



**HAL**  
open science

# Programmation bayésienne des bras manipulateurs

Ruben Garcia Ramirez

► **To cite this version:**

Ruben Garcia Ramirez. Programmation bayésienne des bras manipulateurs. Autre [cs.OH]. Institut National Polytechnique de Grenoble - INPG, 2003. Français. NNT: . tel-00147378

**HAL Id: tel-00147378**

**<https://theses.hal.science/tel-00147378>**

Submitted on 16 May 2007

**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.

**INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE**

N° attribué par la bibliothèque

| / / / / / / / / / / |

**THÈSE**

pour obtenir le grade de

**DOCTEUR DE L'INPG**

***Spécialité : « Imagerie, Vision, Robotique »***

préparée au laboratoire GRAVIR, dans le cadre de l'École Doctorale « **Mathématique, Sciences et Technologies de l'Information, Informatique** »

présentée et soutenue publiquement par

**Rubén Senen GARCÍA RAMÍREZ**

le 27/05/2003

**Titre :**

**Programmation bayésienne  
des bras manipulateurs**

**Directeur de Thèse :**

Emmanuel MAZER

**Composition du jury :**

M.	Roger MOHR	Président
M.	François CHAUMETTE	Rapporteur
M.	Patrick RIVES	Rapporteur
M.	Emmanuel MAZER	Directeur de thèse
M.	Jean Michel BONNET DES TUVES	Examineur
M.	Jacques GUILLET	Examineur



# Remerciements

Je tiens à remercier Emmanuel Mazer pour avoir dirigé ce travail, pour ces multiples critiques et réflexions. J'ai beaucoup apprécié ses connaissances, sa patience et ses qualités scientifiques et humaines.

Je tiens à remercier Olivier Aycard pour sa participation à la rédaction de ce document. J'apprécie beaucoup ses connaissances, sa capacité de rédaction et de vision globale, ses points de vue et surtout son amitié.

Je remercie Pierre Bessière pour ses conseils.

Je tiens à remercier les membres de mon jury. En particulier les rapporteurs ; Monsieur François Chaumette et Monsieur Patrick Rives. Je tiens à remercier aussi Monsieur Roger Mohr, Monsieur Jean Michel Bonet des Tuves et Monsieur Jacques Guillet.

Je remercie Juan Manuel Ahuactzin Larios pour sa confiance et son aide qui m'ont permis d'arriver dans cette équipe.

J'adresse mes remerciements aux membres de l'équipe CYBERMOVE, en particulier Kamel Mekhnacha, Christophe Coué et Olivier Malrait pour leur soutien et leur aide.

Je tiens à remercier le gouvernement du Mexique pour avoir financé cette thèse, gérée par le CONACYT. Je remercie aussi l'Instituto Tecnológico de Puebla pour avoir participé à ce financement.

Je tiens à remercier mes parents, mes frères et soeurs pour leur aide et leurs encouragements constants.

Les derniers mots sont pour remercier ma famille, d'abord mes enfants : Mayela, Alejandra et Rubén pour leurs encouragements constants, leur adaptation à la société française et leur capacité d'apprentissage. Enfin, je tiens à te remercier Manuela pour ta présence, ton aide inestimable, ta patience, ta confiance et ton soutien permanent.



# Résumé

Cette thèse porte sur la programmation bayésienne de bras manipulateurs équipés d'un système de vision stéréoscopique. L'implantation d'une tâche de prise et de dépose d'un objet est donnée en exemple pour évaluer l'approche. Cette tâche met en jeu des modèles géométriques du bras, du système de vision stéréoscopique et des objets manipulés. L'incertitude associée aux modèles géométriques, aux capteurs des robots et au système de vision est prise en compte.

La méthode de programmation est formelle et systématique. Elle consiste en trois parties : La description où sont définies les variables pertinentes et la distribution de probabilité conjointe. L'inférence qui vise à construire une distribution de probabilité sur les commandes motrices sachant les informations sensorielles, et enfin le choix d'une commande à partir de la distribution précédente.

Une nouvelle méthode de programmation de robot dite «inverse» est introduite. Des méthodes de résolutions approchées sont présentées visant à combattre la complexité du calcul liée à cette approche. Une expérimentation qualitative du système de programmation est présentée.

## Mots-clefs

robotique, incertitude géométrique, programmation bayésienne, calibration, intégration numérique, algorithmes génétiques, programmation inverse, programmation probabiliste, inférence, apprentissage, mixture de gaussienne, asservissement visuel.



# Table des matières

<b>Liste des Tableaux</b>	<b>vii</b>
<b>Liste des Figures</b>	<b>ix</b>
<b>Introduction</b>	<b>1</b>
Contribution et Plan de lecture . . . . .	2
<b>1 Éléments Mathématiques</b>	<b>7</b>
1.1 Proposition logique . . . . .	7
1.1.1 Opérations de base de propositions logiques . . . . .	7
1.1.2 Probabilité de propositions logiques . . . . .	8
1.1.3 Règles de calcul sur les propositions logiques . . . . .	9
1.2 Variables . . . . .	10
1.2.1 Variable aléatoire discrète . . . . .	10
1.3 Théorème de Bayes . . . . .	14
1.4 Programme probabiliste . . . . .	15
1.4.1 Description . . . . .	15
1.4.2 Question . . . . .	15
1.5 Suivi de la couleur . . . . .	16
1.5.1 Représentation de la couleur . . . . .	18
1.5.2 Détection de la position d'un objet dans l'image . . . . .	18
1.5.3 Poursuite visuelle . . . . .	21
1.6 Conclusion . . . . .	24
<b>2 Contexte</b>	<b>25</b>
2.1 Travaux de Thrun . . . . .	25
2.1.1 Localisation . . . . .	26
2.1.2 Contrôle . . . . .	28
2.1.3 Programmation et apprentissage . . . . .	29
2.1.4 Discussion . . . . .	30
2.2 Travaux de Lebeltel . . . . .	31
2.2.1 Discussion . . . . .	34
2.3 Travaux de Mekhnacha . . . . .	35



2.3.1	Problème géométrique probabiliste et graphe cinématique . . . . .	36
2.3.2	Construction d'un état de connaissance . . . . .	37
2.3.3	Résolution du problème . . . . .	39
2.3.4	Méthodes de résolution . . . . .	39
2.3.5	Discussion . . . . .	40
2.4	Conclusion . . . . .	40
<b>3</b>	<b>Description des robots</b>	<b>43</b>
3.1	Robot cartésien . . . . .	43
3.1.1	Architecture du robot cartésien . . . . .	45
3.1.2	Système de vision stéréoscopique . . . . .	46
3.2	Bras robot . . . . .	46
3.2.1	Architecture du bras manipulateur . . . . .	49
3.2.2	Pince du bras robot . . . . .	50
3.3	Description de la tâche de "pick and place" . . . . .	51
3.4	Conclusion . . . . .	52
<b>4</b>	<b>Sous-programmes probabilistes</b>	<b>53</b>
4.1	Modèle de mouvement . . . . .	53
4.1.1	Description du problème . . . . .	53
4.1.2	Trajectoire idéale . . . . .	54
4.1.3	Temps de parcours . . . . .	56
4.1.4	Vitesse idéale de parcours . . . . .	58
4.1.5	Variables . . . . .	58
4.1.6	Décomposition pour le mouvement . . . . .	59
4.1.7	Formes paramétriques pour le mouvement . . . . .	59
4.1.8	Inférence probabiliste . . . . .	60
4.1.9	Solution pour le mouvement du bras . . . . .	61
4.1.10	Conclusion . . . . .	63
4.2	Localisation d'objets avec la vision stéréoscopique . . . . .	63
4.2.1	Description du problème . . . . .	64
4.2.2	Graphe cinématique du problème . . . . .	65
4.2.3	Variables pour la localisation stéréoscopique . . . . .	67
4.2.4	Décomposition pour la localisation . . . . .	69
4.2.5	Formes paramétriques . . . . .	69
4.2.6	Inférence probabiliste . . . . .	70
4.2.7	Solution pour la localisation . . . . .	71
4.2.8	Conclusion . . . . .	75
4.3	Modèle de saisie . . . . .	75
4.3.1	Description de la prise . . . . .	76
4.3.2	Description de l'état de connaissance . . . . .	76
4.3.3	Inférence d'une prise . . . . .	79
4.3.4	Conclusion . . . . .	80

4.4	Conclusion	80
<b>5</b>	<b>Programmation bayésienne de robots</b>	<b>83</b>
5.1	Gestion d'activités	83
5.1.1	Description du problème	84
5.1.2	Variables	87
5.1.3	Décomposition	88
5.1.4	Formes paramétriques	89
5.1.5	Inférence	96
5.1.6	Solution	97
5.1.7	Résultat	97
5.1.8	Conclusion	97
5.2	Distribution de probabilités entre positions et actions	98
5.2.1	Calcul des positions de référence	98
5.2.2	Calcul des capteurs d'état	100
5.2.3	Calcul de la position but	101
5.2.4	Calcul de la commande de la pince	102
5.2.5	Conclusion	102
5.3	Intégration pour programmation du "Pick and Place"	102
5.3.1	"Pick and Place"	103
5.3.2	Variables	104
5.3.3	Décomposition	106
5.3.4	Formes paramétriques	108
5.3.5	Commande par inférence probabiliste	111
5.3.6	Commande du système	112
5.3.7	Conclusion	125
5.4	Expérimentation	125
5.5	Conclusion	128
<b>6</b>	<b>Conclusion</b>	<b>131</b>
<b>A</b>	<b>Bibliographie</b>	<b>135</b>
A.1	Travaux de Puget	135
A.1.1	Graphe d'état	135
A.1.2	Incertitude de position	135
A.1.3	Problèmes causés par l'existence de cycles	136
A.1.4	Discussion	137
A.2	Travaux de De la Rosa	137
A.3	Travaux de Ruiz	138
A.3.1	Fondements	139
A.3.2	Famille mathématique	140
A.3.3	Approximation par mixture de distributions normales univariées	140
A.3.4	Apprentissage de la fonction de densité de probabilité conjointe	147

A.3.5	Discussion	148
<b>B</b>	<b>Calibration et problèmes géométriques</b>	<b>149</b>
B.1	Autocalibration d'une caméra	150
B.1.1	Définition	150
B.1.2	Transformation Espace - Caméra	151
B.1.3	Transformation Caméra - Rétine	152
B.1.4	Transformation de la projection sur la rétine à l'image	153
B.1.5	Matrice de projection	155
B.1.6	Calibration	157
B.1.7	Variables	157
B.1.8	Décomposition	159
B.1.9	Formes paramétriques	159
B.1.10	Inférence	159
B.1.11	Solution	160
B.1.12	Conclusion	162
B.2	Vision stéréoscopique	163
B.2.1	Calibration de la paire stéréoscopique	163
B.2.2	Localisation d'un point avec la paire stéréoscopique	168
B.2.3	Conclusion	174
B.3	Calibration du système multirobots	175
B.3.1	Description du problème de la calibration multirobots	175
B.3.2	Graphe cinématique pour la calibration multirobots	176
B.3.3	Variables pour la calibration multirobots	179
B.3.4	Décomposition pour le système multirobots	179
B.3.5	Formes paramétriques pour la calibration multirobots	180
B.3.6	Inférence probabiliste pour la calibration multirobots	181
B.3.7	Solution pour la calibration multirobots	183
B.3.8	Conclusion	184
B.4	Expérimentation	185
B.4.1	Conclusion	186
B.5	Conclusion	187
<b>C</b>	<b>Traitement de la couleur</b>	<b>189</b>
C.1	Algorithme de travis pour aller de RGB vers HSV	189
<b>D</b>	<b>Méthodes de tirage</b>	<b>191</b>
D.1	Tirage direct à partir d'une distribution normale univariée	191
<b>E</b>	<b>Méthode d'intégration de Monte Carlo</b>	<b>193</b>
<b>F</b>	<b>Algorithme génétique continu</b>	<b>197</b>

<b>G Éléments de Géométrie</b>	<b>199</b>
G.1 Transformations géométriques . . . . .	199
G.1.1 Matrice homogène de transformation . . . . .	199
G.1.2 Vecteur de pose . . . . .	200
<b>Bibliographie</b>	<b>203</b>



# Liste des Tableaux

3.1	Spécifications du robot cartésien. . . . .	45
3.2	Spécifications du robot Stäubli Rx90. . . . .	47
3.3	Description de la pince. . . . .	50
4.1	Ensemble d'arcs du graphe pour la localisation. . . . .	68
4.2	Pince vue par la vision stéréoscopique. . . . .	73
4.3	Coordonnées des points sur la pince fournies par la vision stéréoscopique. . . . .	74
4.4	Ensemble d'arcs du graphe probabiliste pour la localisation. . . . .	74
4.5	Position de la pince trouvée par la localisation stéréoscopique. . . . .	75
5.1	Distribution pour l'activité dans le temps $-1$ en sachant l'activité actuelle : $p(A_{-1} A)$ . . . . .	90
5.2	Formes paramétriques pour les capteurs de : position de base, position d'ap- proche et position de prise . . . . .	92
5.3	Formes paramétriques pour les capteurs de : positions de distance avec l'objet, position d'approche avec l'objet et position de pose de l'objet . . . . .	93
5.4	Distributions pour : le capteur de la position de distance, le capteur d'information sur la connaissance de la position de l'objet et le capteur de la position pour déposer l'objet . . . . .	94
5.5	Distributions pour : le capteur indiquant si le bras porte l'objet et le capteur de l'état de la pince . . . . .	95
A.1	Mixture de normales généralisées avec 2 attributs et 5 composantes. . . . .	144
B.1	Les arcs du graphe pour la calibration du système multirobots. . . . .	178
B.2	Description de l'objet. . . . .	185
B.3	Position de l'objet fournie par la V.S. . . . .	186
B.4	Position de l'objet sur le repère du bras. . . . .	186



# Liste des Figures

1.1	Programme probabiliste . . . . .	17
1.2	Robot cartésien . . . . .	17
2.1	Pré-traitement des valeurs de capteurs, ici $Dir = -4$ et $Prox = 3$ (tiré de [Leb99])	32
2.2	Robot avec un degré de liberté (tiré de [Mek99]) . . . . .	35
2.3	Graphe cinématique pour le problème d'inversion géométrique (tiré de [Mek99])	36
3.1	Robot cartésien . . . . .	44
3.2	Modèle physique du robot cartésien . . . . .	45
3.3	Chaîne cinématique du robot cartésien . . . . .	46
3.4	Caméras pour la capture d'images. . . . .	47
3.5	Bras robot stäubli RX90 . . . . .	48
3.6	Pince du bras manipulateur stäubli RX90 . . . . .	48
3.7	Modèle physique du bras manipulateur . . . . .	49
3.8	Chaîne cinématique du bras Stäubli . . . . .	50
3.9	Modèle de la pince. . . . .	51
4.1	Transformation de la position de la pince . . . . .	54
4.2	Trajectoire idéale . . . . .	55
4.3	Points de référence de la pince vus par la paire stéréoscopique. . . . .	64
4.4	Graphe pour la localisation d'un objet. . . . .	66
4.5	Distribution normale avec température. . . . .	72
4.6	Objet avec plan de prise et prises. . . . .	77
4.7	Plan de prise avec les paramètres de la pose de prise . . . . .	78
5.1	Comportement du bras . . . . .	86
5.2	Position d'approche. . . . .	100
5.3	Bras en position d'approche . . . . .	126
5.4	Bras en position de prise de l'objet . . . . .	127
5.5	Bras en position de distance avec l'objet. . . . .	127
5.6	Bras vers et en position d'approche avec l'objet. . . . .	128
5.7	Place de l'objet. . . . .	128
5.8	Bras en position de distance. . . . .	129
5.9	Simulation de mouvement du bras. . . . .	130



A.1	L'incertitude de contrôle et son effet sur l'incertitude de position (tiré de [DLRR96])	138
A.2	La fonction de densité de probabilité d'un phénomène aléatoire approchée par une mixture de distributions normales univariées.	141
A.3	Phénomène aléatoire avec deux attributs et mixture de distributions normales univariées pour approcher la distribution de densité de probabilités du phénomène.	143
B.1	Phénomène physique dans une caméra	150
B.2	Modèle géométrique d'une caméra	151
B.3	Triangles semblables	152
B.4	Transformation de la rétine à l'image	154
B.5	Transformations géométriques et projection dans une caméra	156
B.6	Grille de calibration	158
B.7	Calibration probabiliste	158
B.8	triangulation stéréoscopique.	163
B.9	Grille de calibration vue par la paire stéréoscopique	164
B.10	Transformations dans la paire stéréoscopique	165
B.11	Modèle de la paire stéréoscopique.	165
B.12	Les matrices de projection juste après de la calibration.	166
B.13	Repère stéréoscopique sur la caméra gauche	167
B.14	Graphe pour la calibration du système multirobots	177
B.15	Objet vu par la paire stéréoscopique	186
E.1	Intégration par Monte Carlo.	193
G.1	Position relative entre repères	199

# Introduction

L'utilisation des probabilités bayésiennes en robotique autonome est une démarche classique de cette discipline. Les premières études sur le sujet remontent aux années 1976 avec les travaux de Moravec [MG76]. L'augmentation de la puissance de calcul des ordinateurs rend ces techniques de plus en plus attrayantes pour les roboticiens. Elles ont d'ailleurs été remises au goût du jour par plusieurs groupes de recherche aux États Unis et en Europe. Nos travaux s'inscrivent dans le cadre général de ce "renouveau bayésien" mais aussi dans un projet visant à mieux comprendre l'interaction des systèmes sensori-moteur avec leur environnement. Ce projet trouve son prolongement dans les sciences cognitives où les systèmes sensori-moteur sont issus du vivant. Dans ce prolongement, la programmation bayésienne<sup>1</sup> devient l'outil d'analyse et de synthèse. Il permet à la fois de construire des modèles de fonctionnement pour les systèmes vivants et de réaliser des systèmes artificiels les imitant.

Notre travail s'insère dans ce projet ambitieux en s'intéressant à la programmation probabiliste de robots. Il vise à généraliser les résultats obtenus par notre équipe en la matière. De quels résultats s'agit-il ? Pierre Bessière propose un formalisme rigoureux pour l'inférence probabiliste, Olivier Lebeltel l'applique à la programmation d'un robot mobile et Kamel Mekhnacha à la réalisation d'un système de CAO probabiliste. De là, trois questions principales : le formalisme utilisé est-il applicable partout ? Peut-on étendre le concept de programmation probabiliste à un système complexe ? La modélisation probabiliste peut-elle être utilisée pour commander des bras manipulateurs ?

Sans pouvoir répondre de façon définitive à ces questions nous avançons que le raisonnement probabiliste peut être utilisé comme cadre général de programmation pour des bras manipulateurs équipés de capteurs de vision.

En effet, nous montrerons que le formalisme utilisé est général : il peut s'étendre pour piloter des bras manipulateurs, il accommode aussi bien des capteurs de proximité que des systèmes de vision stéréoscopique, il permet de choisir entre des modèles empiriques et des modèles formels, il autorise plusieurs types de programmation : réactive, planifiée et inverse. Cette gamme de possibilités font de la programmation probabiliste un cadre potentiellement intéressant pour le développement de systèmes autonomes : c'est la thèse que nous voudrions défendre dans ce

---

<sup>1</sup>Dans la suite du document, nous emploierons indifféremment les termes programmation bayésienne et programmation probabiliste

document.

## Contribution et plan de lecture

Très schématiquement, notre contribution tient dans les deux équations explicitées au chapitre 5 : les équations (5.6) et (5.11). Nous les reproduisons ici :

$$\begin{aligned}
 & p \left( \begin{array}{c} Z_1 \dots Z_N \\ B_{g_1} \dots B_{g_N} B_{d_1} \dots B_{d_N} \\ P_g P_d \\ M_1 \dots M_N \\ K_1 \dots K_N \\ PosObj PosTable \\ PosPrise PosApp \\ PosElop PosDep \\ PosAppp PosElo \\ \Theta X \\ S_{base} \\ S_{app} S_{prise} S_{elop} \\ S_{appp} S_{dep} S_{elo} \\ S_{pince} S_{objprise} \\ A_{-1} A_{-2} A_{-3} A \\ MPince \\ But E \dot{\Theta}_{-1} \dot{\Theta} \end{array} \right) \\
 & = \left[ \begin{array}{l} \prod_{i=1}^N p(Z_i) \\ \prod_{i=1}^N p(B_{g_i}) \\ \prod_{i=1}^N p(B_{d_i}) \\ p \left( P_g \mid \begin{array}{c} Z_1 \dots Z_L \\ B_{g_1} \dots B_{g_L} \end{array} c_{cam\_g} \right) \\ p \left( P_d \mid \begin{array}{c} Z_1 \dots Z_L \\ B_{d_1} \dots B_{d_L} \end{array} c_{cam\_d} \right) \\ \prod_{i=1}^N p(M_i \mid P_g P_d B_{g_i} B_{d_i} c_{VS}) \\ \prod_{i=1}^N p(K_i) \\ p \left( PosObj \mid \begin{array}{c} K_1 \dots K_L M_1 \dots M_L \\ Z_1 \dots Z_L c_{loc\_Obj} \end{array} \right) \\ p \left( PosTable \mid \begin{array}{c} K_{L+1} \dots K_N M_{L+1} \dots M_N \\ Z_{L+1} \dots Z_N c_{loc\_Table} \end{array} \right) \\ p(PosPrise \mid PosObj c_{prise}) \\ p(PosDep \mid PosTable PosPrise c_{placer\_objet}) \\ p(PosApp \mid PosPrise) \\ p(PosElop \mid PosPrise) \\ p(PosAppp \mid PosDep) \\ p(PosElo \mid PosDep) \\ p(\Theta) p(X \mid \Theta) \\ p(S_{base} \mid X) \\ p(S_{app} \mid PosApp X) \\ p(S_{prise} \mid PosPrise X) \\ p(S_{elop} \mid PosElop X) \\ p(S_{appp} \mid PosAppp X) \\ p(S_{dep} \mid PosDep X) \\ p(S_{elo} \mid PosElo X) \\ p(S_{pince}) \\ p(S_{objprise}) \\ p(A_{-1}) p(A_{-2}) p(A_{-3}) \\ p(A \mid A_{-1} A_{-2} A_{-3} S_{base} \dots S_{objprise} c_{activité}) \\ p(MPince \mid A) \\ p(But \mid A) \\ p(\dot{\Theta}_{-1}) \\ p(E) \\ p(\dot{\Theta} \mid \Theta \dot{\Theta}_{-1} But E c_{mov}) \end{array} \right) \quad (1)
 \end{aligned}$$

$$p \left( \begin{array}{c|c} \dot{\Theta} \text{ MPince} & \begin{array}{l} Z_1 \dots Z_N \ B_{g_1} \dots B_{g_N} \ B_{d_1} \dots B_{d_N} \\ K_1 = \text{vrai} \dots K_N = \text{vrai} \\ S_{\text{pince}} \ S_{\text{objprise}} \\ A_{-1} \ A_{-2} \ A_{-3} \\ E = 0 \ \Theta \ \dot{\Theta}_{-1} \end{array} \\ \hline & c_{\text{pick\_and\_place}} \end{array} \right) \quad (2)$$

La première équation correspond au modèle de notre système : un robot portique à cinq degrés de liberté, un bras robot à six degrés de liberté et un système de vision stéréoscopique. Ce modèle lie dans une seule distribution de probabilités les 207 variables nécessaires à la description du système et de la tâche à accomplir : ici une opération de “pick and place” . Cette distribution est construite à partir de distributions élémentaires (en partie droite de l’égalité). Chacune de ces distributions élémentaires correspond à la description probabiliste d’une sous-partie du système et peut être définie par une équation similaire.

La deuxième équation correspond à un problème d’inférence probabiliste utilisant la distribution que nous venons de présenter. Une fois résolu, ce problème permet d’obtenir une distribution de probabilité conjointe sur les valeurs des vitesses pour les articulations du bras robot ( $\dot{\Theta}$ ) et sur la variable commandant l’ouverture de la pince (*MPince*). En résolvant ce problème vingt fois par seconde et en tirant les valeurs de ces variables pour les envoyer au contrôleur du robot, on obtient un comportement cohérent du système : le système localise un objet, le saisit et le dépose à un endroit spécifié.

On voit ici plusieurs changements radicaux du point de vue de la programmation des robots :

1. l’incertitude est explicitement prise en compte.
2. le système est considéré globalement : il n’existe pas (au moins dans le principe) de cloisonnement entre variables. Chaque variable influe sur toutes les autres. Par exemple, la précision de la caméra influe sur la position de prise de l’objet.
3. les résultats sont donnés sous forme probabiliste.
4. tout en ressemblant à un système purement réactif (on lit les valeurs des capteurs, on fait l’inférence, puis on affiche les valeurs des actionneurs), la méthode de programmation fait appel à des modèles mathématiques comme les modèles géométriques des robots ou des caméras.
5. enfin, une nouvelle notion de programmation dite “inverse” est introduite. On ne dit plus quoi faire pour une lecture de capteur donnée mais quoi percevoir pour une action donnée.

