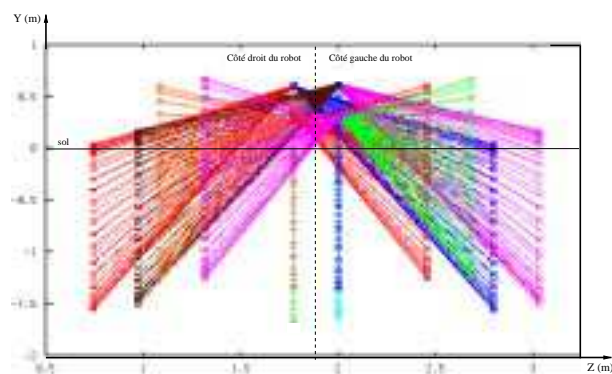


FIG. C.19: Carte de projection de la configuration de la Figure C.14, avec un angle d'ouverture de  $40^\circ$  entre les rayons extrêmes.

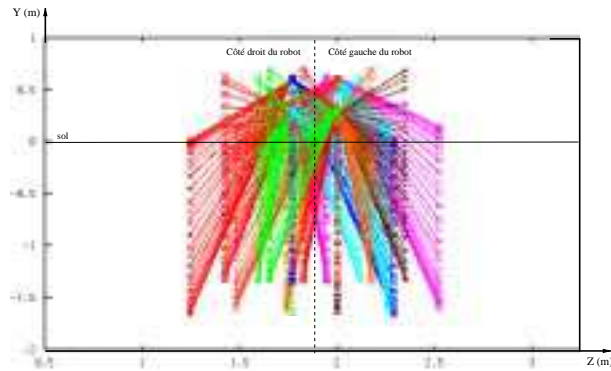


FIG. C.20: Carte de projection de la configuration de la Figure C.15, avec un angle d'ouverture de  $20^\circ$  entre les rayons extrêmes.

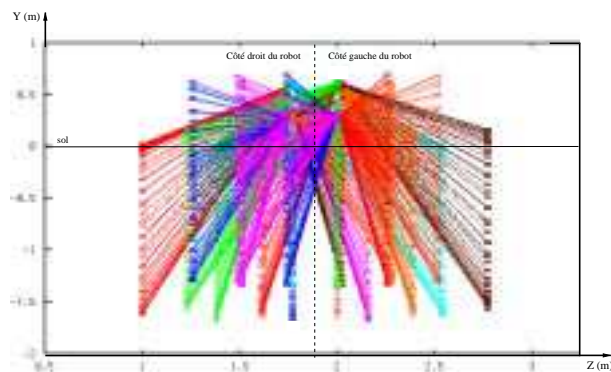


FIG. C.21: Carte de projection de la configuration de la Figure C.15, avec un angle d'ouverture de  $30^\circ$  entre les rayons extrêmes.

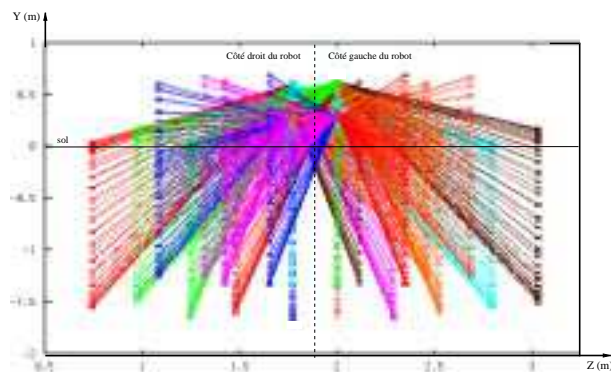


FIG. C.22: Carte de projection de la configuration de la Figure C.15, avec un angle d'ouverture de  $40^\circ$  entre les rayons extrêmes.