Ce rapport se compose de six chapitres et sept annexes.

**Le chapitre 1** établit les bases mathématiques utilisées dans notre approche et la méthode de programmation proposée ; “le programme probabiliste” avec un exemple de poursuite visuelle.

**Le chapitre 2** présente les travaux de Thrun, Leibel et Mekhnacha. Ces travaux ont servi de base à notre recherche. Une bibliographie plus complète peut être trouvée dans l’annexe A.

**Le chapitre 3** présente le dispositif expérimental utilisé pour la mise en œuvre de notre approche. Ce dispositif est composé de deux robots : un robot cartésien équipé d’un système de vision stéréoscopique, et un bras manipulateur équipé d’une pince.

La dernière partie de ce chapitre décrit la tâche de “pick and place” que nous avons résolue dans les chapitres 4 et 5.

**Le chapitre 4** décrit plusieurs sous-programmes probabiliste. Ces sous-programmes seront utilisés dans le chapitre suivant. En particulier nous nous intéressons à l’utilisation de notre méthode pour contrôler les mouvements d’un bras manipulateur. Nous montrons que le processus de prise en compte d’incertitudes géométriques peut être utilisé pour la localisation d’un objet avec la vision stéréoscopique. La fin du chapitre décrit un processus d’inférence et d’apprentissage utilisé pour représenter les positions de prise d’un objet.

**Le chapitre 5** contient notre contribution sur la programmation bayésienne de robots.

Nous présentons une méthode de programmation de robots originale dite, de “programmation inverse”. Au contraire des approches classiques, nous spécifions ce que nous attendons de la valeur d’un capteur étant donnée une action. En utilisant l’inférence probabiliste nous arrivons à inverser le problème et inférer l’action du robot étant données les lectures des capteurs.

Nous présentons ensuite la tâche de “pick and place”. Comme nous l’avons déjà signalé, la distribution conjointe est définie à l’aide de distributions élémentaires correspondant à des sous systèmes ou à des sous tâches. Ces distributions sont décrites en détail dans le chapitre 4 et dans l’annexe B.

**Le chapitre 6,** présente nos conclusions.

**L’annexe A** présente d’autres travaux qui ont aussi inspiré nos recherches. Nous citons les auteurs suivants :

- Puget sur la prise en compte de l’incertitude géométrique en utilisant un graphe d’état,
- De la Rosa considère l’incertitude de position d’un robot mobile et propose une méthode de recalage pour la réduire,
- Ruiz dédie ses efforts à la représentation de la connaissance en utilisant des mélanges de distributions de probabilités normales et à la façon de les construire. Ce travail a été utilisé pour la détermination de la position de prise de l’objet.

**L’annexe B** décrit un ensemble de programmes probabilistes liés à la calibration.

**L'annexe C** décrit l'algorithme de Travis pour passer de la représentation *RGB* de la couleur à la représentation *HSV* de la couleur.

**L'annexe D** contient la description d'une méthode de tirage aléatoire en suivant une distribution de probabilités normale.

**L'annexe E** présente la méthode d'intégration de Monte-Carlo que nous avons utilisée.

**L'annexe F** est utilisé pour décrire l'algorithme génétique continu utilisé dans ce travail.

**L'annexe G** donne les principaux éléments mathématiques utilisés pour le raisonnement géométrique.



# Chapitre 1

## Éléments Mathématiques

Dans ce chapitre, nous présentons le cadre théorique des probabilités sur lequel notre travail est basé et introduisons le concept de programmation bayésienne. Dans un deuxième temps, nous illustrons ce concept sur un exemple de suivi de couleur par un robot cartésien.

### 1.1 Proposition logique

Une *proposition logique* est un énoncé qui ne peut avoir qu'une valeur de vérité à chaque instant. La valeur peut être *vraie* ou *fausse*, cette valeur de vérité doit être définie sans ambiguïté. Par exemple :

1. L'énoncé : "il fait chaud", n'est pas une proposition logique car elle n'est pas clairement définie. En effet, on ne sait pas la température au-dessus de laquelle on considère qu'il fait chaud. Alors nous ne pouvons pas lui assigner avec certitude une valeur de vérité.
2. L'énoncé : "la température est au-dessus de 30 degrés", est une proposition logique car sa valeur de vérité est clairement définie.

Par la suite, nous allons noter les propositions logiques par des lettres minuscules.

#### 1.1.1 Opérations de base de propositions logiques

Les paragraphes suivants présentent une brève description des opérations de base sur les propositions logiques.

##### Produit logique

Il est aussi appelé *conjonction* et dénoté par  $a \wedge b$ , il est commutatif :  $a \wedge b = b \wedge a$ .

##### Somme logique

La *disjonction* ou somme logique notée  $a \vee b$  est commutative.



## Négation logique

Notée  $\neg a$ , elle indique la négation de la proposition logique  $a$ .

### 1.1.2 Probabilité de propositions logiques

La *plausibilité* d'une proposition est considérée dans son sens *subjectiviste* que nous interprétons comme la représentation d'un état de connaissance sur une proposition donnée. Les fondements formels de cette notion de *plausibilité* sont décrits en [BDL<sup>+</sup>99a, BDL<sup>+</sup>99b]. Ces fondements sont soutenus par une théorie du raisonnement probabiliste connue sous le nom *Probability as Logic* [Jay]. Dans cette théorie, le calcul de probabilités est considéré comme une généralisation de la logique formelle. La notion intuitive de plausibilité est formalisée par un ensemble d'axiomes. La probabilité découle mathématiquement de cet ensemble d'axiomes par le théorème de Cox [Cox46, Cox61, Cox78].

#### 1.1.2.1 Probabilité d'une proposition logique

Du cadre précédent, nous considérons la probabilité comme la plausibilité associée à une proposition. Pour fixer cette probabilité, il est nécessaire d'avoir un ensemble de connaissances préalables  $C$  sur le phénomène, c'est-à-dire la probabilité de la proposition logique  $a$  est conditionnée à l'ensemble de connaissances préalables  $C$  indiqué,

$$p(a|C)$$

Étant donné que l'ensemble des connaissances préalables  $C$  est toujours présent et que nous avons besoin d'utiliser la variable  $C$ , nous allons représenter la probabilité de  $a$  sachant l'ensemble des connaissances préalables  $C$  par :

$$p(a)$$

toutefois il faut bien noter que cet abus d'écriture ne doit jamais masquer que toutes les formules écrites dans ce document le sont aux vues de connaissances préalables.

#### Probabilité du produit logique

Elle est notée,

$$p(a \wedge b)$$

Elle représente la probabilité du produit des propositions logiques  $a$  et  $b$ . Cela se lit, "la probabilité de  $a$  et  $b$ ."

## Probabilité de la somme logique

L'expression ;

$$p(a \vee b)$$

représente la probabilité de la somme des propositions logiques  $a$  et  $b$ . Cela se lit, "la probabilité de  $a$  ou  $b$ ."

### 1.1.2.2 Probabilité conditionnelle d'une proposition logique

La probabilité donnée à la proposition logique  $a$  en fonction de la connaissance d'une deuxième proposition logique  $b$ , s'écrit :

$$p(a|b)$$

et se lit ; "la probabilité de  $a$  sachant  $b$ ". Elle dénote la valeur de probabilité de la proposition logique  $a$  sachant la valeur de vérité de la proposition logique  $b$ .

La généralisation de la probabilité conditionnelle d'une proposition logique  $a$ , conditionnée par la connaissance d'autres propositions logiques  $b_i$  avec  $i = 1..n$  s'écrit ainsi :

$$p(a|b_1 \wedge b_2 \wedge \dots \wedge b_n)$$

et se lit, "la probabilité de  $a$  sachant  $b_1$  et  $b_2$  et ... et  $b_n$ ."

## 1.1.3 Règles de calcul sur les propositions logiques

Le théorème de Cox donne les deux règles fondamentales du calcul probabiliste.

Soient deux propositions logiques  $a$  et  $b$ , ces règles sont :

### 1.1.3.1 Règle du produit

Elle est utilisée pour trouver la probabilité de la conjonction logique.

En prenant en compte la propriété de commutativité de la conjonction logique, la règle du produit a deux cas possibles :

$$\begin{aligned} p(a \wedge b) &= p(a) p(b|a) \\ &= p(b) p(a|b), \end{aligned} \tag{1.1}$$

dans la première ligne de l'équation (1.1) la probabilité de la conjonction  $a \wedge b$  est exprimée comme le produit de la probabilité de la proposition logique  $a$  multipliée par la probabilité conditionnelle de la proposition  $b$  en fonction de la valeur de vérité de la proposition  $a$ .

La deuxième ligne est obtenue en considérant que les propositions  $a$  et  $b$  jouent un rôle symétrique. Cette constatation triviale a cependant une conséquence importante : elle exprime la possibilité théorique d'inverser n'importe quel modèle probabiliste.

### 1.1.3.2 Règle de normalisation

Elle exprime le fait que la somme de la probabilité d'une proposition logique  $a$  et sa négation  $\neg a$  est toujours égale à l'unité,

$$p(a) + p(\neg a) = 1 \quad (1.2)$$

### 1.1.3.3 Probabilité de la disjonction

À partir des règles du produit (1.1) et de la normalisation (1.2), nous pouvons déduire la probabilité de la disjonction :

$$\begin{aligned} p(a \vee b) &= 1 - p(\neg a \wedge \neg b) \\ &= 1 - p(\neg a)p(\neg b|\neg a) \\ &= 1 - p(\neg a)[1 - p(b|\neg a)] \\ &= 1 - p(\neg a) + p(\neg a)p(b|\neg a) \\ &= p(a) + p(\neg a)p(b|\neg a) \\ &= p(a) + p(\neg a \wedge b) \\ &= p(a) + p(b)p(\neg a|b) \\ &= p(a) + p(b)[1 - p(a|b)] \\ &= p(a) + p(b) - p(b)p(a|b) \\ &= p(a) + p(b) - p(a \wedge b) \end{aligned}$$

Pour le cas où les propositions logiques  $a$  et  $b$  sont mutuellement exclusives. C'est-à-dire  $p(a \wedge b) = 0$ , l'expression devient :

$$p(a \vee b) = p(a) + p(b) \quad (1.3)$$

## 1.2 Variables

Il est plus courant d'utiliser des variables au lieu de propositions logiques. En conséquence, nous allons décrire dans les paragraphes suivants le concept de variable et les opérations associées. Pour les différencier des propositions, le nom d'une variable commencera par une lettre majuscule ; s'il est nécessaire d'utiliser plus d'une lettre, ce seront généralement des minuscules.

### 1.2.1 Variable aléatoire discrète

Par définition, une *variable aléatoire discrète*  $X$  est un ensemble de propositions logiques  $x_i$ , lesquelles sont :

– Mutuellement exclusives.

Nous appelons l'ensemble de propositions de  $X$  le *domaine de  $X$*  ; dénoté  $D_X$ .

$$\forall i, j \text{ avec } i \neq j, x_i \wedge x_j \text{ est faux,}$$

– exhaustives. À chaque instant au moins une des propositions logiques  $x_i$  est *vraie*. L'expression  $x_i$  indique que la variable  $X$  prend sa  $i^{\text{ème}}$  valeur.

Une variable ne peut prendre qu'une seule valeur de son domaine  $D_X$  à la fois.

Le *cardinal* de l'ensemble  $X$  dénoté  $[X]$  est le nombre de propositions logiques  $x_i$ .

### 1.2.1.1 Opérations sur les variables aléatoires discrètes

Les opérations pour les variables aléatoires discrètes sont les généralisations des opérations pour les propositions logiques.

#### Conjonction de variables

La *conjonction* de deux variables aléatoires discrètes  $X$  et  $Y$ , notée  $XY$ , est définie comme l'ensemble de  $[X] \times [Y]$  propositions logiques  $x_i \wedge y_j$ .

Étant donnée que la conjonction de variables aléatoires discrètes  $XY$  est un ensemble exhaustif de propositions mutuellement exclusives, elle est aussi une variable aléatoire discrète.

#### Disjonction de variables

La *disjonction* de deux variables aléatoires discrètes  $X$  et  $Y$ , dénoté  $X + Y$  est définie comme l'ensemble de  $[X] \times [Y]$  propositions logiques  $x_i \vee y_j$ . Cependant, la *disjonction* n'est pas une nouvelle variable car les  $[X] \times [Y]$  propositions logiques  $x_i \vee y_j$  ne sont pas mutuellement exclusives.

### 1.2.1.2 Distribution de probabilité

Nous avons établi la probabilité d'une proposition logique, cependant une variable aléatoire discrète représente un ensemble de propositions logiques. Nous voulons assigner une valeur de probabilité pour chacune des diverses valeurs de la variable.

Soit la variable aléatoire discrète  $X$  avec le domaine  $D_X$ . Pour chaque valeur  $x_i$  appartenant au domaine  $D_X$ , nous définissons une valeur  $p(X = x_i)$  où la fonction  $p$  représente la *distribution de probabilité* pour le domaine de la variable  $X$ . La *distribution de probabilité* de la variable  $X$  s'écrit :

$$p(X)$$

Les règles du produit (1.1) et de la normalisation (1.2) sont aussi définies pour les variables aléatoires discrètes.

### Règle du produit

Soient les deux variables aléatoires discrètes  $X$  et  $Y$  avec leurs domaines  $D_X$  et  $D_Y$  respectivement, la *règle du produit* pour  $X$  et  $Y$  s'écrit ainsi :

$$\begin{aligned} p(XY) &= p(X)p(Y|X) \\ &= p(Y)p(X|Y) \end{aligned} \quad (1.4)$$

### Règle de la normalisation

Pour la même variable aléatoire discrète  $X$ , la *règle de normalisation* s'écrit :

$$\sum_i p(x_i) = 1 \quad (1.5)$$

### Règle de la marginalisation

Elle est, en général, utilisée pour éliminer ou ajouter une variable dans un produit de variables, elle s'écrit,

$$p(Y) = \sum_X p(XY) \quad (1.6)$$

en utilisant la règle de normalisation, il est possible de déduire la règle de la marginalisation,

$$\begin{aligned} \sum_X p(XY) &= \sum_X p(Y)p(X|Y) \\ &= p(Y) \sum_X p(X|Y) \\ &= p(Y) \end{aligned}$$

#### 1.2.1.3 Densité de probabilité

Dans le cas d'une variable aléatoire définie sur un domaine continu, il n'est pas possible de parler de la probabilité d'une valeur de la variable aléatoire, car le nombre de valeurs est infini. Dans ce cas nous allons parler de sa *densité de probabilité*.

D'après [Jay], nous pouvons définir deux propositions logiques :

$$\begin{aligned} c &\equiv X \leq k \\ d &\equiv X > k, \end{aligned}$$

elles sont exhaustives et mutuellement exclusives. La probabilité de la proposition  $c$  est liée à  $k$  par une fonction  $G$  telle que :

$$p(c) = G(k). \quad (1.7)$$

Soit la proposition logique :

$$r \equiv (a < X \leq b), \quad (1.8)$$

Soient les propositions logiques  $s$  et  $t$ ,

$$s \equiv (X \leq a), \quad (1.9)$$

$$t \equiv (X \leq b), \quad (1.10)$$

nous avons :

$$t = s \vee r.$$

Étant donné que les propositions  $s$  et  $r$  sont mutuellement exclusives, nous pouvons écrire  $p(t)$  en utilisant l'expression de la probabilité de la disjonction de propositions logiques (1.3),

$$p(t) = p(s) + p(r)$$

alors,  $p(r) = p(t) - p(s)$ .

En utilisant les expressions (1.8), (1.9) et (1.10),

$$p(a < X \leq b) = p(X \leq b) - p(X \leq a).$$

De l'expression (1.7),

$$p(a < X \leq b) = G(b) - G(a).$$

Si la fonction  $G$  est différentiable, nous pouvons écrire :

$$p(a < X \leq b) = \int_a^b g(X) dX$$

avec,

$$g(X) = \frac{d}{dX} G(X).$$

La fonction  $g(X)$  est appelée la *densité de probabilité de X*. Dorénavant, elle sera notée  $p(X)$ . Elle a les propriétés suivantes :

- la fonction  $p(X)$  n'a pas de valeurs négatives,
- l'intégrale de la fonction  $p(X)$  est égale à 1 :  $\int p(X) dX = 1$ .

De la même façon que pour le cas discret, les règles du produit et de la normalisation sont aussi définies pour le cas continu.

### Règle du produit

Soient les variables aléatoires  $X$  et  $Y$  avec des domaines continus, la *règle du produit* pour  $X$  et  $Y$  s'écrit :

$$\begin{aligned} p(XY) &= p(X)p(Y|X) \\ &= p(Y)p(X|Y), \end{aligned}$$

la fonction de densité de probabilité  $p(XY)$  est connue comme la *fonction de densité de probabilité conjointe de  $X$  et  $Y$* .

### Règle de la normalisation

La *règle de la normalisation* pour  $X$  avec un domaine continu , s'écrit :

$$\int p(X)dX = 1,$$

impliquant que l'aire au-dessous de la fonction  $p(X)$  est égale à l'unité.

### Règle de la marginalisation

Elle est souvent utilisée pour éliminer ou ajouter une variable de la fonction de densité de probabilité conjointe. Elle s'écrit,

$$p(Y) = \int p(XY)dX,$$

elle est déduite en utilisant la règle de la normalisation,

$$\begin{aligned} \int p(XY)dX &= \int p(Y)p(X|Y)dX \\ &= p(Y) \int p(X|Y)dX \\ &= p(Y). \end{aligned}$$

## 1.3 Théorème de Bayes

Le *théorème de Bayes* [[Bay63](#), [Bre90](#), [Lap95](#)], fondement de notre travail, est déduit de façon directe de la règle du produit. Dans le cas discret nous l'écrivons :

$$\begin{aligned} p(X|Y) &= \frac{p(X)p(Y|X)}{p(Y)} \\ &= \frac{p(X)p(Y|X)}{\sum_X p(X)p(Y|X)} \end{aligned} \tag{1.11}$$

pour le cas continu,

$$\begin{aligned} p(X|Y) &= \frac{p(X)p(Y|X)}{p(Y)} \\ &= \frac{p(X)p(Y|X)}{\int p(X)p(Y|X)dX} \end{aligned} \quad (1.12)$$

## 1.4 Programme probabiliste

Le “Programme probabiliste” proposé par [Leb99] est une méthode formelle de description de systèmes et de problèmes [BDL<sup>+</sup>99a, BDL<sup>+</sup>99b, LDBM00, CB02, CFBM02, DL99, DL00].

Le programme probabiliste se compose de trois parties :

### 1.4.1 Description

La base de la programmation probabiliste est la *description*, elle est dénotée formellement par une “distribution de probabilité conjointe” définie à partir de connaissances préalables du problème et de l’expérience du programmeur. Elle inclut toutes les variables impliquées dans le problème.

Deux phases sont nécessaires à la construction d’une description :

1. la spécification de connaissances préalables  $C$  impliquant :
  - la définition des variables pertinentes,
  - la définition de la distribution de probabilité conjointe en un produit de distributions élémentaires de probabilités et,
  - la définition des formes paramétriques pour ces distributions élémentaires.
2. l’identification des paramètres : Celle-ci vise à instancier les paramètres laisser libre dans la phase précédente. Cette identification se fait à l’aide de données expérimentales. Par exemple, dans une distribution de probabilité normale, on doit fixer les valeurs de la moyenne et de l’écart type à partir d’un ensemble de mesures. Cette phase est aussi appelé “apprentissage”.

### 1.4.2 Question

Une question probabiliste se construit à partir d’une partition de l’ensemble de variables considérées en trois sous-ensembles :

1.  $E_q$  - représente l’ensemble de variables que l’on souhaite connaître (les inconnues); en général l’ensemble de variables motrices  $M$ .
2.  $E_c$  - représente l’ensemble de variables connues; en général l’ensemble de variables sensorielles  $S$ .



3.  $E_i$  - représente l'ensemble de variables cachées ou intermédiaires.

Dès lors une question est une expression de la forme :

$$p(E_q | E_c C_{\text{tâche}}) \quad (1.13)$$

où  $C_{\text{tâche}}$  représente la connaissance préalable sur le problème.

L'inférence consiste à obtenir une représentation pour cette distribution à partir de la distribution de probabilité conjointe :

$$p(E_q E_i E_c | C_{\text{tâche}})$$

Le théorème de Bayes et la règle de la marginalisation permettent de donner une solution triviale à ce problème :

$$\begin{aligned} p(E_q | E_c C_{\text{tâche}}) &= \frac{\sum_{E_i} p(E_q E_i E_c | C_{\text{tâche}})}{\sum_{E_q} \sum_{E_i} p(E_q E_i E_c | C_{\text{tâche}})} \\ &= \frac{1}{Z} \sum_{E_i} p(E_q E_i E_c | C_{\text{tâche}}) \end{aligned} \quad (1.14)$$

Cette expression représente la méthode générale pour répondre à une question probabiliste. Sous cette forme, elle est inutilisable car elle ne prend pas en compte les simplifications permettant de diminuer le nombre de sommes nécessaires à l'obtention du résultat.

### 1.4.2.1 Décision

L'expression (1.14), qui peut être calculée par le procédé d'inférence, est une distribution de probabilité sur les inconnues du problème. Pour avoir une solution, il faut choisir parmi ces valeurs possibles. Il y a diverses stratégies pour choisir une solution : le choix de la valeur la plus probable ou des heuristiques plus complexes impliquant la résolution de problèmes d'optimisation [Ber85].

La structure d'un programme probabiliste est donnée à titre d'exemple dans la figure 1.1. Cette structure sera utilisée tout au long de ce document pour expliciter les modèles probabilistes utilisés. Dans le paragraphe suivant, nous présentons un premier exemple d'un programme probabiliste.

## 1.5 Suivi de la couleur

Dans cette section, nous décrivons une expérience de suivi d'un objet par un robot cartésien (figure 1.2) à cinq degrés de liberté équipé d'une caméra couleur (une description du robot se trouve dans la section 3.1). La caméra est utilisée pour suivre un objet de couleur tenu par le bras manipulateur ou par l'expérimentateur.

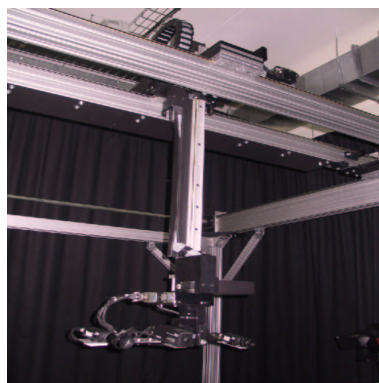
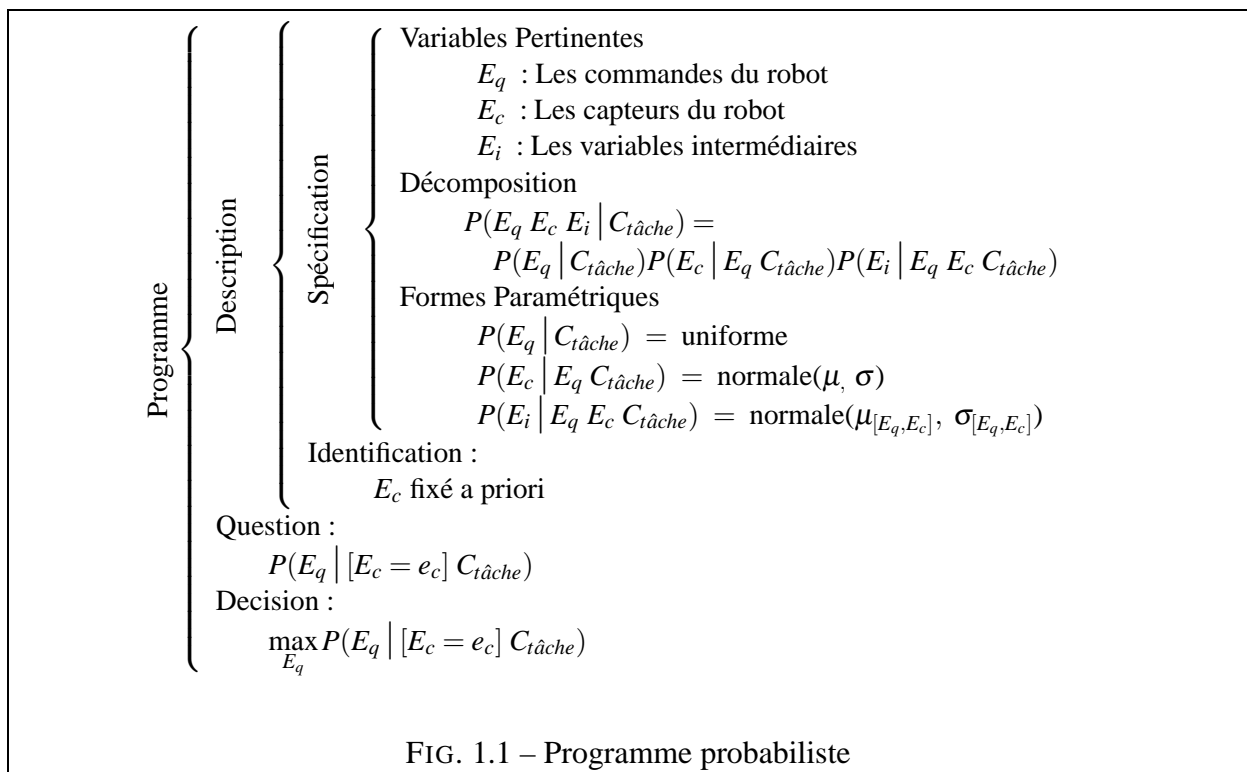


FIG. 1.2 – Robot cartésien

Dans cette expérience, la couleur de l'objet est représentée au moyen de distributions de probabilités, le robot est commandé en vitesse ; le choix de la vitesse est fait avec des tirages aléatoires sur des distributions de probabilités qui la représente.

La description de l'expérience comporte trois parties :

1. la représentation de la couleur,
2. la détection de la position de l'objet dans l'image,
3. la poursuite visuelle.

Nous allons décrire en détail chacune des parties dans les paragraphes suivants.

### 1.5.1 Représentation de la couleur

Il existe divers systèmes pour représenter la couleur ; le système *RGB* est un système d'addition de couleur basé sur la théorie tri-chromatique. Il est facile à utiliser cependant, il dépend du dispositif et de la spécification des couleurs. En utilisant le système *RGB*, une couleur est représentée avec trois valeurs :

- *R* pour le niveau de rouge,
- *G* pour le niveau de vert et,
- *B* pour le niveau de bleu.

Le système *Hue-Saturation-Value (HSV)* de représentation de la couleur est similaire au système *RGB*. L'espace *HSV* se déduit par une transformation linéaire de l'espace *RGB* [Smi78]. La méthode de transformation de *RGB* en *HSV* (Voir annexe C) permet aussi la transformation d'une image en niveau de gris [FM99].

Nous avons choisi de représenter la couleur avec l'espace *HSV* car cette représentation fait une meilleure séparation des couleurs [FM99]. De cette façon, nous avons trois composantes : *H* (Hue), *S* (Saturation) et *V* (Value).

Étant données que les images prises par la caméra de couleur sont représentées avec le système *RGB*, nous faisons, avant de les utiliser, une transformation vers le système *HSV*.

### 1.5.2 Détection de la position d'un objet dans l'image

L'expérience consiste à déterminer la position d'une couleur dans l'image fournie par la caméra de couleur. C'est-à-dire que l'objet sera repéré par sa couleur.

Étant donné que le plus petit élément constitutif d'une image est le point, il faut d'abord déterminer la couleur de chaque point dans l'image. Ensuite, il faut déterminer la position de l'objet à partir de ces points.

#### 1.5.2.1 Appartenance d'un point à l'objet

Nous allons aborder le problème de l'appartenance d'un point à l'objet de façon probabiliste. Ainsi, nous définissons les variables pertinentes.

#### Variables pour déterminer l'appartenance d'un point à l'objet

- *H* représente la composante *Hue* de la couleur,

- $S$  est la composante *Saturation* de la couleur,
- $V$  représente la composante *Value* de la couleur,
- $E$  est une variable booléenne *vraie* si le point est de la couleur de l'objet.

En conséquence, la distribution de probabilités conjointe pour le problème de l'appartenance d'un point à l'objet s'écrit :

$$p(E H S V).$$

le choix d'une décomposition de cette distribution de probabilités est nécessaire pour représenter notre connaissance sur le problème.

### Décomposition pour déterminer l'appartenance d'un point à l'objet

En considérant le système de représentation de la couleur et en faisant l'hypothèse suivante : les composantes  $H$ ,  $S$  et  $V$  de la couleur sont mutuellement indépendantes entre elles sachant que le point a la couleur de l'objet.

On peut choisir la décomposition :

$$p(E H S V) = p(E) p(H|E) p(S|E) p(V|E), \quad (1.15)$$

cette décomposition représente une partie de notre connaissance sur la couleur d'un point. Elle est aussi une simplification du problème.

### Formes paramétriques pour l'appartenance d'un point à l'objet

Le processus de description du problème implique ensuite le choix de formes paramétriques pour chacun des facteurs de la décomposition. Les formes paramétriques :

- de la composante  $H$  de la couleur en sachant que le point est de la couleur de l'objet,

$$p(H | [E = \text{vrai}]) = \text{Normale}(\mu_H, \sigma_H).$$

Pour exprimer notre ignorance sur la valeur de  $H$  quand on ne connaît pas la couleur du point ou quand on sait que le point a une couleur différente de celle de l'objet,

$$p(H | [E = \text{faux}]) = \text{Uniforme},$$

- de la composante  $S$  de la couleur, elle est similaire à  $H$ ,

$$p(S | [E = \text{vrai}]) = \text{Normale}(\mu_S, \sigma_S),$$

$$p(S | [E = \text{faux}]) = \text{Uniforme},$$

- de la composante  $V$  de la couleur, elle est similaire à  $H$  et à  $S$ ,

$$p(V | [E = \text{vrai}]) = \text{Normale}(\mu_V, \sigma_V),$$

$$p(V | [E = \text{faux}]) = \text{Uniforme},$$

– de la couleur du point,

$$p(E) = \text{Uniforme},$$

indique le fait que nous n'avons aucun *a priori* sur la couleur du point.

Les paramètres des distributions normales (la moyenne et l'écart-type) sont tirés d'un ensemble de points représentatifs de la couleur de l'objet. On calcule la moyenne et l'écart-type de chaque composante de la couleur.

Le cadre de connaissance, établi par la décomposition et les formes paramétriques de l'appartenance d'un point à l'objet est noté :  $c_{couleur}$ , il nous permet de faire de l'inférence probabiliste au moyen de questions sur les variables impliquées dans la distribution de probabilité conjointe.

### Inférence probabiliste pour la couleur d'un point

Nous voulons savoir si un point de l'image a la couleur de l'objet à détecter. De façon probabiliste nous allons définir la question suivante :

*Quelle est la probabilité d'un point d'avoir la couleur de l'objet sachant les composantes de la couleur du point ?*

Cela s'écrit,

$$p([E = \text{vrai}] | H S V c_{couleur})$$

En utilisant l'expression (1.15) de la décomposition et en appliquant le théorème de Bayes :

$$\begin{aligned} p(E | H S V c_{couleur}) &\propto p(E H S V c_{couleur}) \\ &\propto p(E | c_{couleur})p(H | E c_{couleur})p(S | E c_{couleur})p(V | E c_{couleur}) \quad (1.16) \end{aligned}$$

### Solution pour la couleur d'un point

La question définie par l'expression (1.16) est utile pour calculer la probabilité d'un point d'avoir la couleur de l'objet. Nous utilisons cette valeur pour indiquer le degré de certitude sur la couleur du point.

Il est important de considérer ce degré de certitude car, de cette façon tous les points seront pris en compte même, s'ils n'ont pas tout à fait la couleur de l'objet. Les points qui ont des petites variations de la couleur seront pris en compte avec un degré de certitude inférieur aux points qui n'ont pas ces sortes de problèmes.

### 1.5.2.2 Détection de la position de l'objet dans l'image

Le but est de repérer un objet représenté par une couleur dans l'image de la caméra. Nous allons calculer le centre de gravité de la couleur représentant l'objet. Nous profitons de l'information que nous avons sur chaque point de l'image :

- les coordonnées  $(u, v)$  de chaque point et,
- la probabilité  $p(E_{(u,v)} | H_{(u,v)} S_{(u,v)} V_{(u,v)} c_{couleur})$  du point d'avoir la couleur de l'objet.

Pour trouver les coordonnées  $O = (O_u, O_v)$  dans l'image du centre de gravité de la couleur, nous faisons les sommes pondérées par la probabilité des coordonnées des points multipliées par leurs probabilités, ainsi :

$$O_u = \frac{1}{W} \sum_{u=1}^N \sum_{v=1}^M u * p([E_{(u,v)} = vrai] | H_{(u,v)} S_{(u,v)} V_{(u,v)} c_{couleur}) \quad (1.17)$$

$$O_v = \frac{1}{W} \sum_{u=1}^N \sum_{v=1}^M v * p([E_{(u,v)} = vrai] | H_{(u,v)} S_{(u,v)} V_{(u,v)} c_{couleur}) \quad (1.18)$$

avec,

- $O$  - représentant la position de l'objet dans l'image,
- $O_u$  - l'abscisse de la position de l'objet,
- $O_v$  - l'ordonnée de la position de l'objet,
- $N$  - le nombre de colonnes de l'image,
- $M$  - le nombre de lignes de l'image,
- $(u,v)$  - la référence de la position  $(u, v)$  d'un point dans l'image,
- $E_{(u,v)}$  - variable booléenne *vraie* si le point  $(u, v)$  dans l'image a la couleur de l'objet,
- $H_{(u,v)}$  - la composante  $H$  de la couleur du point  $(u, v)$ ,
- $S_{(u,v)}$  - la composante  $S$  de la couleur du point  $(u, v)$ ,
- $V_{(u,v)}$  - la composante  $V$  de la couleur du point  $(u, v)$  et,
- $W$  - la somme de la probabilité de tous les points de l'image. Nous pouvons la voir comme une constante de normalisation nécessaire pour trouver la place du centre de gravité de la couleur dans l'image. Son expression mathématique est :

$$W = \sum_{u=1}^N \sum_{v=1}^M p([E_{(u,v)} = vrai] | H_{(u,v)} S_{(u,v)} V_{(u,v)} c_{couleur}).$$

On peut remarquer que l'approche probabiliste permet de prendre en compte tous les points même avec une faible probabilité.

### 1.5.3 Poursuite visuelle

Le but de cette expérience est de suivre un objet qui sera repéré par sa couleur. On va le faire en bougeant le robot de façon à avoir l'objet au centre de l'image de la caméra. La technique que nous utilisons consiste à placer le centre de gravité de l'image de l'objet au centre de la caméra :

c'est une poursuite en deux dimensions, car on utilise directement les informations visuelles extraites de l'image. C'est-à-dire qu'il n'y a pas de reconstruction tridimensionnelle de la scène.

Le robot cartésien (figure 1.2) utilisé pour cette expérience a cinq degrés de liberté dont trois sont des déplacements linéaires selon les axes  $X$ ,  $Y$  et  $Z$  et les deux autres sont des mouvements de rotation : *tangage* et *roulis*.

En utilisant cette architecture (voir section 3.1.1) nous pouvons faire deux asservissements visuels du même type, le premier en bougeant sur les axes cartésiens  $X$  et  $Z$  du robot et le deuxième avec les axes de *tangage* et de *roulis*.

Nous allons aborder le problème de l'asservissement visuel de façon probabiliste.

### 1.5.3.1 Variables

Le choix des variables impliquées dépend de nos connaissances préalables sur le problème. Pour notre cas, nous savons localiser un objet dans l'image, nous savons aussi que le but est de placer l'objet au centre de l'image et nous choisissons de le faire en se déplaçant sur les axes cartésiens  $X$  et  $Z$ .

Nous considérons que les mouvements des moteurs sont mutuellement indépendants en fonction de la connaissance de la position de l'objet. Cette hypothèse nous conduit à définir les variables suivantes :

- $O$  - représente l'objet en coordonnées de l'image, ainsi  $O = (o_u, o_v)$ ,
- $I_u$  - la coordonnée  $u$  du centre de l'image,
- $I_v$  - la coordonnée  $v$  du centre de l'image,
- $D_u$  - la différence entre la coordonnée  $o_u$  de l'objet et la coordonnée  $I_u$  du centre de l'image,
- $D_v$  - la différence entre la coordonnée  $o_v$  de l'objet et la coordonnée  $I_v$  du centre de l'image,
- $V_x$  - la vitesse du moteur qui commande l'axe  $X$ ,
- $V_z$  - la vitesse du moteur qui commande l'axe  $Z$ .

Étant donné que les mouvements sur chaque axe sont considérés indépendants, nous allons nous intéresser au contrôle du moteur de l'axe  $X$  ainsi, nous allons utiliser les variables  $o_v$ ,  $I_v$ ,  $D_v$  et  $V_x$ . Le reste des variables sera utilisé pour contrôler de façon similaire le moteur de l'axe  $Z$ .

La distribution de probabilité conjointe que nous allons étudier est donc :

$$p(V_x, o_v, I_v, D_v)$$

### 1.5.3.2 Décomposition

En prenant en compte les considérations qui ont été faites dans le paragraphe précédent sur les commandes des moteurs, nous choisissons la décomposition :

$$p(o_v I_v D_v V_x) = p(o_v) p(I_v) p(D_v | o_v I_v) p(V_x | D_v) \quad (1.19)$$

nous avons supposé que  $V_x$  est indépendant de  $o_v$  et  $I_v$  en connaissant  $D_v$ . En fait,  $V_x$  dépend de  $o_v$  et  $I_v$  au moyen de  $D_v$  car  $D_v = |I_v - o_v|$ .

### 1.5.3.3 Formes paramétriques

Les formes paramétriques, associées aux facteurs de la décomposition, représentant notre connaissance sur la poursuite visuelle sont :

- $p(o_v) = \text{Uniforme}$  - car l'objet peut être placé n'importe où dans l'image,
- $p(I_v) = \delta_c(I_v)$ <sup>1</sup> - Indique le fait que la coordonnée  $v$  du centre de l'image est définie sans ambiguïté,  $c$  est constante,
- $p(D_v | o_v I_v) = \delta_{\|I_v - o_v\|}(D_v)$  - Indique que la distance entre le centre de l'image est l'objet n'a pas d'incertitude car, dans ce cas nous considérons que le calcul est exact,
- $p(V_x | D_v) = \text{Normale}(\mu_{V_x}, \sigma_{V_x})$  - Indique que la vitesse du moteur a une distribution normale qui est fonction de la distance entre la position de l'objet dans l'image et le centre de l'image. Les paramètres de la distribution normale sont définis par le programmeur. Nous définissons  $\mu_{V_x}$  en fonction de l'accélération  $a$  du moteur et de la distance  $D_v$  [KD99] à parcourir ainsi :

$$\mu_{V_x} = \sqrt{2a D_v}$$

car de cette façon nous prenons en compte les conditions physiques du moteur. L'accélération est considérée constante pour cette expérience. L'écart-type de la distribution normale est défini par l'utilisateur et reste constant.

Les formes paramétriques et la décomposition définissent notre modèle de connaissance sur la poursuite visuelle. Celle-ci est notée :  $c_{ass\_visuel}$ .

### 1.5.3.4 Inférence probabiliste

L'inférence probabiliste se fait en définissant un problème au moyen d'une question. Pour la commande du moteur, la question est la suivante :

*Quelle est la vitesse du moteur connaissant la position de l'objet sur l'image de la caméra ?*

La question est représentée,

$$p(V_x | o_v i_v c_{ass\_visuel}) \quad (1.20)$$

---

<sup>1</sup>  $\delta$  représente un dirac défini :  $\delta_c(I_v) = \begin{cases} 1 & \text{si } I_v = c, \\ 0 & \text{sinon.} \end{cases}$



De l'expression (1.19) de la décomposition et en appliquant le théorème de Bayes :

$$p(V_x | o_v i_v c_{ass\_visuel}) \propto \int_{D_v} p(o_v) p(i_v) p(D_v | o_v i_v) p(V_x | D_v) dD_v \quad (1.21)$$

En utilisant le fait que  $p(i_v)$  et  $p(D_v | o_v i_v)$  sont des diracs, nous obtenons :

$$p(V_x | o_v i_v c_{ass\_visuel}) \propto p(o_v) p(V_x | [D_v = \|i_v - o_v\|]) \quad (1.22)$$

L'expression (1.22) représente l'espace des solutions pour la vitesse du moteur. Il faut donc en choisir une.

### 1.5.3.5 Solution

Dans cet exemple, nous avons choisi le tirage aléatoire sur la distribution de probabilité représentée par (1.22) pour choisir une solution.

Le tirage aléatoire permet de choisir une vitesse chaque dixième de seconde, cette vitesse peut varier entre deux choix successifs correspondants à une même lecture-captur. Le choix aléatoire permet de tenir compte de l'ensemble de connaissances à un instant donné.

### 1.5.3.6 Discussion

Nous voyons qu'il est possible de programmer un comportement de robot uniquement avec le principe probabiliste et le théorème de Bayes.

Nous constatons aussi que la connaissance préalable sur le problème joue de façon primordiale dans la définition probabiliste du problème. Elle permet de bien choisir la décomposition et de définir les formes paramétriques associées. En fait, le succès de la solution est complètement lié à cette connaissance préalable.

La programmation bayésienne est capable d'adapter le comportement du robot en changeant les paramètres des distributions de probabilité. Cela est le cas pour l'expérience de suivi de la couleur en se déplaçant sur les axes de *tangage* et *roulis*.

## 1.6 Conclusion

Nous avons présenté dans ce chapitre les éléments mathématiques utilisés tout au long de ce document. Puis, nous avons détaillé le principe de programmation probabiliste qui est central dans notre travail. Enfin, nous l'avons illustré sur un exemple de suivi de couleur.

Dans le chapitre suivant, nous décrivons des travaux sur l'approche probabiliste en robotique.

# Chapitre 2

## Contexte

Nous décrivons dans ce chapitre les travaux qui ont particulièrement inspiré notre travail. Ceux de Thrun qui montrent l'importance pratique du calcul probabiliste en robotique. Ceux de Lebetel qui portent sur l'utilisation d'un cadre probabiliste formel et systématique de programmation de robots et, ceux de Mekhnacha qui portent sur l'utilisation du formalisme bayésien pour la manipulation et la représentation des incertitudes géométriques. Une étude bibliographique plus complète est présentée dans l'annexe A. Elle concerne à la fois d'autres travaux portant sur l'utilisation des probabilités en robotique comme ceux de Puget [Pug89], De la Rosa [DLRR96] et une méthode de raisonnement probabiliste que nous utiliserons pour la saisie d'objets due à Ruiz [RdTG98].

### 2.1 Travaux de Thrun

Dans leurs principes les travaux de Thrun sont proches des nôtres. Ces travaux s'intéressent en particulier aux problèmes de perception et d'action pour des robots mobiles évoluant dans des environnements imparfaitement connus.

Pour lui, les robots actuels doivent être conçus pour aider les gens dans la vie courante [LDH98, RBF<sup>+</sup>00, SS00], ils doivent donc être prévus pour agir dans des environnements naturels et donc être capables de gérer les incertitudes inhérentes à ce type de situations.

Pour lui les sources d'incertitudes sont :

- l'environnement : comme par exemple les mouvements possibles de personnes autour des robots,
- l'information tirée des capteurs,
- le modèle du robot et des phénomènes physiques et,
- le calcul informatique lui même [TBB<sup>+</sup>00].

Selon Thrun : *L'approche probabiliste est mieux adaptée que les approches ignorant les incertitudes du robot* [Thr00]. Nous allons présenter les principaux concepts et méthodes utilisés par Thrun pour la localisation, le contrôle, la programmation et l'apprentissage des robots.

## 2.1.1 Localisation

La localisation est une composante clé pour les robots mobiles [BEF96, KBM]. Le problème revient à préciser la position du robot dans l'environnement à partir d'information capteur. Ce problème demeure quel que soit la nature des dispositifs employés.

Pour Thrun, le robot modélisant sa propre ignorance est la clé des robots autonomes [Thr00]. En estimant sa position par une distribution de probabilité, le système est mieux à même selon lui de prendre la décision adéquate.

La localisation du robot est donnée par une distribution de probabilité,

$$p(s_t | d_{0..t}, m)$$

où,

- $s_t$  est la position du robot à l'instant  $t$ ,
- $d_{0..t}$  sont les données acquises et les ordres envoyés par le robot depuis le début de l'expérimentation et,
- $m$  est l'*a priori* pour le monde.

La probabilité de  $s_t$  est recherchée en supposant que les données arrivent selon la séquence :

$$d_{0..t} = o_0, a_0, o_1, a_1, \dots, a_t, o_t$$

où les  $o_i$  représentent les observations, c'est-à-dire, les données tirées des capteurs comme des télémètres laser ou des images de caméras et les  $a_i$  représentent des données liées aux commandes. On peut formuler la localisation comme une chaîne de Markov et modéliser cette dernière à l'aide du théorème de Bayes. Cela donne :

$$\begin{aligned}
 p(s_t | d_{0..t}, m) &= p(s_t | o_0, a_0, o_1, a_1, \dots, a_{t-1}, o_t, m) \\
 &= \eta_t p(o_t | o_0, \dots, a_{t-1}, s_t, m) p(s_t | o_0, \dots, a_{t-1}, m) \\
 &= \eta_t p(o_t | s_t, m) p(s_t | o_0, \dots, a_{t-1}, m) \\
 &= \eta_t p(o_t | s_t, m) \\
 &\quad \int p(s_t | o_0, \dots, a_{t-1}, s_{t-1}, m) p(s_{t-1} | o_0, \dots, a_{t-1}, m) d_{s_{t-1}} \\
 &= \eta_t p(o_t | s_t, m) \\
 &\quad \int p(s_t | a_{t-1}, s_{t-1}, m) p(s_{t-1} | o_0, \dots, o_{t-1}, m) d_{s_{t-1}} \\
 &= \eta_t p(o_t | s_t, m) \\
 &\quad \int p(s_t | a_{t-1}, s_{t-1}, m) p(s_{t-1} | d_{0..t-1}, m) d_{s_{t-1}} \tag{2.1}
 \end{aligned}$$

où  $\eta_t$  est une constante de normalisation.

Nous obtenons donc une expression probabiliste permettant de calculer la probabilité de l'état actuel du robot. Pour simplifier la notation nous utiliserons par la suite  $b_t(s_t)$  au lieu de  $p(s_t|o_0, a_0, o_1, a_1, \dots, a_t, o_t, m)$ . Ainsi, l'expression (2.1) s'écrit :

$$b_t(s_t) = \eta_t p(o_t | s_t, m) \int p(s_t | a_{t-1}, s_{t-1}, m) p(s_{t-1} | d_{0\dots t-1}, m) ds_{t-1} \quad (2.2)$$

Cette expression est connue dans la littérature sous le nom de : *Markov localisation* [BFHS96, FBT99, KCK96, KS96, SK95, Thr98b], elle peut aussi représenter l'équation d'actualisation du filtre de Kalman [Kal60], qui est historiquement la plus connue dans le domaine.

La mise en place de l'équation (2.2) implique la connaissance de deux densités de probabilité :

1.  $p(s_t | a_{t-1}, s_{t-1}, m)$ .
2.  $p(o_t | s_t, m)$ .

La première représente le modèle des conséquences des actions  $a_{t-1}$  sur la position du robot, on peut la voir comme la généralisation probabiliste de la cinématique du robot mobile. La deuxième représente le modèle probabiliste de perception [TBB<sup>+</sup>00]. La description détaillée de ces deux modèles se trouve en [FBT99].

Comme le dit Thrun, “choisir une bonne méthode de représentation pour les densités de probabilité est fondamental pour la conception d'algorithmes probabilistes” [TBB<sup>+</sup>00].

Pour des problèmes de grandes dimensions, l'idée de base est d'approcher la distribution de probabilité  $b(s)$ . Par exemple, on peut utiliser une méthode de *Monté-Carlo* où  $b(s)$  est approchée avec un ensemble pondéré d'échantillons. La distribution définie par les échantillons constitue une approximation de la distribution désirée.

On parle alors de la méthode de *localisation de Monte-Carlo* [BFHS96, FBT99, KCK96, KS96]. L'algorithme de localisation de Monte-Carlo commence avec un ensemble  $s_t$  de  $k$  échantillons créés à partir d'une loi uniforme : la valeur de chaque échantillon est donc de  $\frac{1}{k}$ , on nomme cette valeur le *facteur d'importance* [Rub88], l'algorithme met ensuite en place l'équation (2.2) pour construire un deuxième ensemble d'échantillons à partir de l'ensemble actuel, de la réponse à une action  $a_{t-1}$  et d'une observation  $o_t$  :

1. construire un échantillon aléatoire  $s_{t-1}$  à partir de  $b_{t-1}(s_{t-1})$  avec une probabilité obtenue des *facteurs d'importance*.
2. pour l'échantillon  $s_{t-1}$ , construire la position suivante  $s_t$  correspondant à la distribution de probabilité du modèle de mouvement  $p(s_t | a_{t-1}, s_{t-1}, m)$ .
3. donner le facteur d'importance pour cet échantillon en fonction de  $p(o_t | s_t, m)$  et ajouter le nouvel échantillon à l'ensemble représentant  $b_t(s_t)$ .
4. refaire les pas de 1 à 3,  $n$  fois. À la fin, il faut normaliser les facteurs d'importance dans le dernier ensemble d'échantillons représentant  $b_t(s_t)$ .