# Bibliographie

- [ABY86] P. Alevizos, J.D. Boissonnat and M. Yvinec. An optimal  $O(\log n)$  algorithm for contour reconstruction from rays. *3rd ACM Symposium on Computational Geometry*, pages 162–170, 1986.
- [AG85] W. Armstrong and M. Green. The dynamics of articulated rigid bodies for purposes of animation. In *Graphics Interface*, pages 407–415, May 1985.
- [Ali99] B. El Ali. *Contribution à la commande du centre de masse des robots marcheurs*. Thèse de doctorat, Université J. Fourier - Grenoble II, December 1999.
- [And83a] G. André. Capteurs d'environnement en robotique. *Techniques de l'Ingénieur, traité Mesures et Contrôle*, R8, Automatique et robotique :R7748 :1–12, 1983.
- [And83b] G. André. *Conception et modélisation de systèmes de perception proximétrique - Application à la commande en téléopération*. Thèse de doctorat, Université de Rennes II, October 1983.
- [AT78] S.G. Akl and G.T. Toussaint. An improved algorithm to check for polygon similarity. *Information Processing Letters*, 7(3) :127–128, 1978.
- [BB88] R. Barzel and A.H. Barr. A modeling system based on dynamics. In *Proceedings of ACM SIGGRAPH*, pages 179–188. Addison Wesley, July 1988.
- [BBG<sup>+</sup>94] N.I. Badler, R. Bindiganavale, J.P. Granieri, S. Wei and X. Zhao. Posture interpolation with collision avoidance. In *Computer Graphics*, pages 13–20, 1994.
- [BC89] A. Bruderlin and T. Calvert. Goal-directed dynamic animation of human walking. In *Computer Graphics*, pages 233–242, July 1989.
- [BC93] A. Bruderlin and T. Calvert. Interactive animation of personalized human locomotion. In *Graphics Interface*, pages 17–23, 1993.
- [BFM88] J.D. Boissonnat, B. Faverjon and J.P. Merlet. *Techniques de la robotique - Perception et planification - Tome 2*. Hermès, Paris, 1988.
- [BMT96] R. Boulic, R. Mas and D. Thalmann. A robust approach for the center of mass position control with inverse kinetics. *Journal of Computers and Graphics*, 20(5), 1996.
- [BMTT90] R. Boulic, N. Magnenat-Thalmann and D. Thalmann. A global human walking model with real-time kinematic personification. *Visual Computer*, 6(6) :344–358, December 1990.

- [Bou86] R. Boulic. *Conception assistée par ordinateur de boucles de commande avec capteurs en robotique et en téléopération*. Thèse de doctoratΓUniversité de Rennes IIΓNovember 1986.
- [BOW98] O. BruneauΓF. Ben OueddouΓand P-B. Wieber. Dynamic transition simulation of a walking anthropomorphic robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*Γpages 1392–1397ΓMay 1998.
- [BPZ96] M. BrudkaΓA. PacutΓand T. Zielinska. Processing of images created with the use of ultrasonic distance sensor. *5th National Conference on Robotics, Poland*Γ42 :316–23Γ1996.
- [BRRP97] B. BodenheimerΓC. RoseΓS. RosenthalΓand J. Pella. The process of motion capture : Dealing with the data. In *Eurographics Workshop on Computer Animation and Simulation*Γpages 3–18ΓSeptember 1997.
- [BT92] R. Boulic and D. Thalmann. Combined direct and inverse kinematic control for articulated figures motion editing. *Computer Graphics Forum*Γ11(4) :189–202Γ1992.
- [BW95] A. Bruderlin and L. Williams. Motion signal processing. In *Computer Graphics*Γpages 97–104ΓAugust 1995.
- [BW98] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Computer Graphics*Γpages 43–54Γ1998.
- [CHP89] J.E. ChadwickΓD.R. HaumannΓand R.E. Parent. Layered construction for deformable animated characters. *Computer Graphics*Γ23(3) :243–252ΓJuly 1989.
- [Coh92] M.F. Cohen. Interactive spacetime control for animation. In *Computer Graphics*Γpages 293–302ΓJuly 1992.
- [CY87] R. Cole and C. Yap. Shape from probing. *Journal of Algorithms*Γ8 :19–38Γ1987.
- [CYMTT92] M. CarignanΓY. YangΓN. Magnenat-ThalmannΓand D. Thalmann. Dressing animated synthetic actors with complex deformable clothes. *Computer Graphics*Γ26(2) :99–104ΓJuly 1992.
- [DH55] J. Denavit and R.S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Applied Mechanics*Γpages 215–221ΓJune 1955.
- [DLV96] L. DorstΓM. Van LambalgenΓand F. Voorbraak. *Reasoning with Uncertainty in Robotics*. Springer-Verlag; BerlinΓGermanyΓ1996.
- [Elf90] A. Elfes. Occupancy grid : A stochastic spatial representation for active robot perception. In *Proceeding of the Sixth Conference on Uncertainty in AI*Γ1990.
- [Fau96] F. Faure. An energy-based method for contact force computation. In *Proceedings of Eurographics'96*Γpages 357–366ΓAugust 1996. Computer Graphics ForumΓVolume 15ΓNumber 3.
- [Fau98] F. Faure. Interactive solid animation using linearized displacement constraints. In *Computer Animation and Simulation*Γpages 61–72Γ1998.