L'algorithme de la localisation de Monte-Carlo est réputé efficace [DBFT99, DFBT99, LV00, TFB00].

## 2.1.2 Contrôle

Contrôler les actions du robot dans un cadre incertain est en général mieux approprié [Thr00] mais donne lieu à des algorithmes complexes.

Nous examinons l'algorithme de décision probabiliste connue comme *l'algorithme de navigation côtière* [KCK96]. Le cadre mathématique pour cet algorithme comme beaucoup d'autres algorithmes probabilistes est connu sous le nom *de processus de décision de Markov partialement observable* [Mon82, SS73, Son78]. Il est conçu pour réagir de façon optimale en présence d'incertitude dans des tâches où il faut prendre plusieurs décisions pour arriver au but. Pour traiter le problème du choix d'une action, l'algorithme utilise une fonction de coût  $C(s)$  qui définit le but. Par exemple, si le but est une position particulière de l'espace, la fonction de coût peut prendre la valeur 0 en cet endroit et par ailleurs elle vaudra 1. Pour atteindre la position but, il est nécessaire d'effectuer plusieurs actions. L'objectif est de minimiser les coûts cumulatifs prévus. Cela s'exprime par :

$$J = \sum_{\tau=t+1}^{t+T} E[C(s_\tau)] \quad (2.3)$$

Cette expression indique que l'espérance du coût est prise sur tous les états futurs.

Le processus construit une fonction d'évaluation du coût cumulatif prévu, c'est-à-dire que si on a la distribution de probabilité  $b$  et la position de départ  $s$ , on exprime la fonction d'évaluation comme suit :

$$V(b) = \int \sum_{\tau=t+1}^{t+T} E[C(s_\tau|s_t)] b(s_t) d_s \quad (2.4)$$

La fonction peut être calculée en ajustant de façon recursive la valeur de  $b$  [SB98],

$$V(b) \leftarrow \min_a \int [V(b') + C(b')] p(b'|a, b, m) d_{b'} \quad (2.5)$$

De cette façon, on affecte à  $V(b)$  la valeur attendue pour l'état  $b'$  prévu, alors le coût actuel d'un état  $b'$  est calculé en intégrant sur tous les états  $C(b') = \int C(s') b'(s') d_{s'}$ . La distribution conditionnelle  $p(b'|a, b, m)$  représente l'espace prévu pour le prochain état. Elle est calculée ainsi :

$$p(b'|a, b, m) = \int p(b'|o', a, b, m) p(o'|a, b, m) d_{o'} \quad (2.6)$$

où  $p(b'|o', a, b, m)$  est un Dirac défini avec l'équation (2.2),

$$p(o'|a, b, m) = \int \int p(o'|s', m) p(s'|a, s, m) b(s) d_{s'} d_s \quad (2.7)$$

Maintenant, il est possible de trouver la politique optimale en cherchant l'action minimisant  $V(b)$ ,

$$\pi(b) = \underset{a}{\operatorname{argmin}} \int V(b') p(b'|a, b, m) d_{b'} \quad (2.8)$$

En fait, cette approche ne permet de traiter que des problèmes de taille moyenne dès lors que l'on discrétise  $s$  [KLC98]. Selon Thrun [TBB<sup>+</sup>00], ce problème est rendu moins crucial par la possibilité d'utiliser des valeurs approximatives pour  $b(s)$ .

### 2.1.3 Programmation et apprentissage

Pour Thrun, les architectures de commande de robots se divisent en deux catégories : celles basées sur la programmation et les autres sur l'apprentissage. L'objectif de Thrun est de proposer une méthode permettant d'intégrer harmonieusement ces deux extrêmes. Pour cela il propose un prototype de programmation de robot appelé *CES* (*C for Embedded Systems*).

**Types de données probabilistes :** Ils sont utilisés pour déclarer des variables représentant des distributions de probabilités sur les valeurs des variables. Par exemple, la variable  $x$  de type *probint* est une variable qui peut prendre des valeurs entières. À chaque valeur sera liée une probabilité. Le programmeur peut définir la granularité de la distribution.

Le système garantit que la somme des probabilités de toutes les valeurs d'une variable vaut 1 et que les valeurs de probabilité sont toujours positives.

**Constantes :** *CES* a diverses façons d'assigner des distributions de probabilité aux variables. Par exemple :

- un Dirac initialisé,  $x = 3.14$ ; veut dire que  $Pr(x) = 1$  si  $x = 3.14$  et  $Pr(x) = 0$  si  $x \neq 3.14$ ,
- une histogramme est défini,

$$x = \{\{1, 0.5\}, \{5, 0.3\}, \{10, 0.2\}\};$$

- indique  $Pr(1) = 0.5$ ,  $Pr(5) = 0.3$ ,  $Pr(10) = 0.2$  et  $Pr(x) = 0.0$  pour  $x \neq 1, 2, 10$ ,
- une distribution uniforme déclarée,

$$x = \text{UNIFORM1D}(0.0, 1.0);$$

- représente la distribution uniforme sur l'intervalle  $[0.0, 1.0]$ ,
- une distribution normale s'écrit,

$$x = \text{NORMAL1D}(0.0, 1.0);$$

c'est la distribution normale de moyenne 0.0 et d'écart-type 1.0.

**Opérations Arithmétiques :** les opérations arithmétiques sont faites sur les domaines des variables. Elles engendrent des nouvelles distributions.

Par exemple ; soient  $x$ ,  $y$  et  $z$  trois variables probabilistes. Les distributions de  $x$  et  $y$  sont :  $x = \{\{0, 0.5\}, \{3, 0.5\}\}$ ; et  $Pr(0 \leq y < 10) = 0.1$ ,  $Pr(y) = 0$  si  $y \notin [0, 10)$  respectivement.

L'instruction  $z = x + y$  génère une nouvelle distribution pour  $z$  où, ses valeurs sont toutes les sommes possibles des valeurs de  $x$  et  $y$ . La probabilité des valeurs de  $z$  est le produit de la probabilité marginale correspondante, alors la distribution de  $z$  résultant est  $Pr(0 \leq z < 3) = 0.05$ ,  $Pr(3 \leq z < 10) = 0.1$ ,  $Pr(10 \leq z < 13) = 0.05$  et  $Pr(z) = 0.0$  si  $z \notin [0, 13)$ .

**Convention de types :** *CES* donne la possibilité de passer d'un type probabiliste à un type conventionnel et vice-versa. Par exemple ; si  $x$  est de type flottant probabiliste et  $y$  est de type flottant,

$$x = (\text{probfloat})y;$$

implique que  $x$  devient un Dirac avec la valeur de vérité reçue de  $y$ .

**Indépendance :** pour faire les opérations, *CES* suppose toujours une mutuelle indépendance entre les variables.

**Opérateur Bayes :** cet opérateur dénoté  $\#$  est utilisé pour dénoter la conjonction de plusieurs variables probabilistes. Par exemple ; soient les variables probabilistes  $x$ ,  $y$  et  $z$ , alors  $x = y\#z$  produit la distribution  $Pr_x(a) = \eta Pr_y(a) Pr_z(a)$  où  $a$  appartient au domaine de  $x$  et de  $y$  et,  $\eta$  est un facteur de normalisation.

**Apprentissage :** dans *CES* le but est d'intégrer la programmation de robots et l'apprentissage. *CES* mélange donc la programmation traditionnelle avec l'apprentissage, cela veut dire que le programmeur commence avec un niveau élémentaire de fonctionnalité et laisse des trous dans le programme. Ces trous seront remplis au moment de l'apprentissage.

*CES* a des fonctions spécifiques d'apprentissage avec des paramètres de :

- mise au point : ils sont modifiés de façon automatique,
- contrôle : ils sont définis par le programmeur,
- contrôle interne comme le type de la fonction d'approximation, la dimension des paramètres d'entrée et de sortie où, le type de fonction peut être un réseau neuronal, une fonction polynômiale ou linéaire, etc. pour plus de détails lire [Thr98a].

## 2.1.4 Discussion

Les méthodes probabilistes proposées par Thrun sont dédiées aux problèmes de localisation, de contrôle, de programmation et d'apprentissage pour les robots mobiles. Thrun a la conviction que seule l'approche probabiliste propose une solution véritablement opérationnelle aux

problèmes se posant aujourd’hui en robotique [Thr00].

L’approche de Thrun souffre selon nous de plusieurs travers :

1. la distribution de probabilité  $b_t(s_t)$  peut devenir très complexe en fonction de nombre de capteurs et d’actions. À notre avis, seule l’explicitation d’hypothèses d’indépendance sur les modèles capteur peut permettre de combattre cette complexité.
2. le cas de variables probabilistes continues n’est pas considéré dans l’approche.

Le langage de programmation et d’apprentissage *CES* est une bonne idée cependant :

1. en général il n’y a pas besoin d’établir une relation parallèle entre les types de données standard et les types probabilistes.
2. les descriptions probabilistes sont très limitées et dans le cas général les distributions obtenues peuvent nécessiter une place mémoire peu réaliste en pratique.
3. l’opérateur Bayes semblant le plus important pour faire l’inférence probabiliste n’est pas suffisamment décrit et peut se montrer à l’usage explosif en temps de calcul.

Dans l’approche probabiliste, nous sommes d’accord avec Thrun pour affirmer que le choix de la représentation des distributions est important pour la programmation probabiliste [Thr00].

Nous sommes aussi d’accord avec lui pour dire qu’il faut prendre en compte l’incertitude inhérente aux systèmes physiques dans leurs modèles pour bien les gérer.

## 2.2 Travaux de Lebeltel

Les travaux de Lebeltel [Leb99] portent sur la *Programmation Bayésienne de Robots*, il utilise une démarche formelle et systématique pour traiter de l’incertitude qui, comme pour Thrun (voir la section 2.1), constitue la problématique principale de la programmation de robots dans des environnements naturels.

Selon [Leb99] la programmation probabiliste de robots consiste en trois phases :

1. la phase de spécification impliquant le choix de variables du problème, le choix d’une décomposition de la distribution de probabilité conjointe et les formes paramétriques associées aux facteurs de la décomposition choisie.
2. la phase d’identification consiste à déterminer les paramètres des formes paramétriques soit à partir des données expérimentales, soit fixés *a priori*.
3. la phase d’utilisation consistant à poser une question à la décomposition sur certaines de ses variables (les variables cherchées) en fonction d’autres (les variables connues). Le résultat de cette question est une distribution de probabilité sur les variables cherchées.



Ces phases déterminent le programme probabiliste décrit dans la section 1.4.

Pour montrer l'intérêt de la programmation probabiliste, nous présentons une expérience d'évitement d'obstacles avec le robot Khepera (tiré de [Leb99]). Dans cette expérience, le robot apprend tout d'abord à éviter des obstacles par une phase de téléopération. Dans un deuxième temps, il évite les obstacles de façon autonome.

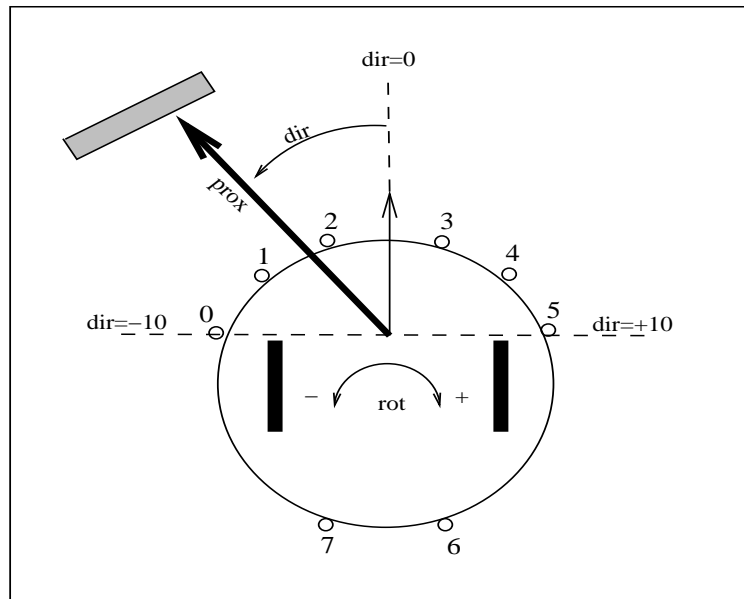


FIG. 2.1 – Pré-traitement des valeurs de capteurs, ici  $Dir = -4$  et  $Prox = 3$  (tiré de [Leb99])

**Description.** Le problème de contrôle du robot Khepera implique deux variables sensorielles obtenues au moyen d'un pré-traitement de l'information fournie par ses capteurs. Ces variables sont :

1.

$$Dir = Floor \left( \frac{90 * (Px5 - Px0) + 45 * (Px4 - Px1) + 5 * (Px3 - Px2)}{9 * (1 + \sum_{i=0}^5 Pxi)} \right)$$

indiquant la direction de l'obstacle par rapport au robot avec les valeurs entières entre  $-10$  (l'objet est à gauche) et  $+10$  (l'objet est à droite).

2.

$$Prox = Floor \left( \frac{\max(Px0, Px1, Px2, Px3, Px4, Px5)}{64} \right)$$

indiquant la proximité de l'obstacle avec les valeurs entières entre  $0$  (pas d'obstacle) et  $15$  (l'obstacle est très proche).

Les expressions précédentes considèrent les valeurs des six capteurs de proximité ( $Px0$ ,  $Px1$ ,  $Px2$ ,  $Px3$ ,  $Px4$ ,  $Px5$ ) du robot (figure 2.1).

À ces variables sensorielles il faut ajouter une variable motrice  $Vrot$  représentant la vitesse de rotation du robot. La vitesse de translation sera considérée comme constante pour simplifier l'exposé.

La distribution de probabilité conjointe pour le contrôle du robot est :

$$P(Prox Dir Vrot | C)$$

La décomposition choisie, en considérant que  $Prox$  et  $Dir$  sont mutuellement indépendants, s'écrit :

$$P(Prox Dir Vrot | C) = P(Prox | C) P(Dir | C) P(Vrot | Prox Dir C)$$

Les formes paramétriques pour les distributions élémentaires en prenant en compte qu'il n'y a pas d'*a priori* sur la direction et la proximité de l'obstacle sont :

$$P([Prox = prox_i] | C) = U_{kProx} = \frac{1}{16}$$

$$P([Dir = dir_j] | C) = U_{kDir} = \frac{1}{21}$$

La probabilité de  $Vrot$  en sachant  $Dir$  et  $Prox$  est représentée par :

$$P(Vrot | [Prox = prox_i][Dir = dir_j] C) = Normale_{\mu_{[prox_i, dir_j]}, \sigma_{[prox_i, dir_j]}(Vrot)}$$

Les paramètres  $\mu$  et  $\sigma$  sont calculés à partir de moyennes et d'écart types expérimentaux sur  $Vrot$  quand le robot est manoeuvré par téléopération. De fait, il y a une distribution normale  $Vrot$  pour chaque situation sensorielle : soit 21 (nombre de cas pour  $Dir$ ) multiplié par 16 (nombre de cas pour  $Prox$ ).

Pour chaque situation sensorielle, il faut retenir :

- le nombre  $n$  de fois où le robot se trouve dans cette situation,
- la somme  $S_1 = \sum Vrot$  des valeurs de  $Vrot$  quand le robot se trouve dans cette situation,
- la somme  $S_2 = \sum Vrot^2$  de carré de ces valeurs dans la même situation.

La mise à jour de ces valeurs est faite à chaque mesure capteur ; la moyenne et l'écart type sont facilement calculés par :

$$\mu[Dir, Prox] = \frac{S_1[Dir, Prox]}{n[Dir, Prox]}$$

$$\sigma[Dir, Prox] = \sqrt{\frac{S_2[Dir, Prox]}{n[Dir, Prox]} - \mu^2[Dir, Prox]}$$

Ces valeurs sont initialisées ainsi :  $n = 0.01$ ,  $S_1 = 0$  et  $S_2 = 0.36$ . Elles sont rapidement modifiées pendant le processus de la téléopération.

L'ensemble de ces connaissances est nommé  $Cp\_Obs$ .

**Question.** Pour faire l'inférence probabiliste on pose la question :

$$P(Vrot \mid Dir \ Prox \ Cp\_Obs)$$

Elle définit une distribution de probabilité sur  $Vrot$  en sachant la direction  $Dir$  et la distance  $Prox$  de l'obstacle.

**Choix d'une valeur pour  $Vrot$ .** Ce choix est fait par un tirage aléatoire sur la densité de probabilité marginale décrite par la dernière expression.

### 2.2.1 Discussion

La méthode de programmation de robots proposée par [Leb99] établit une démarche formelle et systématique en trois phases :

1. spécifier la connaissance préalable :
  - les connaissances préalables structurelles définissant l'ensemble de variables impliquées,
  - les connaissances préalables de dépendance décomposant la distribution conjointe en un produit de distributions élémentaires,
  - les connaissances préalables d'observation décrivant les formes paramétriques associées à chaque distribution élémentaire.
2. l'inférence probabiliste au moyen d'une question probabiliste laquelle va définir l'espace solution pour les inconnues du problème.
3. la sélection d'une solution au problème au moyen d'un tirage aléatoire sur la distribution de probabilité représentant l'espace de solutions.

Selon nous, une des contributions principales de ce travail est de fournir une méthode de spécification de tâches robotiques originale. Cette méthode est principalement déclarative et finalement ne considère en aucun cas l'architecture logicielle nécessaire à sa réalisation. Elle ouvre la voie à des architectures dynamiques s'établissant selon l'information disponible, la tâche à accomplir, la puissance du processeur et les hypothèses d'indépendance mutuelle pour le choix de la décomposition. Cependant nous adressons deux remarques principales à ces travaux :

1. sa méthode ne travaille que sur des espaces discrets de dimension fixe. À notre avis, une amélioration possible est de pouvoir dynamiquement changer le pas de discrétisation d'une variable.

2. les expériences réalisées ne manipulent que deux variables motrices, à notre avis la méthode peut s'appliquer sans problème à des systèmes plus complexes.

Nous allons retenir l'idée de programme probabiliste dans notre travail. Notre objectif étant de montrer que ce concept peut s'appliquer à des systèmes sensori-moteur beaucoup plus complexes.

## 2.3 Travaux de Mekhnacha

Les travaux de [Mek99] portent sur l'utilisation du formalisme bayésien pour la manipulation d'incertitude en géométrie. Ces travaux sont bien adaptés aux problèmes géométriques se posant dans les systèmes de CAO-Robotique et notamment ceux reliés au calibrage. Ils permettent de prendre en compte les incertitudes liées à la dimension des composants du robot, des objets impliqués dans la scène et ils considèrent aussi l'incertitude sur l'information acquise par les capteurs.

Nous reprendrons à notre compte ses concepts dans nos calculs géométriques pour la localisation d'objets. Nous allons illustrer ces méthodes de spécification et de résolution sur un exemple simple.

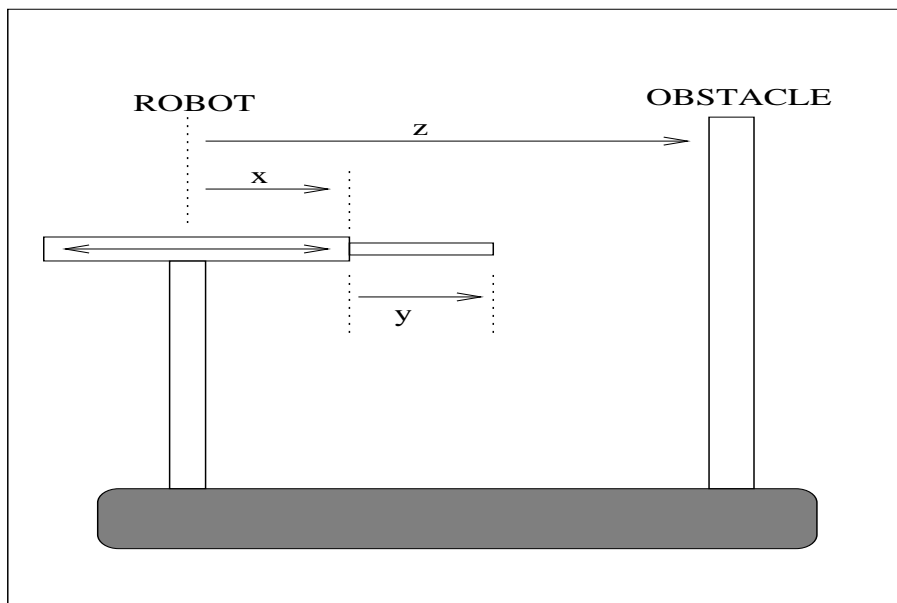


FIG. 2.2 – Robot avec un degré de liberté (tiré de [Mek99])

La figure 2.2 décrit un problème d'inversion cinématique élémentaire pour un robot planaire à un degré de liberté  $X$ . Son outil terminal a pour longueur  $Y$ . La tâche envisagée est de toucher

l'obstacle situé à une distance  $Z = z_0$  du centre du robot. Le problème est donc de trouver la valeur de  $X$  permettant de toucher l'obstacle.

### 2.3.1 Problème géométrique probabiliste et graphe cinématique

La méthode proposée par [Mek99] débute avec la description du problème géométrique au moyen d'un graphe orienté appelé graphe cinématique :

1. un sommet du graphe cinématique correspond à un repère géométrique du problème.
2. un arc exprime la relation géométrique existant entre deux repères. Il est labélisé par un vecteur de pose  $Q = (x, y, z, r_x, r_y, r_z)^T$  et est utilisé pour représenter la position relative entre les deux repères. Les trois premiers paramètres représentent la translation entre les deux repères impliqués et les trois derniers paramètres représentent la rotation.

[Mek99] justifie le choix du vecteur de pose pour représenter les relations géométriques par le fait qu'il permet facilement de représenter l'incertitude liée à la relation géométrique qu'il représente.

Dans le cadre de notre exemple, le graphe cinématique est donné (figure 2.3). Ici les vecteurs de pose n'ont qu'une dimension.

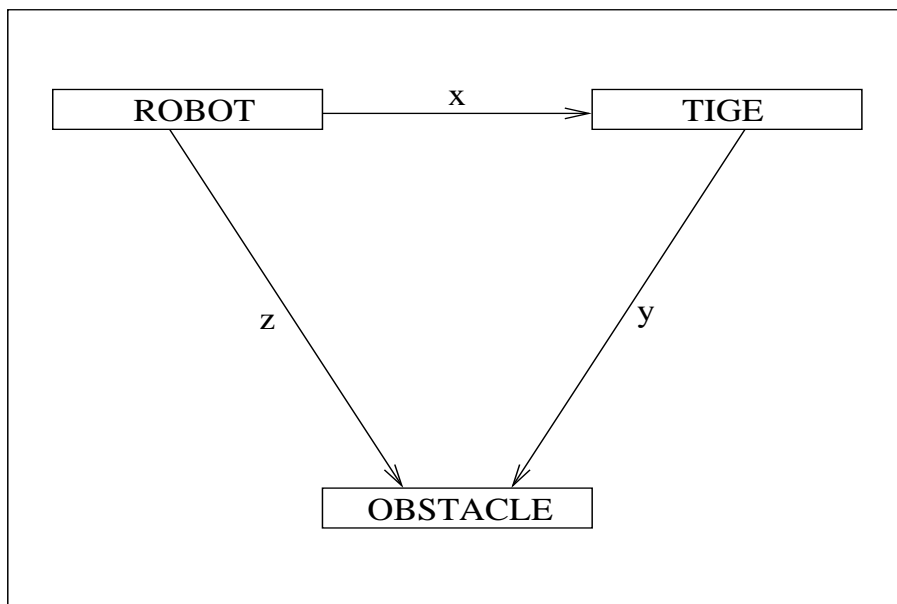


FIG. 2.3 – Graphe cinématique pour le problème d'inversion géométrique (tiré de [Mek99])

Dans ce graphe, on remarque la présence d'un cycle. Celui-ci traduit une contrainte géométrique globale. Cette contrainte représente la tâche à effectuer : on sait que, pour que la tâche

soit réalisée, la boucle cinématique formée par les liens  $X$ ,  $Z$  et  $Y$  doit être fermée. D'autres contraintes globales, dites de fermeture, peuvent apparaître dans un graphe cinématique, elles définissent d'autres tâches ou des contraintes d'ordre mécanique impliquées par la tâche.

En plus des contraintes globales de fermeture, [Mek99] introduit la notion de contrainte locale. Ces contraintes sont attachées cette fois à des arcs du graphe cinématique. Ces contraintes ont pour but de définir les conditions géométriques à réaliser pour satisfaire la tâche ou certaines contraintes physiques. Dans notre exemple, une contrainte locale pourrait traduire les butées mécaniques de l'axe  $X$ .

## 2.3.2 Construction d'un état de connaissance

[Mek99] construit un état de connaissance sous la forme d'une distribution de probabilité conjointe en utilisant un graphe cinématique.

À partir de la démarche de [Leb99] cet état de connaissance se construit de la façon suivante.

### 2.3.2.1 Choix des Variables

L'ensemble des variables se compose des :

1. variables des vecteurs de pose :

$$\{Q_{i_1j_1}, \dots, Q_{i_mj_m}\}$$

ces variables sont réelles.

2. variables booléennes de fermeture :

$$\{K_i\}_{i=1}^p$$

3. variables booléennes de contraintes locales

$$\{H_j\}_{j=1}^m$$

sur les paramètres des vecteurs de pose  $Q$ .

Ainsi pour construire l'état de connaissance correspondant à notre exemple nous devons définir :

1. les variables composant les vecteurs de pose :  $X, Y, Z$ .
2. la variable booléenne de fermeture correspondant au cycle du graphe :  $K$ .
3. la variable booléenne correspondant aux contraintes sur les butées mécaniques de l'axe  $X$  :  $H$

### 2.3.2.2 Définition de la décomposition

L'obtention de la distribution conjointe passe par la définition d'un produit de distributions élémentaires. Cette suite traduit une partie de notre connaissance sur la géométrie du problème. Dans notre cas, cette décomposition s'écrit :

$$p(X Y Z K H) = p(X) p(Y) p(Z) p(H|X) p(K|XYZ).$$

Parmi les connaissances préalables nécessaires à l'écriture de cette décomposition citons :

- l'indépendance des variables de pose en l'absence d'information sur la fermeture,
- l'indépendance des contraintes locales et des contraintes de fermeture.

### 2.3.2.3 Instanciation des distributions

Chaque élément de la décomposition doit être instancié par une distribution calculable. Le choix de ces distributions traduit lui aussi une partie des connaissances du problème ;

$$P(X) = \text{Uniforme}.$$

Cette affectation traduit notre ignorance sur la valeur de la variable articulaire.

$$P(Y) = \text{Normale}(\mu_y, \sigma_y) = \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{1}{2}\left(\frac{y-\mu_y}{\sigma_y}\right)^2}.$$

cette affectation traduit la précision d'usinage de l'outil du robot.

$$P(Z) = \text{Normale}(\mu_z, \sigma_z) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-\frac{1}{2}\left(\frac{z-\mu_z}{\sigma_z}\right)^2}.$$

cette affectation traduit la précision sur le positionnement de l'obstacle.

$$p([H = \text{vrai}] | X) = \begin{cases} 1 & \text{si } a \leq x \leq b, \\ 0 & \text{sinon.} \end{cases}$$

cette affectation traduit la contrainte sur les butées mécaniques. Où, la plage de valeurs pour l'axe  $X$  est compris entre  $a$  et  $b$ .

$$p([K = \text{vrai}] | X Y Z) = \begin{cases} 1 & \text{si } -Z + X + Y = 0, \\ 0 & \text{sinon.} \end{cases}$$

cette affectation traduit la contrainte sur la fermeture cinématique nécessaire à la réalisation de la tâche.

### 2.3.3 Résolution du problème

Partant de l'état de connaissance représenté par la distribution conjointe, on définit le problème suivant :

*“Quelle est la valeur la plus probable pour  $X$  pour que le robot touche l'obstacle ?”*

Pour répondre à cette question, on pose d'abord le problème d'inférence probabiliste en cherchant la distribution :

$$p(X|[H = vrai] [K = vrai]).$$

Puis on utilise une méthode de décision visant à considérer que la réponse à ce problème d'inversion est donnée par la valeur de  $X$  maximisant cette distribution.

$$x_{solution} = \max_x p(X|[H = vrai] [K = vrai]).$$

### 2.3.4 Méthodes de résolution

La première contribution de [Mek99] est de montrer que la plupart des problèmes géométriques rencontrés en robotique se ramènent à un double problème d'intégration et d'optimisation.

Dans le cas de notre exemple cela donne :

$$x_{solution} = \max_x p(X) \int \int p(Y) p(Z) p([H = vrai] | X) p([K = vrai] | X Y Z) dz dy$$

La deuxième contribution de [Mek99] est de montrer que ces problèmes peuvent être résolus par des méthodes numériques dans des cas pratiques. Pour cela il fait plusieurs remarques :

1. La présence de boucles cinématiques entraîne une réduction de la dimension de l'espace d'intégration. En effet, les boucles cinématiques permettent de calculer directement les valeurs de certaines variables d'intégration pour lesquelles la distribution conjointe n'est pas nulle. Dans notre exemple :

$$\int \int \frac{p(Y) p(Z)}{p([K = vrai] | X Y Z)} dz dy = \int \frac{p(Y) p(Z = X + Y)}{p([H = vrai] | X)} dy$$

2. la maximisation porte sur une fonction intégrale. Cette particularité permet à l'algorithme d'optimisation de “régler” la précision numérique du calcul de l'intégrale.
3. les contraintes locales peuvent servir de filtre pour les méthodes Monte-Carlo. En effet, les contraintes sur les butées mécaniques définissent les valeurs minimum et maximum de chaque variable.

Ces remarques ont amené [Mek99] à développer une méthode originale de résolution basée sur les algorithmes génétiques [Hol75, Muh92, Ala94, FB91, Har94] et les méthodes de Monte-Carlo [DBFT99, DGdF00, LC98, Nea93, TFB00].



### 2.3.5 Discussion

Le système proposé par [Mek99] est dédié aux problèmes de traitement d'incertitudes géométriques. Il utilise le formalisme bayésien pour la représentation et la manipulation de ces incertitudes dans des systèmes de Robotique et CAO-Robotique.

Cette approche prend en compte les incertitudes liées au modèle, aux capteurs et à l'environnement.

Le style de modélisation par spécification de contraintes sur les paramètres de pose et la représentation de l'incertitude par des distributions de probabilité nous paraissent bien adaptés.

L'utilisation des méthodes de Monte-Carlo pour l'intégration, imbriquées dans la méthode de maximisation, est une façon robuste et élégante pour obtenir une solution aux problèmes posés.

Nous avons adapté et utilisé ce système dans notre travail pour l'étalonnage de robots et la localisation d'un objet à partir de la vision stéréoscopique. Cependant nous pensons que la méthode de résolution utilisée (méthode de Monté-Carlo imbriquée dans un algorithme génétique) reste très lourde et la rend de fait inutilisable pour les applications temps réel.

## 2.4 Conclusion

Les travaux de [Leb99] et de [Mek99] portent sur l'utilisation de l'approche bayésienne en robotique.

Le premier est dédié à la programmation bayésienne de robots proposant une méthode de programmation de robots formelle et systématique. Il définit des programmes bayésiens allant du simple comportement réactif à de la reconnaissance d'objets par un robot mobile en passant par des programmes de fusion de données.

Il définit aussi un mécanisme de séquençement de tâches élémentaires pour réaliser une tâche complexe. Enfin, il donne des principes de construction d'architecture de contrôle pour robots. Pour démontrer la validité de sa proposition il l'applique à la programmation d'une tâche de surveillance d'un robot mobile.

Le deuxième dédié à la représentation et la manipulation des incertitudes géométriques dans des systèmes de robotique impliquant des modèles géométriques du système et de l'environnement. Il propose de représenter les paramètres des relations géométriques par des distributions de probabilité. Étant donnée cette spécification, une distribution de probabilité est construite sur l'ensemble des paramètres du modèle géométrique pour un problème donné. La distribution marginale sur les paramètres inconnus est inférée à l'aide des règles de la probabilité. Pour résoudre le problème posé par la distribution marginal, [Mek99] propose une méthode d'optimisation ori-

ginale et très puissante impliquant l'approximation d'intégrales par la méthode de Monte-Carlo.

Dans notre travail, nous reprenons partiellement les résultats de [Leb99] et de [Mek99] et les développons pour résoudre un problème plus complexe : un problème de saisie et de déplacement d'objets par un bras manipulateur. Des travaux de [Leb99], nous réutilisons le mécanisme de séquençement ainsi que la notion d'architecture que nous améliorons nettement. Des travaux de [Mek99], nous reprenons partiellement la méthode d'estimation des intégrales en proposant plusieurs adaptations plus adéquates à notre problème.

Dans le chapitre suivant, nous décrivons les robots et le système de vision utilisés et nous décrivons notre problème.



# Chapitre 3

## Description des robots

Dans les chapitres précédents, nous avons présenté les bases mathématiques nécessaires à notre approche et les travaux qui ont inspiré cette thèse.

Nous décrivons dans ce chapitre les robots utilisés pour réaliser une tâche de saisie et de déplacement d'objets, ainsi que les modèles les représentant.

La première partie du chapitre correspond au robot cartésien portant le système de vision stéréoscopique. Nous présentons son modèle géométrique car nous l'utilisons dans notre travail. Ensuite, le bras manipulateur est décrit ainsi que son modèle géométrique. Après, nous présentons la description de la pince du bras robot et nous justifions cette description du point de vue du système stéréoscopique. La fin du chapitre est dédiée à la description de la tâche de "pick and place".

### 3.1 Robot cartésien

Le robot cartésien (figure 3.1) a été construit par la société SINTERS en 1996. Il est constitué d'un cadre aluminium BOSH où il peut se déplacer sur trois axes cartésiens (translation), ces axes ont des servomoteurs CC de marque PARVEX (Gec Alsthom) couplés à des variateurs de marque TSA (TRM 6018). Il a été conçu pour porter trois caméras : deux noir et blanc et une couleur. Les caméras sont positionnées à un endroit où elles peuvent effectuer deux types de rotation. Les plateaux tournants permettant les mouvements de rotation des deux caméras sont de marque MICRO-CONTROLE et ils sont couplés à des variateurs de vitesse de marque INFRA-NOR (MS60).

Pour l'acquisition d'images, le robot cartésien est équipé avec une carte VME de Blue Wave Systems (ex LSI), avec un module MDC40ic et un module MD44. Cette carte se trouve dans le rack de contrôle du robot cartésien et permet l'acquisition couleur (R,G,B) ou noir et blanc (de une à trois images synchrones).

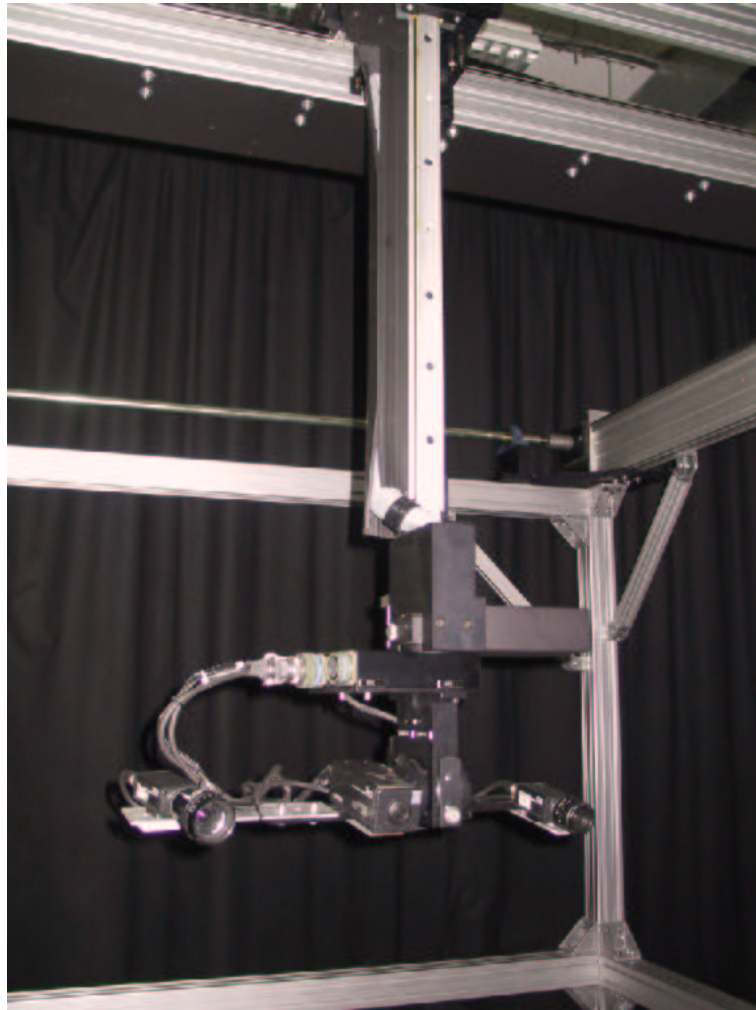


FIG. 3.1 – Robot cartésien

Les caractéristiques physiques du robot sont décrites dans le tableau (3.1). On trouve pour chacun des axes : la vitesse maximale et minimale de déplacement, la résolution du codeur et la course de déplacement.

La partie commande du robot est composée d'une carte VME MOTOROLA (MVME167) avec un processeur 68040 et une carte GREENSPRING (VIP610) en guise de IP-module. Le système se contrôle à partir de VxWorks.

Le robot occupe une place d'environ 2,5 mètres de côté par 2 mètres de hauteur. Il est placé en face du bras manipulateur RX90, décrit plus tard dans le chapitre.

ART	CARACTÉRISTIQUES			
	Course	Vitesse Maximale	Vitesse Minimale	Rés. Codeur (trait/tour)
<b>X</b>	2360 mm	60 cm/s	1 cm/s	500
<b>Y</b>	2000 mm	60 cm/s	1 cm/s	500
<b>Z</b>	800 mm	30 cm/s	1 cm/s	500
<b>Orientation</b>	+/- 170°	0.2 t/s	0.003 t/s	500
<b>Inclinaison</b>	+/- 170°	0.1 t/s	0.003 t/s	500

TAB. 3.1 – Spécifications du robot cartésien.

### 3.1.1 Architecture du robot cartésien

La figure 3.2 montre l'architecture physique du robot cartésien avec ses cinq degrés de liberté.

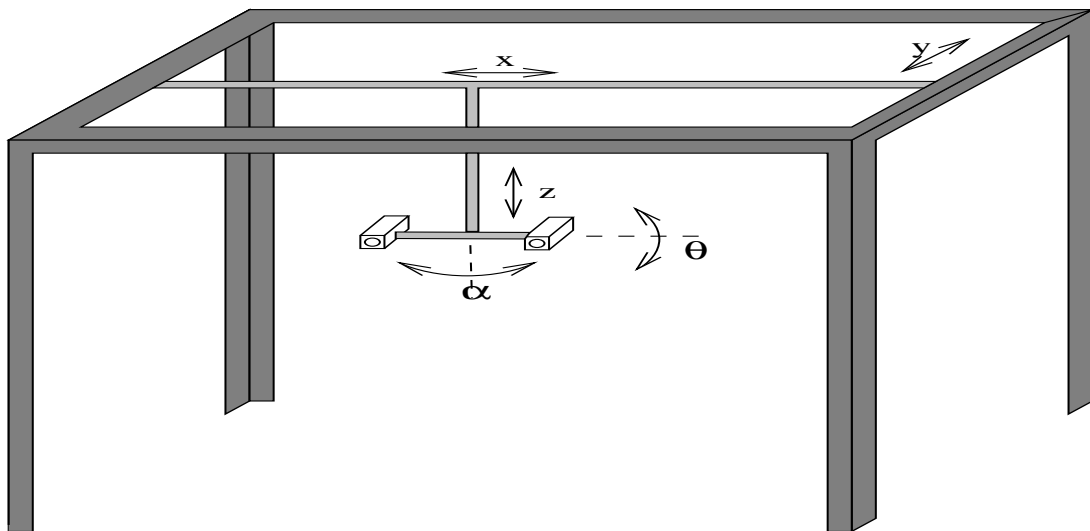


FIG. 3.2 – Modèle physique du robot cartésien

Nous appelons *configuration du robot* l'ensemble des valeurs de ses cinq axes.

La figure 3.3 présente le modèle physique du robot au moyen d'une chaîne cinématique impliquant des transformations géométriques entre les axes. Ces transformations sont représentées par des vecteurs de pose (voir annexe G.1.2). Elles nous permettent de connaître la position du dernier repère ( $RC_3$ ) du robot cartésien sur son repère de base ( $RC_0$ ).

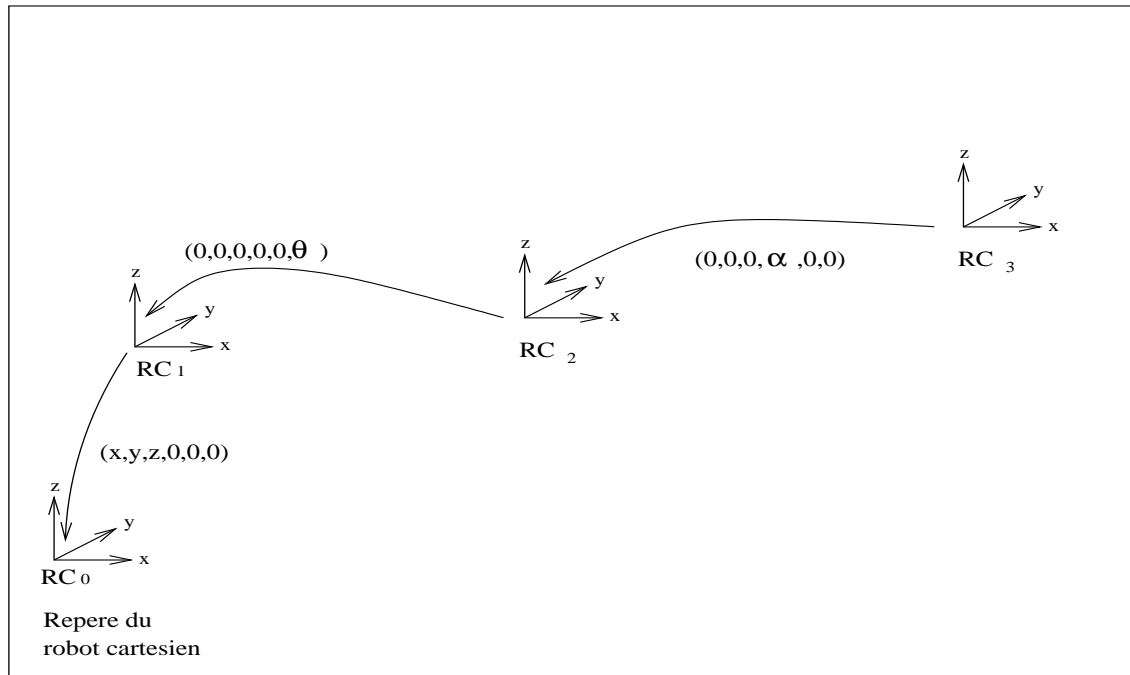


FIG. 3.3 – Chaîne cinématique du robot cartésien

### 3.1.2 Système de vision stéréoscopique

Le système de vision stéréoscopique comporte deux caméras noir et blanc (figure 3.4) placées sur le robot cartésien. La liaison physique entre les deux caméras est fixe, autrement dit, les caméras ne bougent pas l'une par rapport à l'autre. Elles peuvent prendre des images en 256 niveaux de gris d'une taille de 512x512 points. Elles ont les caractéristiques suivantes :

- résolution horizontale de 570TV lignes/560TV lignes,
- compactes et légères,
- haut sensibilité,
- fonction de redémarrer,
- fonction de retour à l'état initial.

L'acquisition d'image est commandée et fournit une image par caméra.

## 3.2 Bras robot

Le robot Stäubli modèle RX90 (figure 3.5) est un bras manipulateur à six degrés de liberté, ayant sur sa dernière articulation une pince avec deux doigts parallèles (figure 3.6) pour prendre des objets.



FIG. 3.4 – Caméras pour la capture d’images.

ART	CARACTÉRISTIQUE			
	Amplitude	Répartition Amp. (degrés)	Vitesse (degrés/s)	Résolution ang. (degrés)
A <sub>0</sub>	320	+/-160.0	236	0.00087
A <sub>1</sub>	275	+/-137.5	200	0.00087
A <sub>2</sub>	285	+/-142.5	286	0.00072
A <sub>3</sub>	540	+/-270.0	401	0.00101
A <sub>4</sub>	190	+100/-90	320	0.00117
A <sub>5</sub>	540	+/-270.0	580	0.00220

TAB. 3.2 – Spécifications du robot Stäubli Rx90.

Les six articulations rotoïdes du bras robot ont des caractéristiques différentes que l’on trouve dans le tableau (3.2). Il fournit, pour chacune des articulations, l’amplitude, la répartition de l’amplitude, la vitesse maximale et la résolution.

Par rapport à l’espace de travail, sans inclure la longueur de la pince, le rayon maximal est de 900 mm tandis que la minimale est 290 mm. Sa vitesse maximale cartésienne est de 150 cm/s.



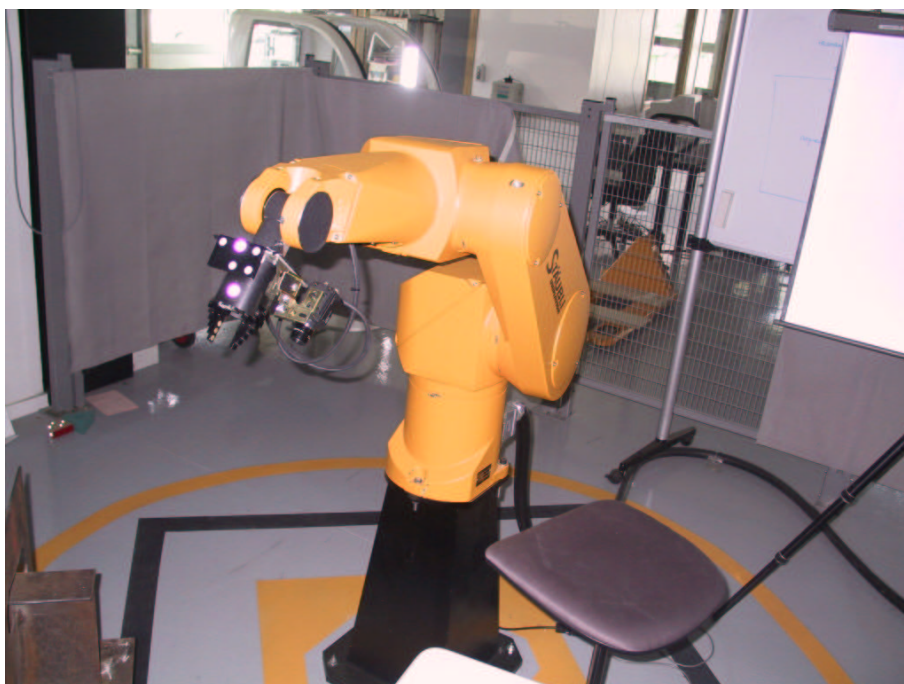


FIG. 3.5 – Bras robot stäubli RX90

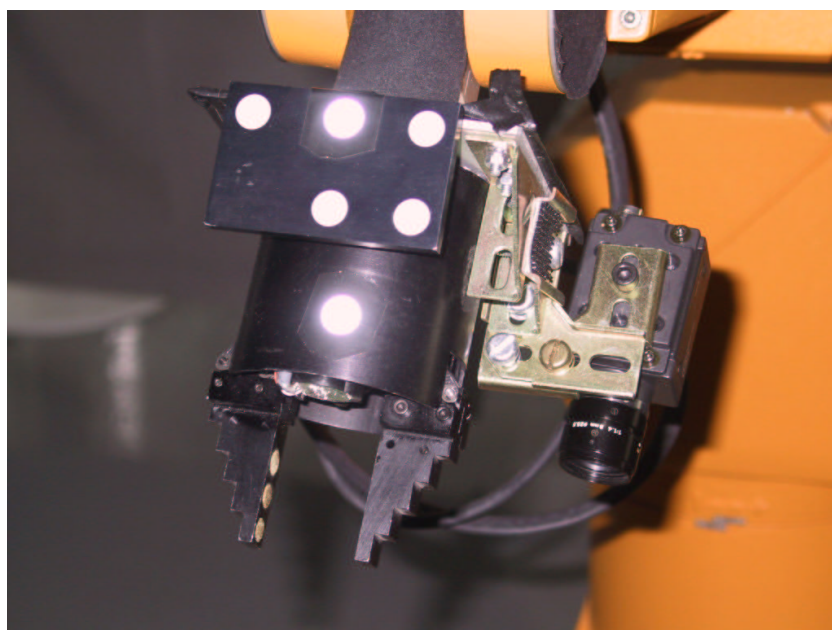


FIG. 3.6 – Pince du bras manipulateur stäubli RX90

Le système de contrôle a été développé par les moyens robotiques de l'INRIA Rhône Alpes. Les fonctions de contrôle sont accessibles en langage C impliquant la commande en position et la commande en vitesse.

La commande en position consiste à affecter les valeurs des articulations avec la position articulaire désirée. Par contre, la commande en vitesse consiste à affecter aux articulations la vitesse à laquelle elles doivent se déplacer.

### 3.2.1 Architecture du bras manipulateur

Le modèle physique du bras manipulateur est montré dans la figure 3.7, le modèle géométrique correspondant est représenté par la figure 3.8. C'est un graphe cinématique représentant les transformations géométriques ayant lieu dans le bras.