- [FDCGM97] F. Faure, G. Debunne, M.P. Cani-Gascuel and F. Multon. Dynamic analysis of human walking. In *Eurographics Workshop on Computer Animation and Simulation* pages 53–65, September 1997.
- [FKU77] H. Fuchs, Z. Kedem and S.P. Uselton. Optimal surface reconstruction from planar contours. *Communications of the ACM* 20(10) :693–702, 1977.
- [Gén98] F. Génot. *Contributions à la modélisation et à la commande des systèmes mécaniques de corps rigides avec contraintes unilatérales*. Thèse de doctorat INPG - Grenoble II, January 1998.
- [GG94] J.D. Gascuel and M.P. Gascuel. Displacement constraints for interactive modeling and animation of articulated structures. In *The Visual Computer* pages 191–204, March 1994.
- [Gle98] M. Gleicher. Retargetting motion to new characters. In *Computer Graphics* pages 33–42, 1998.
- [GM85] M. Girard and A.A. Maciejewski. Computational modelling for the computer animation of legged figures. *Computer Graphics* 19(3) :263–270, 85.
- [GR96] S. Guo and J. Roberge. A high-level control mechanism for human locomotion based on parametric frame space interpolation. In *Eurographics Workshop on Computer Animation and Simulation* pages 95–107, Poitiers, France, September 1996. Springer Verlag.
- [Gre91] M. Green. Using dynamics in computer animation : Control and solution issues. In *Making Them Move* pages 281–314. Morgan Kaufmann Publishers, inc., 1991.
- [GVP91] M.P. Gascuel, A. Verroust and C. Puech. A modeling system for complex deformable bodies suited to animation and collision processing. *The Journal of Visualization and Computer Animation* 2(3), August 1991.
- [HBG92] D. House, D. Breen and P. Getto. On the dynamic simulation of physically-based particle-system models. In *Third Eurographics Workshop on Animation and Simulation*, Cambridge, England, September 1992.
- [Hod96] J. Hodgins. Three-dimensional human running. In *Proceedings of the IEEE Conference on Robotics and Automation*, April 1996. Minneapolis, Minnesota.
- [HP97] J.K. Hodgins and N.S. Pollard. Adapting simulated behaviors for new characters. *Computer Graphics* 15(3) :153–162, August 1997.
- [HWBO95] J.K. Hodgins, W.L. Wooten, D.C. Brogan and J.F. O'Brien. Animating human athletics. In *Computer Graphics* pages 71–78, August 1995.
- [IC87] P.M. Isaacs and M.F. Cohen. Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics. In *Computer Graphics* pages 215–224, July 1987.
- [IC88] P.M. Isaacs and M.F. Cohen. Mixed method for complex kinematic constraints in dynamic figure animation. In *The Visual Computer* pages 296–305, December 1988.