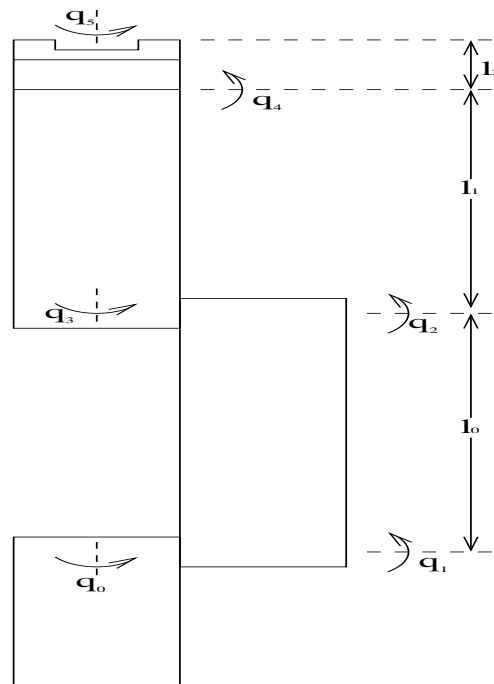


FIG. 3.7 – Modèle physique du bras manipulateur

Le modèle géométrique est utilisé pour calculer la position de la dernière articulation du bras (celle portant la pince) sur son repère de base (les détails de ce calcul sont abordés plus tard dans le document).

Le modèle du bras n'est pas suffisant pour définir la position de la pince car, il n'inclut pas les dimensions de la pince. Pour cela, nous allons définir un modèle de la pince.

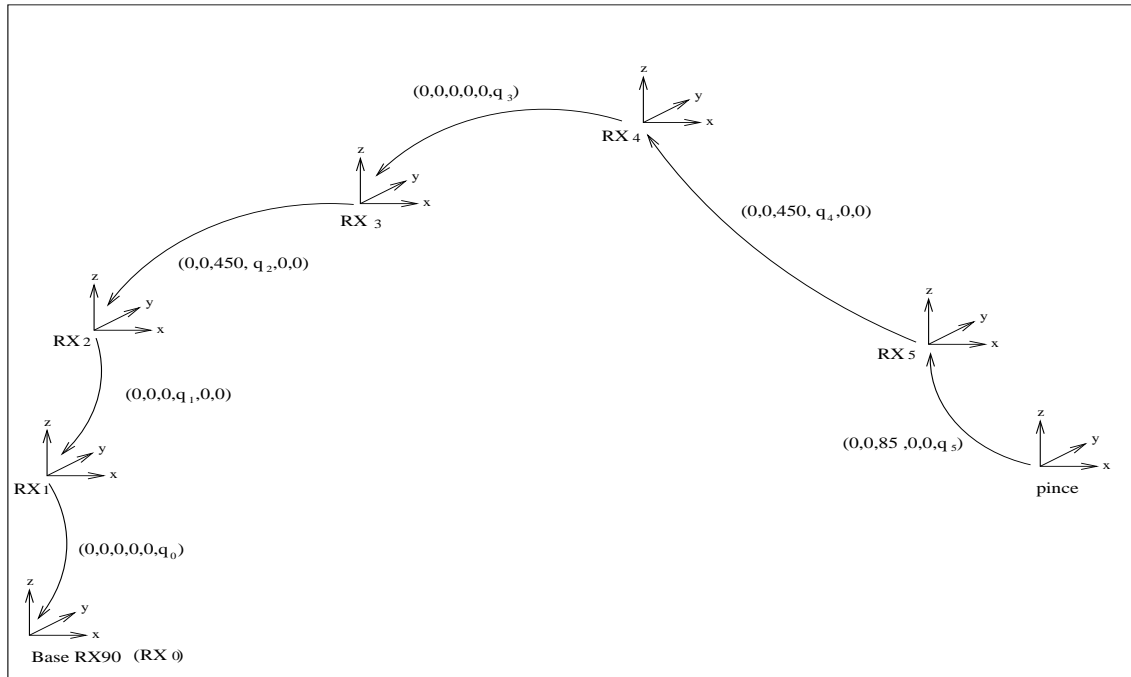


FIG. 3.8 – Chaîne cinématique du bras Stäubli

### 3.2.2 Pince du bras robot

Pour des raisons de simplicité de repérage des objets avec le système stéréoscopique, nous avons pris la décision de décrire la pince (figure 3.6) du robot au moyen de points caractéristiques [MCSP01] qui sont physiquement collés sur elle. Ces points sont en général appelés points de référence. Ils sont décrits par leurs coordonnées dans le repère propre à la dernière articulation du bras.

La figure 3.9 représente le modèle de la pince avec les cinq points de référence que nous avons considérés et les axes cartésiens.

point	x	y	z
0	63.5	-28.0	10.0
1	63.5	0.5	9.5
2	63.5	32.0	10.0
3	63.5	-28.0	40.0
4	63.5	0.0	40.0

TAB. 3.3 – Description de la pince.

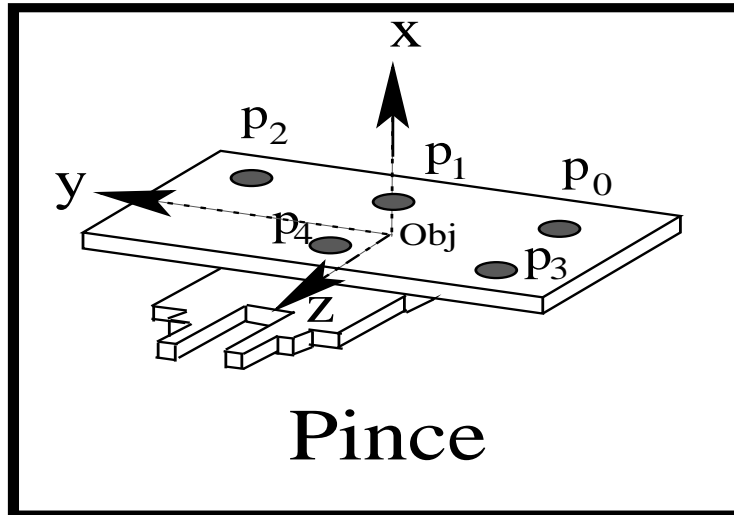


FIG. 3.9 – Modèle de la pince.

Le tableau (3.3) contient la description des cinq points de référence de la pince.

### 3.3 Description de la tâche de “pick and place”

Dans cette section, nous décrivons la tâche de “pick and place” à l’aide des deux robots présentés précédemment. En effet, c’est la tâche à laquelle nous nous sommes intéressés dans notre travail. Cette tâche consiste à saisir un objet avec le bras manipulateur et à le déposer à une position précise dans l’environnement : typiquement sur une table dont la position est connue.

Pour cela, le bras manipulateur enchaîne plusieurs opérations élémentaires. Initialement, le bras est immobile en attendant de l’information sur la position de l’objet à saisir. Au moment où il connaît la position de cet objet dans son espace de travail, il s’en approche pour le saisir. Ensuite, il ralentit pour positionner la pince. Puis, il s’arrête pour saisir l’objet en fermant la pince. Il se dirige vers la position où il faut poser l’objet. Il ralentit pour positionner la pince, puis s’arrête pour poser l’objet en ouvrant sa pince. Finalement, il vient se replacer dans une position d’attente.

Pour réaliser chacune de ces opérations élémentaires, le système robotique doit résoudre deux problèmes :

1. connaître la position de l’objet et la position de la table dans l’espace de travail. Ces positions sont obtenues par un traitement sur les images fournies par le capteur de vision stéréoscopique.
2. d’autre part, le bras doit planifier une trajectoire le menant de sa position courante à la position de l’objet ou à celle de la table.

Dans la suite de ce document, nous allons présenter comment nous avons utilisé la programmation probabiliste pour résoudre l’ensemble des problèmes constituant la tâche de “pick and

place”.

### 3.4 Conclusion

- Dans ce chapitre, nous avons décrit les deux robots utilisés dans nos expériences, à savoir :
- un robot cartésien équipé d’un système de vision stéréoscopique pour percevoir l’environnement,
  - un bras robot équipé d’une pince afin de saisir des objets, les déplacer et les déposer.

Nous avons défini les modèles géométriques des robots à partir de leurs modèles physiques et, nous avons défini un modèle de la pince au moyen des points de référence. Ces modèles seront utilisés pour que le bras soit capable d’interpréter l’information fournie par le système de vision stéréoscopique.

Pour finir, nous avons présenté la tâche de “pick and place”. Nous avons décrit cette tâche sous forme d’un enchaînement d’opérations élémentaires. De plus, nous avons remarqué que pour accomplir chacune de ces tâches nous devons être capables d’interpréter l’information fournie par le système de vision stéréoscopique, ainsi que de calculer des trajectoires permettant au bras de se déplacer dans l’espace de travail.

Dans les deux chapitres suivants, nous détaillons les programmes bayésiens nécessaires pour réaliser la tâche de “pick and place” d’un objet en utilisant ces dispositifs robotiques.

# Chapitre 4

## Sous-programmes probabilistes

Dans ce chapitre, nous décrivons certains des sous-programmes probabilistes qui seront mises en œuvre pour réaliser la saisie d'un objet à partir de la vision stéréoscopique. Ces programmes concernent :

1. le déplacement du bras manipulateur.
2. la localisation d'un objet.
3. la saisie de l'objet

D'autres sous-programmes probabilistes seront nécessaires pour réaliser cette saisie. Ils seront utilisés pour résoudre les problèmes de calibration suivants :

1. calibration d'une caméra.
2. calibration d'un système de vision stéréoscopique.
3. calibration d'un système multirobots.

Ces programmes sont décrits dans l'annexe B. Ils reprennent les principes utilisés dans ce chapitre.

### 4.1 Modèle de mouvement

Pour que le bras (figure 3.5) puisse saisir des objets avec sa pince, il faut qu'il soit capable de se déplacer de sa position initiale à la position où se trouve l'objet. Nous détaillons le problème dans la section suivante.

#### 4.1.1 Description du problème

Nous définissons :

- *l'espace opérationnel* comme l'espace de travail du bras. Dans cet espace nous utilisons les coordonnées cartésiennes.
- *l'espace articulaire* représente l'ensemble de valeurs des six articulations du bras.

Nous voulons commander en vitesse le bras pour aller d'un point de départ à un point d'arrivée. De plus, nous voulons que le bras soit capable de changer dynamiquement sa trajectoire. Pour cela, il faut calculer sa vitesse en un temps relativement bref. Pour notre problème, nous considérons que ce temps ne doit pas être supérieur à dix millièmes de seconde.

Notre objectif est donc de calculer les vitesses des six articulations du bras en un temps maximal de dix millièmes de seconde. Pour cela, nous allons décrire les différentes étapes du calcul à savoir :

1. la trajectoire idéale calculée en fonction de la position actuelle du bras et du but,
2. le temps de parcours calculé en fonction de la trajectoire idéale, de l'accélération et la vitesse maximale de mouvement du bras dans l'espace de travail,
3. la vitesse de parcours définie en fonction du temps et de la trajectoire idéale,
4. le problème probabiliste du calcul des vitesses des six articulations impliquant la vitesse de parcours dans l'espace de travail.

Nous décrivons chacun des étapes dans les paragraphes suivants.

#### 4.1.2 Trajectoire idéale

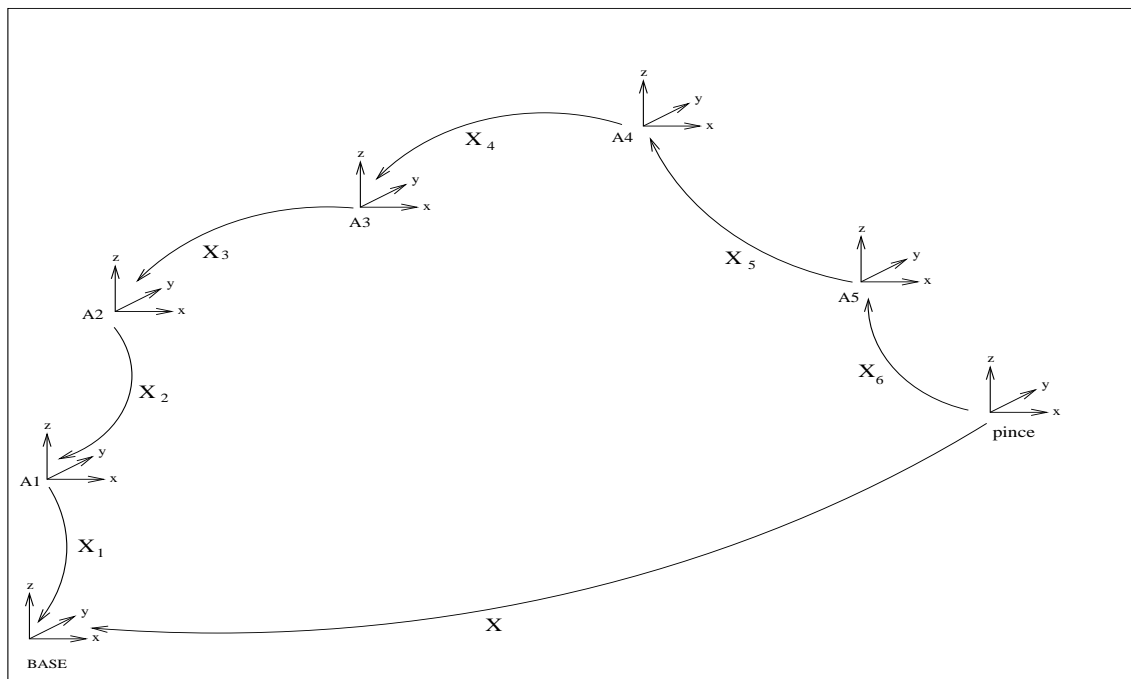


FIG. 4.1 – Transformation de la position de la pince

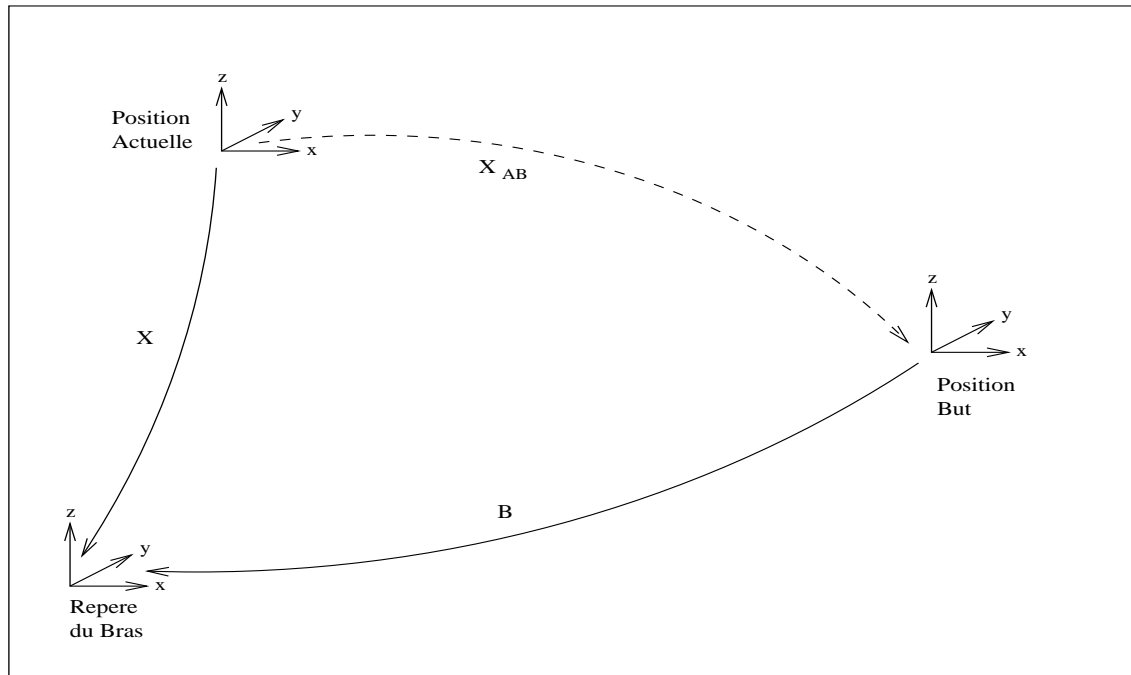


FIG. 4.2 – Trajectoire idéale

Le calcul de la trajectoire idéale dépend de la position actuelle du bras et de la position but. La position actuelle est calculée à partir de la géométrie du bras (figure 4.1 correspondant au modèle géométrique du bras, voir section 3.2.1) :

$$T_X = T_{X_1} * T_{X_2} * T_{X_3} * T_{X_4} * T_{X_5} * T_{X_6} \quad (4.1)$$

où :

- $T_{X_i}$  représente la matrice homogène de transformation correspondant au vecteur de pose  $X_i$  de la  $i^{\text{ème}}$  articulation du bras, avec  $i = 1 \dots 6$ ,
- $X$  est le vecteur de pose représentant la position actuelle de la pince dans le repère du bras.

La position  $B$  du but dépend de l'application, elle peut être fournie par la vision stéréoscopique ou par n'importe quelle autre application. En conséquence, nous considérons qu'elle est déjà connue et qu'elle peut être actualisée tous les dix millièmes de seconde.

La figure 4.2 représente le graphe cinématique du problème de mouvement du bras. L'expression :

$$X_{AB} = B - X \quad (4.2)$$

représente la trajectoire idéale de la position actuelle à la position du but.

Le vecteur de pose  $X_{AB} = (\rho, \omega)^T$  est composé de :



1. un vecteur de translation

$$\rho = (x, y, z)^T$$

où  $x$ ,  $y$  et  $z$  représentent les déplacements désirés sur les axes cartésiens  $X$ ,  $Y$  et  $Z$  respectivement,

2. un vecteur de rotation

$$\omega = (r_x, r_y, r_z)^T$$

où,  $r_x$ ,  $r_y$  et  $r_z$  représentent l'axe de rotation et  $\|\omega\|$  l'angle de rotation.

Les deux vecteurs  $\rho$  et  $\omega$  représentent la translation et la rotation nécessaires pour aller de la position actuelle  $X$  au but  $B$ . C'est-à-dire la trajectoire idéale.

### 4.1.3 Temps de parcours

Dans cette section, nous calculons le temps de parcours de la trajectoire idéale en fonction de l'accélération et vitesse maximale du bras.

La génération de mouvement dans l'espace opérationnel a l'avantage de contrôler la géométrie de la trajectoire dans un espace cartésien. Par contre, les limites physiques et cinématiques du bras varient selon *la configuration du bras* (position articulaire). Étant donnée la variabilité de ces limites, nous utilisons simplement les limites fournies dans la documentation du bras.

Les limites pertinentes du bras pour le mouvement sont :

- $\dot{\Theta}_M$  - représentant la vitesse angulaire maximale,
- $\ddot{\Theta}$  - représentant l'accélération angulaire,
- $\dot{X}_M$  - représentant la vitesse cartésienne maximale,
- $\ddot{X}$  - représentant l'accélération cartésienne.

Pour un déplacement à vitesse maximale, nous calculons d'abord la distance d'accélération nécessaire pour atteindre cette vitesse maximale et la distance pour s'arrêter :

$$d = v_0 t + \frac{1}{2} a t^2 \quad (4.3)$$

$$v = a t \quad (4.4)$$

De l'expression (4.4),

$$t = \frac{v}{a}$$

Si on substitue  $t$  en (4.3) en considérant  $v_0 = 0$ ; nous avons l'expression de la distance en fonction de l'accélération et de la vitesse [KD99].

$$d = \frac{v^2}{2a} \quad (4.5)$$

avec cette expression, nous pouvons calculer la distance  $d$  pour atteindre la vitesse  $v$  en partant d'une vitesse initiale nulle.

En considérant que l'accélération et la décélération sont égales, nous pouvons considérer que la distance pour atteindre la vitesse  $v$  est égale à la distance pour arrêter le bras quand il bouge à la vitesse  $v$ .

Pour calculer le temps de parcours d'une trajectoire, nous avons besoin de connaître la vitesse de mouvement dans l'espace opérationnel. Pour la calculer, nous avons :

- la distance cartésienne de la trajectoire idéale, dénotée  $\|\rho\|$ ,
- la distance angulaire du parcours, dénotée  $\|\omega\|$ ,
- la vitesse angulaire maximale  $\dot{\Theta}_M$ ,
- la vitesse cartésienne maximale  $\dot{X}_M$ ,
- l'accélération articulaire  $\ddot{\Theta}$  et,
- l'accélération cartésienne  $\ddot{X}$ .

Pour calculer la vitesse maximale de translation  $\|\dot{\rho}_m\|$  pour ce mouvement on utilise l'expression (4.5),

$$\|\dot{\rho}_m\| = \begin{cases} \dot{X}_M & \text{si } \frac{\dot{X}_M^2}{\ddot{X}} < \|\rho\| \\ \sqrt{\|\rho\|\ddot{X}} & \text{sinon} \end{cases} \quad (4.6)$$

où,

- $\frac{\dot{X}_M^2}{\ddot{X}}$  représente la distance nécessaire pour atteindre la vitesse maximale plus la distance nécessaire pour arrêter le bras,
- $\frac{\dot{X}_M^2}{\ddot{X}} < \|\rho\|$  implique que le bras a assez d'espace pour bouger à sa vitesse de translation maximale  $\dot{X}_M$ ,
- $\frac{\dot{X}_M^2}{\ddot{X}} > \|\rho\|$  implique que le bras ne peut pas dépasser la vitesse de translation  $\sqrt{\|\rho\|\ddot{X}}$ .

De façon similaire pour la grandeur de la vitesse angulaire maximale  $\|\dot{\omega}_m\|$ ,

$$\|\dot{\omega}_m\| = \begin{cases} \dot{\Theta}_M & \text{si } \frac{\dot{\Theta}_M^2}{\ddot{\Theta}} < \|\omega\| \\ \sqrt{\|\omega\|\ddot{\Theta}} & \text{sinon} \end{cases} \quad (4.7)$$

Nous pouvons maintenant calculer le temps de parcours  $t_\rho$  pour faire la translation et le temps de parcours  $t_\omega$  pour faire la rotation,

$$t_\rho = \frac{\|\rho\|}{\|\dot{\rho}_m\|}$$

$$t_\omega = \frac{\|\omega\|}{\|\dot{\omega}_m\|}$$

Nous synchronisons les mouvements en translation et en rotation en utilisant le temps le plus grand pour gérer le déplacement,

$$t = \max\{t_\rho, t_\omega\} \quad (4.8)$$

#### 4.1.4 Vitesse idéale de parcours

Nous venons de calculer le temps  $t$  de parcours en fonction des vitesses de translation et rotation maximales cependant, la vitesse de parcours n'a pas encore été définie car le temps pour faire la rotation est en général différent du temps pour faire la translation.

Dans les paragraphes précédents, nous avons déterminé :

1. le temps  $t$  de parcours,
2. la trajectoire cartésienne  $\rho$ ,
3. le trajectoire angulaire  $\omega$ .

Le vecteur de vitesse de translation idéale  $\dot{\rho}^*$  et le vecteur de vitesse de rotation idéale  $\dot{\omega}^*$  pour parcourir la trajectoire  $X_{AB}$  en respectant les limites du bras sont calculés par :

$$\dot{\rho}^* = \frac{1}{t}\rho \quad (4.9)$$

$$\dot{\omega}^* = \frac{1}{t}\omega \quad (4.10)$$

Nous indiquons la vitesse idéale dans ce parcours,

$$\begin{aligned} \dot{X}^* &= (\dot{\rho}^*, \dot{\omega}^*)^T \\ &= (\dot{x}^*, \dot{y}^*, \dot{z}^*, \dot{r}_x^*, \dot{r}_y^*, \dot{r}_z^*)^T \end{aligned} \quad (4.11)$$

où  $\dot{X}^*$  est la structure de données représentant la vitesse de parcours idéale dans l'espace opérationnel.

Dans le paragraphe suivante, nous définissons le problème probabiliste pour le mouvement du bras en utilisant : la vitesse idéale de parcours, la configuration du bras (position articulaire et vitesse articulaire), son accélération et sa vitesse maximale.

#### 4.1.5 Variables

Nous abordons le problème du calcul de la vitesse des articulations avec la programmation bayésienne. Nous considérons les variables suivantes :

- $\Theta$  - position articulaire actuelle. Dans notre cas, c'est un vecteur à six composantes contenant les valeurs des six articulations du bras nécessaires pour calculer la position de la pince (voir section 3.2.1),

- $\dot{\Theta}$  - vitesse articulaire, il contient les six vitesses des articulations au temps  $t$ ,
- $\dot{\Theta}_{-1}$  - vitesse articulaire au temps  $t - 1$ . Elle délimite la vitesse  $\dot{\Theta}$  en fonction de l'accélération et de la vitesse maximale,
- $X$  - représente la position cartésienne du bras (position de la pince sur le repère du bras),
- $B$  - représente la position du but dans l'espace cartésien,
- $\dot{X}$  - représente la vitesse du bras (vitesse de déplacement de la pince) dans l'espace cartésien,
- $\dot{X}^*$  - vitesse cartésienne idéale (pour le déplacement de la pince),
- $E$  - c'est la différence entre la vitesse  $\dot{X}$  du bras et la vitesse idéale  $\dot{X}^*$ . Elle représente l'erreur entre la vitesse idéale et la vitesse commandée.

La distribution de probabilité conjointe pour le mouvement en vitesse du bras s'écrit :

$$p(\Theta \dot{\Theta}_{-1} X B \dot{\Theta} \dot{X} \dot{X}^* E),$$

elle contient toutes les variables impliquées dans le problème. Puisque l'accélération et la vitesse maximale du bras sont considérées fixes, nous ne les écrivons pas dans la distribution de probabilité conjointe.

#### 4.1.6 Décomposition pour le mouvement

Le choix de la décomposition traduit notre connaissance du problème. Les produits élémentaires constituant la décomposition sont choisis en fonction des relations existantes entre l'espace articulaire et l'espace opérationnel (voir section 3.2.1).

La décomposition choisie est la suivante,

$$\begin{aligned} & p(\Theta \dot{\Theta}_{-1} X B \dot{\Theta} \dot{X} \dot{X}^* E) \\ & = \\ & p(\Theta) p(B) p(\dot{\Theta}_{-1}) p(\dot{\Theta}|\dot{\Theta}_{-1}) p(X|\Theta) p(\dot{X}|\Theta\dot{\Theta}) p(\dot{X}^*|XB) p(E|\dot{X}\dot{X}^*) \end{aligned} \quad (4.12)$$

Les hypothèses d'indépendances conditionnelles non évidentes sont les suivantes :

1. sachant la position articulaire  $\Theta$ , la position cartésienne  $X$  peut être déterminée de manière exacte. Il faut remarquer que l'inverse n'est pas vraie,
2. sachant la position articulaire  $\Theta$  et la vitesse articulaire  $\dot{\Theta}$ , la vitesse cartésienne  $\dot{X}$  peut être déterminée de manière exacte grâce à la jacobienne,
3. nous avons choisi de définir la vitesse cartésienne idéale  $\dot{X}^*$  uniquement en fonction du but  $B$  et de la position actuelle  $X$ ,
4. l'intervalle de temps  $\Delta_t$  entre  $t$  et  $t - 1$  est considéré constant.

#### 4.1.7 Formes paramétriques pour le mouvement

Les descriptions des formes paramétriques associées aux facteurs de la décomposition définissent le cadre de connaissance sur le mouvement du bras.

Les formes paramétriques choisies sont :

- $p(\Theta) = Uniforme$  - Indique le fait qu'il n'y a pas d'*a priori* sur les valeurs des articulations,
- $p(B) = Uniforme$  - Indique que le but peut être n'importe où dans l'espace opérationnel,
- $p(\dot{\Theta}_{-1}) = Uniforme$  - La dernière vitesse articulaire du bras peut être n'importe laquelle dans la plage de vitesses de ses articulations,
- $p(\dot{\Theta}|\dot{\Theta}_{-1}) = Uniforme$  - La distribution uniforme définit la plage de valeurs pour les vitesses des articulations ainsi,

$$[\max\{-\dot{\Theta}_{max}, \dot{\Theta}_{-1} - \Delta_t \dot{\Theta}\}, \min\{\dot{\Theta}_{max}, \dot{\Theta}_{-1} + \Delta_t \dot{\Theta}\}] \quad (4.13)$$

- $p(X|\Theta) = \delta_{g_1(\Theta)}(X)$ <sup>1</sup> - La position cartésienne  $X$  est fonction de la position articulaire  $\Theta$ . La fonction  $g_1(\Theta)$  représente la transformation géométrique directe (eqn. 4.1). Nous considérons ce calcul sans incertitude.
- $p(\dot{X}|\Theta\dot{\Theta}) = \delta_{g_2(\Theta,\dot{\Theta})}(\dot{X})$  - La vitesse cartésienne  $\dot{X}$  est fonction de la position articulaire  $\Theta$  et de la vitesse articulaire  $\dot{\Theta}$ . La fonction  $g_2(\Theta, \dot{\Theta})$  est définie par l'expression :

$$\dot{X} = J_{\Theta}\dot{\Theta},$$

où  $J$  représente la jacobienne à la position articulaire  $\Theta$ ,

- $p(\dot{X}^*|XB) = \delta_{g_3(X,B)}(\dot{X}^*)$  - La vitesse idéale  $\dot{X}^*$  est fonction de la position cartésienne actuelle  $X$  et de la position du but  $B$ . L'expression  $g_3(X, B)$  est calculée par (4.11).
- $p(E|\dot{X}\dot{X}^*) = Normale(\|\dot{X} - \dot{X}^*\|, \Sigma)$  - Indique que la différence entre la vitesse commandée et la vitesse idéale suit une distribution normale. La moyenne est la différence entre les vitesses (l'erreur constatée) et  $\Sigma$  représente la matrice de covariance, définie par :

$$\Sigma = \begin{pmatrix} \sigma^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2 \end{pmatrix}$$

En donnant cette matrice nous supposons les composants de  $E$  comme indépendants.

Le cadre de connaissance que nous venons d'établir est nommé  $c_{mov}$ . Nous allons l'utiliser pour inférer la vitesse des articulations pour amener le bras de sa position actuelle au but.

#### 4.1.8 Inférence probabiliste

La question probabiliste pour inférer les vitesses des articulations en utilisant le cadre de connaissance  $c_{mov}$  est :

---

<sup>1</sup>  $\delta$  représente un dirac défini :  $\delta_{g_1(\Theta)}(X) = \begin{cases} 1 & \text{si } X = g_1(\Theta), \\ 0 & \text{sinon.} \end{cases}$

Quelle est la probabilité de la vitesse des articulations sachant la position articulaire actuelle, la position du but, la dernière vitesse articulaire et que la vitesse cartésienne commandée doit être égale à la vitesse idéale ?

Le problème défini par la question s'écrit,

$$p(\dot{\Theta} \mid \Theta B \dot{\Theta}_{-1} [E = 0] c_{mov})$$

En utilisant l'expression de la décomposition (eqn. 4.12) et le théorème de Bayes,

$$p(\dot{\Theta} \mid \Theta B \dot{\Theta}_{-1} [E = 0] c_{mov}) \int_{\dot{X}} \int_{\dot{X}^*} p(\Theta) p(B) p(\dot{\Theta}_{-1}) p(\dot{\Theta} \mid \dot{\Theta}_{-1}) p(X \mid \Theta) p(\dot{X} \mid \Theta \dot{\Theta}) p(\dot{X}^* \mid XB) p([E = 0] \mid \dot{X} \dot{X}^*) d_{\dot{X}^*} d_{\dot{X}} d_X \quad (4.14)$$

En considérant que les termes  $p(X \mid \Theta)$ ,  $p(\dot{X} \mid \Theta \dot{\Theta})$ ,  $p(\dot{X}^* \mid XB)$  sont des distributions de type dirac, nous pouvons simplifier cette expression pour obtenir :

$$p(\dot{\Theta} \mid \Theta B \dot{\Theta}_{-1} [E = 0] c_{mov}) \int_{\dot{X}} \int_{\dot{X}^*} p(\Theta) p(B) p(\dot{\Theta}_{-1}) p(\dot{\Theta} \mid \dot{\Theta}_{-1}) p([E = 0] \mid g_2(\Theta, \dot{\Theta}) g_3(g_1(\Theta), B)) p(\dot{\Theta} \mid \dot{\Theta}_{-1}) p([E = 0] \mid g_2(\Theta, \dot{\Theta}) g_3(g_1(\Theta), B)) \quad (4.15)$$

La dernière expression définit une distribution de probabilité représentant l'espace solution pour les vitesses des articulations du bras.

#### 4.1.9 Solution pour le mouvement du bras

La distribution étant connue, il nous faut maintenant déterminer une valeur unique de  $\dot{\Theta}$ . Pour cela, Nous allons choisir celle qui maximise  $p(\dot{\Theta} \mid \Theta B \dot{\Theta}_{-1} [E = 0] c_{mov})$  ;

$$\begin{aligned} \dot{\Theta}_{solution} &= \max_{\dot{\Theta}} p(\dot{\Theta} \mid \Theta B \dot{\Theta}_{-1} [E = 0] c_{mov}) \\ &= \max_{\dot{\Theta}} p(\dot{\Theta} \mid \dot{\Theta}_{-1}) p([E = 0] \mid g_2(\Theta, \dot{\Theta}) g_3(g_1(\Theta), B)) \end{aligned} \quad (4.16)$$

L'expression de  $p(\dot{\Theta} \mid \dot{\Theta}_{-1})$  donne un fort *a priori* sur la vitesse des articulations sachant la vitesse à  $t - 1$ . En effet celle-ci est bornée par les possibilités d'accélération du bras. Pour trouver une solution au problème précédent il nous suffit de considérer les valeurs de  $\dot{\Theta}$  se trouvant dans cet intervalle.

Une solution acceptable est de discrétiser cet intervalle et de conduire une maximisation exhaustive de l'expression (4.16).

Nous avons testé cette solution avec plusieurs niveaux de discrétisation : 3, 5 ou 7 vitesses par articulation. Bien entendu, le mouvement est plus souple quand la discrétisation est plus fine. Par contre, le temps de réponse devient exponentiellement plus grand. Nous avons finalement utiliser des pas de discrétisation variables par articulation. Ces pas sont fonction de la taille de la plage de vitesses des articulations (eqn. 4.13).

Il existe cependant une alternative plus classique à la résolution de ce problème de minimisation : le contrôle en vitesse par inversion de la jacobienne.

$$\begin{aligned} \max_{\dot{\Theta} \in [\max\{-\dot{\Theta}_{max}, \dot{\Theta}_{-1} - \Delta_t \dot{\Theta}\}, \min\{\dot{\Theta}_{max}, \dot{\Theta}_{-1} + \Delta_t \dot{\Theta}\}]} p(\dot{\Theta} \mid \Theta B \dot{\Theta}_{-1} [E = 0] c_{mov}) \\ = \\ \max_{\dot{\Theta}} \prod_{i=0}^{n-1} \text{Normale}(\|\dot{x}_i - \dot{x}_i^*\|, \sigma) \end{aligned} \quad (4.17)$$

avec,

- $\dot{x}_i = g_2(\Theta, \dot{\Theta})$  représente la vitesse dans l'espace cartésien.
- $\dot{x}_i^* = g_3(g_1(\Theta), B)$  représentant la vitesse idéale.
- $n$  est égal à 6.

Nous écrivons la distribution normale comme le produit de distributions sur les paramètres car, la matrice de covariance nous indique que les paramètres sont indépendants les uns des autres (les éléments en dehors de la diagonale principale sont nulles).

Puisque la distribution de probabilités est strictement positive, nous pouvons penser à maximiser son logarithme,

$$\begin{aligned} \max_{\dot{\Theta}} p(\dot{\Theta} \mid \Theta B \dot{\Theta}_{-1} [E = 0] c_{mov}) &= \max_{\dot{\Theta}} \ln \prod_{i=0}^{n-1} \text{Normale}(\|\dot{x}_i - \dot{x}_i^*\|, \sigma) \\ &= \max_{\dot{\Theta}} \ln \prod_{i=0}^{n-1} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2} \left( \frac{\dot{x}_i^* - \dot{x}_i}{\sigma} \right)^2} \\ &= \max_{\dot{\Theta}} \sum_{i=0}^{n-1} \left[ \ln \left( \frac{1}{\sqrt{2\pi\sigma}} \right) - \frac{1}{2} \left( \frac{\dot{x}_i^* - \dot{x}_i}{\sigma} \right)^2 \right] \\ &= \min_{\dot{\Theta}} \sum_{i=0}^{n-1} \left( \frac{\dot{x}_i^* - \dot{x}_i}{\sigma} \right)^2 \\ &= \min_{\dot{\Theta}} \sum_{i=0}^{n-1} (\dot{x}_i^* - \dot{x}_i)^2 \end{aligned} \quad (4.18)$$

Alors, la solution cherchée est celle minimisant la distance entre le vecteur de vitesse idéale

$\dot{X}^*$  et le vecteur de vitesse obtenu par  $\dot{X} = J_{\Theta}\dot{\Theta}$ .

Une façon de trouver les vitesses est de poser le système,

$$\dot{X}^* = J_{\Theta}\dot{\Theta}$$

et de le résoudre en calculant,

$$\dot{\Theta} = J_{\Theta}^{-1}\dot{X}^* \quad (4.19)$$

où,  $J_{\Theta}^{-1}$  représente la pseudo-inverse du jacobien. Nous l'obtenons avec la méthode de Greville. Cela correspond à chercher  $\dot{\Theta}$  par la méthode de moindre carré.

#### 4.1.10 Conclusion

Tout en ayant constaté la faisabilité de la méthode par discrétisation, nous avons finalement retenu la méthode classique de contrôle en vitesse en lui donnant une interprétation probabiliste.

Les résultats obtenus sont très satisfaisants car le mouvement du bras est souple et l'inférence est fait dans le temps prévu. Le temps de réponse est de moins de 10 millièmes de seconde. Le bras ralentit lorsqu'il s'approche du but, quand le bras a assez de distance pour se déplacer à sa vitesse maximale (toujours limitée par des raisons de sûreté), il augmente sa vitesse.

Nous avons simulé le changement de la position but et nous avons constaté que le bras modifie sa trajectoire pour aller vers le nouveau but. Ce changement de trajectoire est fait de façon souple.

## 4.2 Localisation d'objets avec la vision stéréoscopique

Nous avons besoin de localiser des objets pour pouvoir les saisir. Pour cela, nous allons utiliser le système de vision stéréoscopique.

Dans l'annexe [B.2.2](#), nous montrons comment trouver la position d'un point dans l'espace à partir de ses projections dans les caméras du système de vision en utilisant l'inférence probabiliste. Dans cette section, nous allons repérer la position d'un objet solide à partir de cette capacité élémentaire. La méthode utilisée est aussi probabiliste. Elle reprend les principes du raisonnement géométrique proposés par [\[Mek99\]](#) dans le cadre probabiliste. Elle s'appuie sur l'utilisation d'un modèle géométrique spécifique des objets où chaque objet est représenté par un ensemble de points caractéristiques.



### 4.2.1 Description du problème

Le problème consiste à trouver la position d'un objet au moyen de la description géométrique de l'objet et des positions des points de référence fournis par la vision stéréoscopique (voir figure 4.3). La description géométrique d'un objet est faite en décrivant les coordonnées de chacun des points de référence dans le repère propre à l'objet (voir section 3.2.2). Ces points sont directement matérialisés sur l'objet. Dans un cadre plus orienté vision ces points seraient obtenus à partir de pré-traitement de l'image.

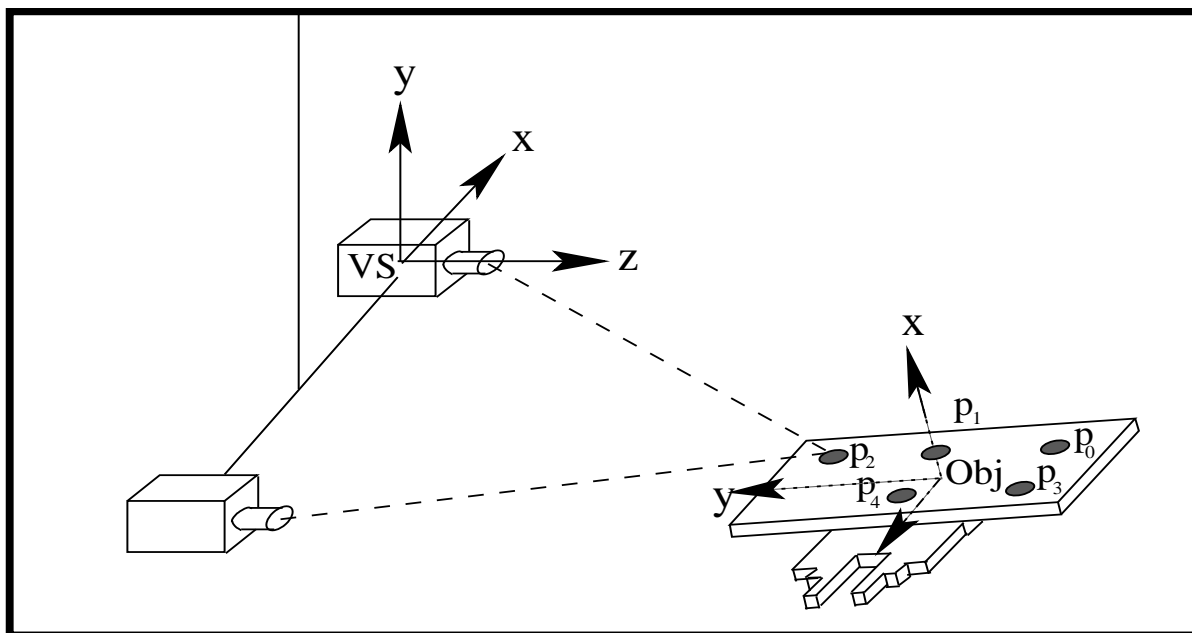


FIG. 4.3 – Points de référence de la pince vus par la paire stéréoscopique.

Le point de référence  $P_i$ , avec  $i = 0 \dots n - 1$ , dans le repère  $Obj$  de l'objet est représenté par un vecteur de pose  $Z_i$ .

Les coordonnées des points  $P_i$ , fournis par la vision stéréoscopique (annexe B.2), sont exprimées dans le repère  $VS$  du système de vision par des vecteurs de pose  $M_i$ .

Dans le cadre de cette étude, nous ne considérons pas le problème de la mise en correspondance des points, nous supposons que les points obtenus par le système de vision correspondent un à un aux points de l'objet. En pratique, cela se fait en demandant à l'opérateur d'indiquer un par un les points à localiser.

De la description précédente, les données du problème sont les suivantes :

1. le repère  $Obj$  de l'objet.
2. la description géométrique de l'objet (les points caractéristiques  $Z_i$ ).

3. le repère  $VS$  de la vision stéréoscopique.

4. les positions des points caractéristiques fournies par la vision stéréoscopique (les  $M_i$ ).

Le problème est donc de trouver la position ( $PosObj$ ) d'un objet en fonction de sa description géométrique (les  $Z_i$ ) et de l'information sur les positions des points de référence fournie par le système de vision stéréoscopique (les  $M_i$ ).

### Exemple

Les 5 points de référence sur la pince ont pour description :

$$Z_i = (Z_i.x, Z_i.y, Z_i.z, -, -, -)$$

avec  $i = 0, 1, 2, 3, 4$ .

Les paramètres  $Z_i.x$ ,  $Z_i.y$  et  $Z_i.z$  représentent les coordonnées  $x$ ,  $y$  et  $z$  du point  $P_i$  sur le repère de l'objet ( $Obj$ ). Les paramètres correspondants à la rotation dans le vecteur de pose n'ont pas besoin d'être définis dans le cas de points : ils sont représentés par le symbole  $-$ .

Les vecteurs de pose obtenus par le système de vision sont :

$$M_i = (M_i.x, M_i.y, M_i.z, -, -, -)$$

Les expressions  $M_i.x$ ,  $M_i.y$  et  $M_i.z$  représentent les coordonnées  $x$ ,  $y$  et  $z$ , du point de référence  $P_i$ , fournies par la vision stéréoscopique. En suivant nos hypothèses, ces points correspondent un à un aux points de l'objet. C'est-à-dire que  $Z_i$  sur l'objet correspond à  $M_i$  sur le système de vision.

Pour placer le problème dans le cadre probabiliste nous allons nous aider d'une structure de donnée particulière : le *graphe cinématique*. Car il a l'avantage de représenter facilement les relations géométriques.

## 4.2.2 Graphe cinématique du problème

Les données décrites dans le dernier paragraphe nous permettent de construire un graphe orienté représentant le problème de la localisation (figure 4.4). Ce graphe est appelé *graphe cinématique* [Mek99]. Il a été décrit dans la section 2.3.1.

Le graphe cinématique est représenté par :

1. l'ensemble de sommets,

$$S = \{VS, Obj, P_0, \dots, P_{n-1}\}.$$

2. l'ensemble d'arcs,

$$A = \{M_0, \dots, M_{n-1}, PosObj, Z_0, \dots, Z_{n-1}\}.$$

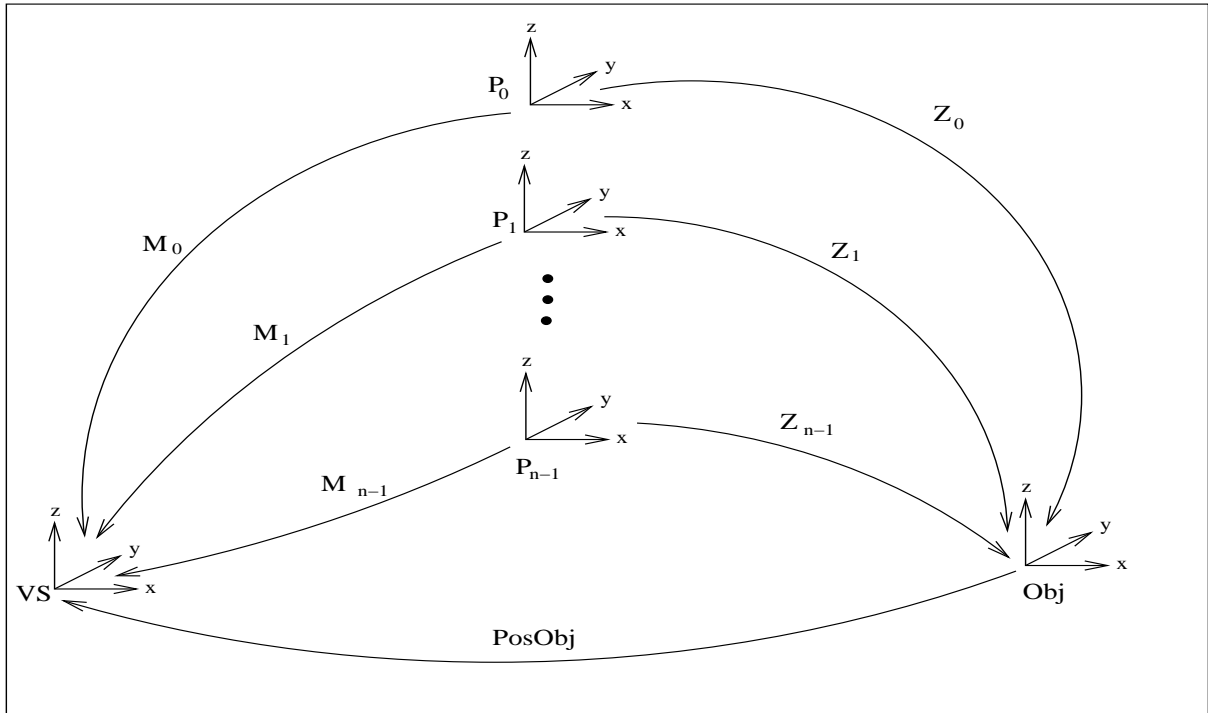


FIG. 4.4 – Graphe pour la localisation d'un objet.

Les sommets représentent les repères géométriques :

- $P_i$  représente le repère propre à chaque point de référence sur l'objet, avec  $i = 0 \dots n - 1$ ,
- $VS$  est le repère du système de vision stéréoscopique et,
- $Obj$  est le repère de l'objet.

Les arcs représentent les transformations géométriques entre les repères. Ils sont représentés avec des vecteurs de pose ;

- $Z_i$  représente la position du point caractéristique  $P_i$  dans le repère  $Obj$  de l'objet,
- $M_i$  représente la position du point caractéristique  $P_i$  dans le repère  $VS$  de la vision stéréoscopique et,
- $PosObj$  représente la position de l'objet  $Obj$  dans le repère de la vision stéréoscopique  $VS$ .

Le problème de la localisation stéréoscopique est résolu en trouvant cette position.

Dans la figure 4.4, on note la présence de plusieurs cycles représentant les contraintes globales du problème de localisation. Chaque cycle indique une contrainte géométrique correspondant à la fermeture d'une boucle cinématique. Ces contraintes expriment simplement que dans un problème géométrique la position d'un repère est unique.

Puisqu'il existe un grand nombre de cycles, il faut déterminer un ensemble minimal de cycles permettant de couvrir complètement le graphe. Si les contraintes sont respectées sur cet ensemble alors, elles seront respectées sur tous les cycles et par la même toutes les contraintes seront

respectées.