- [IRT81] V. T. Inman, H. Ralston and F. Todd. *Human Walking*. Baltimore ; London : Williams and Wilkins, 1981.
- [KB93] H. Ko and N.I. Badler. Straight line walking animation based on kinematic generalization that preserves the original characteristics. In *Graphics Interface*, pages 9–16, Toronto, Ontario, Canada, May 1993.
- [KB96] H. Ko and N.I. Badler. Animating human locomotion with inverse dynamics. In *IEEE Computer Graphics and Applications*, pages 50–59, March 1996.
- [KC96] H. Ko and J. Cremer. VRLOCO : real-time human locomotion from positional input streams. *Presence* 5(4) :367–380, 1996.
- [Ko94] H. Ko. *Kinematic and Dynamic techniques for Analyzing, Predicting, and Animating Human Locomotion*. PhD thesis, University of Pennsylvania, 1994.
- [KS97] T. Komura and Y. Shinagawa. A muscle-based feed-forward controller of the human body. In *Proceedings of Eurographics'97*, pages 166–176, 1997. Computer Graphics Forum, Volume 16, Number 3.
- [Las87] J. Lasseter. Principles of traditional animation applied to 3D computer animation. *Computer Graphics* 21(4) :35–44, July 1987.
- [Lau83] C. Laureau. Perception géométrique tridimensionnelle en robotique. *Techniques de l'Ingénieur, traité Mesures et Contrôle* R8, Automatique et robotique :R7750 :1–10, 1983.
- [LGC94] Z. Liu, S.J. Gortler and M.F. Cohen. Hierarchical spacetime control. In *Proceedings of ACM SIGGRAPH*, pages 35–42, Orlando, Florida, July 1994. Addison Wesley.
- [LH91] D. Langer and M. Hebert. Building qualitative elevation maps from side scan sonar data for autonomous underwater navigation. *IEEE International Conference on Robotics and Automation* 3 :2478–83, 1991.
- [LvdPF96] J. Laszlo, M. van de Panne and E. Fiume. Limit cycle control and its application to the animation of balancing and walking. In *Computer Graphics*, pages 155–162, August 1996.
- [MA96] F. Multon and B. Arnaldi. Gait customization using wavelets. In *Likely Characters '96*. Snowbird, USA, October 1996.
- [Mai95] R. Maiocchi. 3D character animation using motion capture. In *Interactive Computer Animation*, Prentice Hall, 1995. N. Magnenat-Thalmann and D. Thalmann (eds.).
- [Man76] G. Manacher. An application of pattern matching to a problem in geometrical complexity. *Information Processing Letters* 2(1) :6–7, 1976.
- [MBT96] T. Molet, R. Boulic and D. Thalmann. A real time anatomical converter for human motion capture. In *Eurographics Workshop on Computer Animation and Simulation*, pages 79–94, September 1996.
- [Meh99] D. Meharzi. *Caractérisation en vue de l'implémentation de capteurs sur un robot*. Rapport de stage, 1999.

- [MHBT97] T. Molet, Z. Huang, R. Boulic and D. Thalmann. An animation interface designed for motion capture. In *Proceedings of Computer Animation*, pages 77–85, May 1997.
- [Mul98] F. Multon. *Contrôle du mouvement des humanoïdes de synthèse*. Thèse de doctorat, Université de Rennes II, October 1998.
- [NM93] J.T. Ngo and J. Marks. Spacetime constraints revisited. In *Computer Graphics*, pages 343–350, August 1993.
- [NRTMT95] H. Noser, O. Renault, D. Thalmann and N. Magnenat-Thalmann. Navigation for digital actors based on synthetic vision, memory and learning. *Computer & Graphics*, 19(1) :7–19, 1995.
- [Opa96] A. Opalach. *Implicit Surfaces for Modelling and Character Animation*. PhD Thesis, University of Sheffield, Department of Computer Science, 1996.
- [RGBC96] C. Rose, B. Guenter, B. Bodenheimer and M.F. Cohen. Efficient generation of motion transitions using spacetime constraints. In *Computer Graphics*, pages 147–154, August 1996.
- [RH91] M.H. Raibert and J.K. Hodgins. Animation of dynamic legged locomotion. In *Computer Graphics*, pages 349–358, July 1991.
- [RTJ89] Y. Roth-Tabak and R. Jain. Building an environment model using depth information. *Computer*, 22(6) :85–90, June 1989.
- [SBE91] C. Samson, M. Le Borgne and B. Espiau. *Robot Control : The Task-Function Approach*. Clarendon Press, Oxford Science Publications, U.K., 1991.
- [SH81] R. Siegel and J.R. Howell. *Thermal Radiation Heat Transfer*. Second Edition, McGraw-Hill, 1981.
- [SIE53] J.B. Saunders, V.T. Inman and H.D. Eberhart. The major determinants in normal and pathological gaits. *J. Bone Joint Surgery*, 35A :543–558, 1953.
- [Sim94] K. Sims. Evolving virtual creatures. In *Computer Graphics*, pages 15–22, July 1994.
- [SPCM97] F. Scheepers, R. E. Parent, W. E. Carlson and S. F. May. Anatomy-based modeling of the human musculature. In *Computer Graphics*, pages 163–172, August 1997.
- [ST95] J. Shen and D. Thalmann. Interactive shape design using metaballs and splines. In *Implicit Surfaces'95—the First Eurographics Workshop on Implicit Surfaces*, pages 187–195, Grenoble, France, April 1995.
- [Sto99] D. Stora. *Animation de formes organiques*. Rapport de stage de DEA, 1999.
- [Sug84] K. Sugihara. An  $n \log n$  algorithm for determining the congruity of polyhedra. *Journal of Computer and Systems Sciences*, 29 :36–47, 1984.
- [Tou80] G.T. Toussaint. Pattern recognition and geometrical complexity. *IEEE on Pattern Recognition, Miami*, pages 1324–1347, 1980.
- [TSC96] D. Thalmann, J. Shen and E. Chauvineau. Fast human body deformations for animation and vr applications. In *Proceedings of Computer Graphics International '96*, pages 166–174. IEEE Computer Society Press, June 1996.



- [Tur95] R. Turner. Leman : A system for construsting and animating layered elastic characters. In *Computer Graphics- Developments in Virtual Environments* pages 185–203. Academic Press. San Diego. CA. June 1995.
- [UAT95] M. Unuma. K. Anjyo. and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *Computer Graphics* pages 91–96. August 1995.
- [VBSD90] M. Vukobratovic. B. Borovac. D. Surla. and D. Stokic. *Scientific Fundamentals of Robotics 7. Biped Locomotion : Dynamics Stability, Control and Application*. Springer-Verlag. New York. 1990.
- [VCMT95] P. Volino. M. Courchesne. and N. Magnenat-Thalmann. Versatile and efficient techniques for simulating cloth and other deformable objects. *Computer Graphics* pages 137–144. August 1995.
- [VDO92] C. L. Vaughan. B. L. Davis. and J. C. O'Connor. *Dynamics of Human Gait*. Human Kinetics Publishers. Champaign. Illinois. 1992.
- [vdP97] M. van de Panne. From footprints to animation. *Computer Graphics Forum* 16(4) :211–223. October 1997.
- [vdPF93] M. van de Panne and E. Fiume. Sensor-actuator networks. In *Computer Graphics* pages 335–342. August 1993.
- [vdPKF94] M. van de Panne. R. Kim. and E. Fiume. Virtual wind-up toys for animation. In *Graphics Interface* pages 208–215. May 1994.
- [VMTJT96] P. Volino. N. Magnenat-Thalmann. S. Jianhua. and D. Thalmann. An evolving system for simulating clothes on virtual actors. *IEEE Computer Graphics and Applications* 16(5) :42–51. 1996.
- [WF97] M. Walter and A. Fournier. Growing and animating polygonal models of animals. In *Proceedings of Eurographics'97* pages 151–158. 1997. Computer Graphics Forum. Volume 16. Number 3.
- [Wil91] J. Wilhelms. Dynamic experiences. In *Making Them Move* pages 265–279. Morgan Kaufmann Publishers. inc. 1991.
- [Win90] D.A. Winter. *Biomechanics and Motor Control of Human Movement*. John Wiley & Sons. 1990.
- [WK88] A. Witkin and M. Kass. Spacetime constraints. In *Computer Graphics* pages 159–168. August 1988.
- [WP95] A. Witkin and Z. Popovic. Motion warping. In *Computer Graphics* pages 105–108. August 1995.
- [WW92] A. Watt and M. Watt. *Advanced Animation and Rendering Techniques - Theory and Practice*. ACM Press. Addison-Wesley. 1992.
- [ZCK97] Q-H. Zhu. Y. Chen. and A. Kaufman. Real-time biomechanically-based muscle volume deformation using fem. In *Proceedings of Eurographics'98* pages 276–284. 1997. Computer Graphics Forum. Volume 17. Number 3.
- [Zel82] D. Zeltzer. Motor control techniques for figure animation. *IEEE Computer Graphics and Applications* 2(9) :53–59. 82.
- [Zel83] D. Zeltzer. Knowledge-based animation. *ACM SIGGRAPH/SIGART, Workshop on Motion* pages 187–192. April 83.