On sait que le nombre minimal de cycles permettant de couvrir complètement un graphe est le nombre d'arcs  $a$  moins le nombre de sommets  $s$  plus 1 [GM90]. Dans ce cas :

- le nombre d'arcs est  $a = 2n + 1$  car il y a deux arcs pour chaque point caractéristique  $P_i$  ; l'arc  $M_i$  sur le repère du système de vision et l'arc  $Z_i$  sur le repère de l'objet plus l'arc  $PosObj$  entre le repère de l'objet et le repère du système de vision,
- le nombre de sommets  $s = n + 2$  représente les  $n$  points de référence plus le repère de la vision stéréoscopique et le repère de l'objet.

Le nombre minimal de cycles dans le graphe est,

$$\begin{aligned} \text{Nombre\_de\_cycles} &= a - s + 1 \\ &= 2n + 1 - (n + 2) + 1 \\ &= n, \end{aligned}$$

c'est-à-dire que pour le problème de la localisation stéréoscopique, il y a un cycle pour chaque point de référence.

Nous avons implanté une version modifiée de l'algorithme de Dijkstra [Dij59] pour trouver les cycles de longueur minimale contenant le plus grand nombre de vecteurs de pose avec des paramètres inconnus.

Nous pouvons maintenant utiliser notre méthodologie de description probabiliste du problème.

### 4.2.3 Variables pour la localisation stéréoscopique

Les variables du problème sont :

1. des variables composant les vecteurs de pose.

$$\{M_0, \dots, M_{n-1}, PosObj, Z_0, \dots, Z_{n-1}\},$$

elles sont labélisées par ( $F$ ), ( $L$ ) ou ( $X$ ) selon qu'elles caractérisent des variables *fixes*, *libres* ou *inconnues* du problème :

- *fixe* ou ( $F$ ) quand sa valeur est fixe. Dans ce cas il ne s'agit pas d'une variable probabiliste, elle n'apparaîtra pas explicitement dans la modélisation.
- *libre* ou ( $L$ ) quand sa valeur est définie au moyen d'une distribution de probabilité représentant la précision du dispositif associé à cet élément,
- *inconnue* ou ( $X$ ) quand nous voulons connaître la valeur de cet élément du vecteur de pose. Cette valeur peut être estimée par une distribution de probabilité représentant un *a priori* sur sa valeur.

2. des variables booléennes de fermeture correspondant chacune à une contrainte.

$$\{K_i\}_{i=0}^{n-1}$$

## Exemple

Pour le problème de la localisation, nous pouvons construire l'état de connaissances suivant :

1. les variables composant les vecteurs de pose. Elles sont décrites dans le tableaux (4.1). Le type de paramètre  $X$  et  $L$  dans la troisième colonne indique qu'il s'agit d'une variable probabiliste. Le domaine de leurs valeurs est  $\mathbb{R}$ .

Les variables *libres* sont les paramètres des arcs  $M_i$ . Ainsi, les variables libres sont :

$$\{M_{0.x}, M_{0.y}, M_{0.z}, \dots, M_{n-1.x}, M_{n-1.y}, M_{n-1.z}\}$$

et les variables *inconnues* sont les paramètres de l'arc  $PosObj$ . Alors, elles sont :

$$\{PosObj.x, PosObj.y, PosObj.z, PosObj.r_x, PosObj.r_y, PosObj.r_z\}$$

2. les  $n$  variables booléennes de fermeture,

$$\{K\}_{i=0}^{n-1}$$

Arcs	Distribution de ( $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z$ )	Type de paramètre
$M_0$	$M_{0.x} = Normale(\mu_{M_{0.x}}, \sigma_{VS.x})$ $M_{0.y} = Normale(\mu_{M_{0.y}}, \sigma_{VS.y})$ $M_{0.z} = Normale(\mu_{M_{0.z}}, \sigma_{VS.z})$	$L$ $L$ $L$
$\vdots$	$\vdots$	$\vdots$
$M_{n-1}$	$M_{n-1.x} = Normale(\mu_{M_{n-1.x}}, \sigma_{VS.x})$ $M_{n-1.y} = Normale(\mu_{M_{n-1.y}}, \sigma_{VS.y})$ $M_{n-1.z} = Normale(\mu_{M_{n-1.z}}, \sigma_{VS.z})$	$L$ $L$ $L$
$PosObj$	$PosObj.x = Uniforme[a_1, b_1]$ $PosObj.y = Uniforme[a_2, b_2]$ $PosObj.z = Uniforme[a_3, b_3]$ $PosObj.r_x = Uniforme[a_4, b_4]$ $PosObj.r_y = Uniforme[a_5, b_5]$ $PosObj.r_z = Uniforme[a_6, b_6]$	$X$ $X$ $X$ $X$ $X$ $X$
$Z_0$	$(Z_{0.x}, Z_{0.y}, Z_{0.z}, -, -, -)$	$(F, F, F)$
$\vdots$	$\vdots$	$\vdots$
$Z_{n-1}$	$(Z_{n-1.x}, Z_{n-1.y}, Z_{n-1.z}, -, -, -)$	$(F, F, F)$

TAB. 4.1 – Ensemble d'arcs du graphe pour la localisation.

Le cadre de connaissance établi par les variables permet de définir la distribution de probabilité conjointe pour la localisation avec la vision stéréoscopique :

$$p(M_0 \dots M_{n-1} PosObj K_0 \dots K_{n-1}). \quad (4.20)$$

Un choix de décomposition est maintenant nécessaire pour réduire la complexité de calcul en exprimant notre connaissance sur le problème.

#### 4.2.4 Décomposition pour la localisation

Pour calculer explicitement cette distribution conjointe, nous faisons le choix de la décomposition suivante :

$$p \left( \begin{array}{c} M_0 \dots M_{n-1} \\ PosObj \end{array} \middle| K_0 \dots K_{n-1} \right) = p(M_0 \dots M_{n-1}) p(PosObj) \prod_{i=0}^{n-1} p(K_i | PosObj M_i) \quad (4.21)$$

avec :

$$p(PosObj) = p(PosObj.x) p(PosObj.y) p(PosObj.z) p(PosObj.r_x) p(PosObj.r_y) p(PosObj.r_z)$$

car, nous avons établi l'hypothèse de mutuelle indépendance entre les inconnues.

$$p(M_0 \dots M_{n-1}) = \prod_{i=0}^{n-1} p(M_i)$$

où  $p(M_i) = p(M_i.x) p(M_i.y) p(M_i.z)$  car les variables libres sont mutuellement indépendantes.

Dans cette décomposition nous faisons l'hypothèse :

$$P(PosObj) = P(PosObj | M_0 \dots M_{n-1})$$

Cela veut dire que nous considérons que la position de l'objet est indépendante des mesures dès lors qu'on ignore les contraintes de fermeture. Cette hypothèse signifie simplement qu'on ne peut interpréter des mesures qu'en présence de modèle.

Par ailleurs, nous considérons que toutes les composantes de tous les vecteurs sont indépendants entre elles.

#### 4.2.5 Formes paramétriques

##### Variables liées à la vision stéréoscopique

Dans le tableau (4.1), les vecteurs de pose nommés  $M_i$  représentent la position des points de référence dans le repère de la vision :  $VS$ . La valeur de chaque coordonnée  $x, y$  et  $z$  de chaque point est connue avec une certaine précision autour de la valeur donnée par la mesure. Cette hypothèse justifie le choix d'une distribution normale centrée sur la valeur de la mesure.

Par exemple, la variable  $M_{0.x}$  est décrite par une distribution de probabilité normale avec :

- moyenne  $\mu_{M_0.x}$  qui est la valeur fournie par la vision stéréoscopique,
- écart-type  $\sigma_{V_S.x}$  associé à tous les composants  $x$  fournis par la vision stéréoscopique.

### Variables du vecteur de pose $PosObj$

Ce vecteur contient les variables inconnues du problème, comme indiqué dans le tableau (4.1) :

$$PosObj.x, PosObj.y, PosObj.z, PosObj.r_x, PosObj.r_y, PosObj.r_z,$$

Un *a priori* uniforme est donné pour chacune des composantes. Cela indique que la seule connaissance que nous pouvons donner au système est la plage de variation de chacune de ces variables. Cette plage peut se voir comme des contraintes locales. Par exemple, si l'objet est repéré par la vision stéréoscopique, il doit être forcément en face d'elle, c'est-à-dire que la composante  $PosObj.z$  de la position de l'objet par rapport à la paire stéréoscopique ne peut pas être négative. Cela permet de réduire l'espace de recherche.

### Contrainte de fermeture

Elles représentent les contraintes géométriques globales du problème et correspondent aux cycles dans le graphe cinématique (fig. 4.4). Ces cycles contiennent toujours le vecteur de pose  $PosObj$  contenant les inconnues.

La distribution  $p(K_i | PosObj M_i)$  est un dirac donnée par :

$$p(K_i | PosObj M_i) = \begin{cases} 1 & \text{si } I = T_{PosObj} * T_{Z_i} * T_{M_i}^{-1}, \\ 0 & \text{sinon,} \end{cases} \quad (4.22)$$

où :

- $T$  est la transformation homogène liée a un arc du graphe cinématique.  $T^{-1}$ , correspond à un arc parcouru en sens inverse du sens indiqué dans le graphe cinématique (fig. 4.4),
- $I$  représente la matrice identité  $4 \times 4$ ,
- $PosObj$  contient les variables inconnues (éléments du vecteur de pose  $PosObj$ ),
- $M_0 \dots M_{n-1}$  représente les variables libres.

Nous notons  $c_{loc\_obj}$  la description du problème de la localisation d'un objet avec la vision stéréoscopique que l'on vient d'établir.

## 4.2.6 Inférence probabiliste

Nous cherchons la position d'un objet à l'aide de la vision stéréoscopique. Grâce à l'état de connaissance définie par la décomposition précédente nous pouvons établir le problème suivant :

Quelle est la probabilité de la position de l'objet sachant que toutes les contraintes géométriques sont respectées ?

L'expression probabiliste correspondante est la suivante :

$$p(PosObj \mid [K_0 = vrai] \dots [K_{n-1} = vrai] c_{loc\_obj}). \quad (4.23)$$

En appliquant sur la décomposition (4.21) le théorème de Bayes :

$$p(PosObj \mid [K_0 = vrai] \dots [K_{n-1} = vrai] c_{loc\_obj}) \\ p(PosObj) \int_{M_0} \dots \int_{M_{n-1}} p(M_0 \dots M_{n-1}) \prod_{i=0}^{n-1} p([K_i = vrai] \mid PosObj M_i) dM_{n-1} \dots dM_0 \quad (4.24)$$

Il est possible de réduire la complexité de l'expression précédente en utilisant les contraintes de fermeture.

L'intégrale de l'expression (4.24) est nulle presque partout sauf pour une valeur bien particulière de  $M_0 \dots M_{n-1} : \{M_0 \dots M_{n-1}\}_0$ . Cette valeur s'obtient en calculant la valeur de la position des points de référence pour une valeur donnée de  $PosObj$  :

$$T_{M_i} = T_{PosObj} * T_{Z_i}, \quad (4.25)$$

L'équation (4.24) devient :

$$p(PosObj \mid [K_0 = vrai] \dots [K_{n-1} = vrai] c_{loc\_obj}) \\ p(PosObj) P(\{M_0 \dots M_{n-1}\}_0) = p(PosObj) P_{M_0 \dots M_{n-1}}(f(PosObj)) \quad (4.26)$$

où  $P_{M_0 \dots M_{n-1}}$  représente les distributions de probabilités associées aux variables libres et  $f$  la fonction décrite par l'expression (4.25).

## 4.2.7 Solution pour la localisation

La tâche est maintenant de choisir une position qui soit une solution au problème de la localisation. Nous choisissons de prendre la position qui maximise  $p(PosObj \mid [K_0 = vrai] \dots [K_{n-1} = vrai] c_{loc\_obj})$ .

$$\max_{PosObj} p \left( PosObj \left| \begin{array}{l} [K_0 = vrai] \\ \vdots \\ [K_{n-1} = vrai] \\ c_{loc\_obj} \end{array} \right. \right) = \max_{PosObj} p(PosObj) \prod_{i=0}^{n-1} p(f_i(PosObj)) \quad (4.27)$$

Le processus de maximisation est décrit dans le paragraphe suivant.



### 4.2.7.1 Maximisation

Étant donnée la dimension relativement importante de l'espace de recherche (6) et la non-convexité de la fonction à optimiser (les fonctions  $f_i$  ne sont pas convexes) nous avons choisi les *algorithmes génétiques* pour chercher la solution. Ils sont bien adaptés pour ce type de problème [Ala94, DBFT99] et bien documentés [FB91, Ala94, Muh92, Gre88].

L'algorithme génétique que nous utilisons est décrit dans l'annexe F, il est classé comme un algorithme génétique continu car la représentation des individus de la population est faite avec des nombres réels.

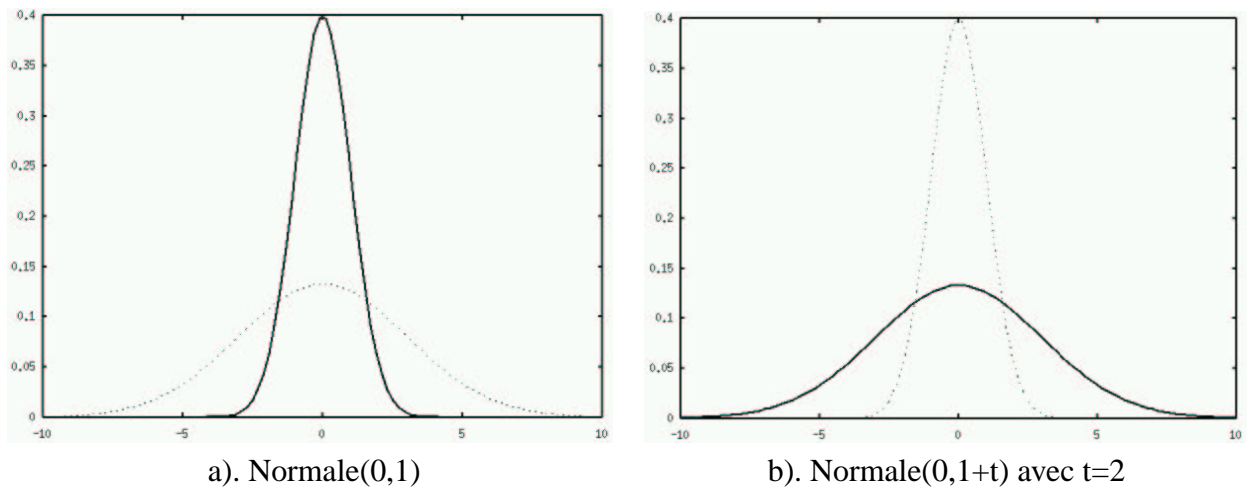


FIG. 4.5 – Distribution normale avec température.

### Température

Les solutions des problèmes de géométrie sont très localisées dans un espace de recherche de grande dimension. Cela veut dire que la fonction d'évaluation est souvent presque nulle pour une valeur particulière de  $PosObj$ . Pour améliorer la recherche et réduire le temps de calcul, nous avons été amenés à relaxer la fonction à optimiser en introduisant un concept de *température*.

Le concept de température consiste à élargir de façon artificielle l'espace ou la fonction prend des valeurs significativement grande. Nous le faisons en augmentant l'écart-type des distributions de probabilité des variables libres du problème comme dans la figure 4.5 :

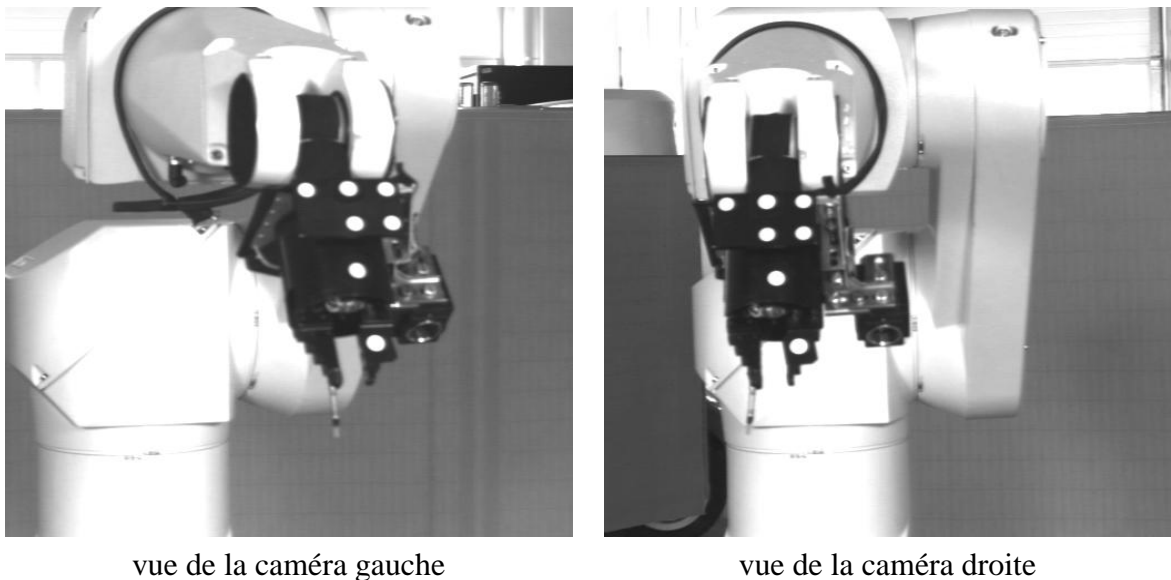
- la figure a représente une distribution  $Normale(0, 1)$ ,
- la figure b représente une distribution  $Normale(0, 1 + t)$  avec  $t = 2$ . Alors, la température change la plage de valeurs pour lesquelles la probabilité est  $> \epsilon$ .

C'est-à-dire que la température agrandit l'espace où l'on trouve des valeurs assez grande de la fonction à optimiser. Au début de la recherche, l'écart-type des distributions de probabilité est augmenté artificiellement de façon à ce que l'on ait des valeurs  $> \varepsilon$  pour tous les individus de la population. Au fur et à mesure de l'amélioration de la population par l'algorithme génétique, la température est réduite à zéro pour arriver à la bonne solution.

#### 4.2.7.2 Exemple pour la localisation

L'exemple montre le processus de la localisation stéréoscopique. Il consiste à trouver la position de la pince du bras robot en utilisant les images sur les caméras de la paire stéréoscopique.

Le modèle de la pince est décrit dans section 3.2.2.



TAB. 4.2 – Pince vue par la vision stéréoscopique.

Les images à partir desquelles la paire stéréoscopique fournit l'information sur la pince sont montrées dans la figure 4.2.

Le processus de calcul de la position d'un point avec la vision stéréoscopique (décrit en B.2) fournit les données montrées dans le tableau (4.3).

Le modèle de l'objet, l'information fournie par la vision stéréoscopique (tableau 4.3) et les formes paramétriques associées aux variables nous permettent de construire le tableau (4.4) contenant toutes les données nécessaires pour faire l'inférence probabiliste.

point	Z		
	x	y	z
0	65.7830	-65.0795	1240.68
1	37.7585	-65.2347	1235.05
2	7.0833	-63.6357	1227.78
3	69.5588	-38.0685	1228.14
4	42.1906	-37.4924	1222.33

TAB. 4.3 – Coordonnées des points sur la pince fournies par la vision stéréoscopique.

Arcs	Distribution de $(x, y, z, r_x, r_y, r_z)$	Type de paramètre
$M_0$	$M_{0.x} = Normale(65.7830, 0.02)$	L
	$M_{0.y} = Normale(-65.0795, 0.02)$	L
	$M_{0.z} = Normale(1240.6800, 0.03)$	L
$M_1$	$M_{1.x} = Normale(37.7585, 0.02)$	L
	$M_{1.y} = Normale(-65.2347, 0.02)$	L
	$M_{1.z} = Normale(1235.0500, 0.03)$	L
$M_2$	$M_{2.x} = Normale(7.0833, 0.02)$	L
	$M_{2.y} = Normale(-63.6357, 0.02)$	L
	$M_{2.z} = Normale(1227.7800, 0.03)$	L
$M_3$	$M_{3.x} = Normale(69.5588, 0.02)$	L
	$M_{3.y} = Normale(-38.0685, 0.02)$	L
	$M_{3.z} = Normale(1228.1400, 0.03)$	L
$M_4$	$M_{4.x} = Normale(42.1906, 0.02)$	L
	$M_{4.y} = Normale(-37.4924, 0.02)$	L
	$M_{4.z} = Normale(1222.3300, 0.03)$	L
PosObj	$PosObj.x = Uniforme[-900.00, 900.00]$	X
	$PosObj.y = Uniforme[-900.00, 900.00]$	X
	$PosObj.z = Uniforme[0.00, 2000.00]$	X
	$PosObj.r_x = Uniforme[-\pi, \pi]$	X
	$PosObj.r_y = Uniforme[-\pi, \pi]$	X
	$PosObj.r_z = Uniforme[-\pi, \pi]$	X
$Z_0$	$(63.50, -28.00, 10.00, -, -, -)$	$(F, F, F)$
$Z_1$	$(63.50, 0.50, 9.50, -, -, -)$	$(F, F, F)$
$Z_2$	$(63.50, 32.00, 10.00, -, -, -)$	$(F, F, F)$
$Z_3$	$(63.50, -28.00, 40.00, -, -, -)$	$(F, F, F)$
$Z_4$	$(63.50, 0.00, 40.00, -, -, -)$	$(F, F, F)$

TAB. 4.4 – Ensemble d'arcs du graphe probabiliste pour la localisation.

PosObj					
x	y	z	r <sub>x</sub>	r <sub>y</sub>	r <sub>z</sub>
25	-46	1295	-1.5541	1.3924	0.7561

TAB. 4.5 – Position de la pince trouvée par la localisation stéréoscopique.

En utilisant la méthode d'inférence décrite, nous arrivons à trouver la solution montrée dans le tableau (4.5). Elle représente les coordonnées du vecteur de pose *PosObj* qui est la relation géométrique entre l'objet et la vision stéréoscopique.

### 4.2.8 Conclusion

Nous avons utilisé l'approche probabiliste pour résoudre le problème d'inversion géométrique consistant à calculer la position d'un objet par le système de vision stéréoscopique à partir d'un ensemble de points le caractérisant. L'intérêt majeur de l'approche est la prise en compte des incertitudes liées à la précision des caméras du système.

Nous avons abordé le problème lié à la résolution d'intégrales toujours impliqués dans le modèle bayésien et nous avons montré qu'à partir de nos *a priori* le problème peut être contourné en utilisant les contraintes de fermeture.

La solution choisie est celle maximisant la distribution de probabilités définie par la question en utilisant un algorithme génétique. Pour accélérer la recherche de la solution, nous avons relaxé la fonction à optimiser en agrandissant les écart-types des variables libres pour les premiers pas de la maximisation et en les réduisant au fur et à mesure de la recherche.

Cette méthode de localisation fournit la position de l'objet dans le repère du système de vision stéréoscopique. Cette position doit être fournie au bras dans son repère afin qu'il le saisisse. Dans l'annexe B.3, nous calculons la transformation permettant de passer d'un de ces repères à l'autre.

## 4.3 Modèle de saisie

Dans la section précédente, nous avons appris à calculer la position d'un objet dans l'espace cartésien (c.-à-d., le repère lié au système de vision stéréoscopique). Dans l'annexe B.3, nous avons montré comment passer de cette position de l'objet dans le repère du système stéréoscopique à sa position dans le repère du bras. En effet, nous avons choisi de faire l'ensemble des manipulations en utilisant le repère lié au bras. Nous devons donc maintenant déterminer la position de saisie de l'objet (c.-à-d., la position à laquelle la pince doit se positionner pour saisir l'objet) à partir de la position de l'objet. C'est un problème complexe, car pour un objet particulier, il existe de nombreuses positions de prises possibles. Nous allons modéliser ces positions

par une distribution de probabilité.

Nous allons utiliser un mélange (ou mixture) de distributions normales univariées pour représenter l'ensemble des positions de prises. Les fondements mathématiques des mélanges de distributions se trouvent dans [RdTG98]. Nous présentons dans l'annexe A.3 les principes de base pour les utiliser. Dans ce paragraphe nous nous limiterons à leur application.

Nous allons nous intéresser plus particulièrement aux prises que l'on peut décrire dans un plan.

La position de prise que nous obtenons est déterminée à partir de la position de l'objet : ce qui veut dire que la position de prise est déterminée dans le repère lié à l'objet à saisir. Pour finir, nous expliquons comment nous calculons la position de prise dans le repère lié au bras.

### 4.3.1 Description de la prise

Pour un objet quelconque, nous pouvons définir des plans de prise dans le repère de l'objet. Pour chaque plan, il sera possible d'avoir plusieurs positions de prise. Sans perte de généralité, nous limiterons nos explications à un seul plan de prise.

La figure 4.6a représente un objet de forme irrégulière sur lequel on définit un plan de prise. Ce plan, placé au centre de l'objet, (figure 4.6b) est bien adapté à la saisie par une pince à mors parallèle.

Une *position de prise* (figure 4.7) sur ce plan est définie au moyen d'un vecteur de pose,

$$pprise = (x, y, \theta)^T$$

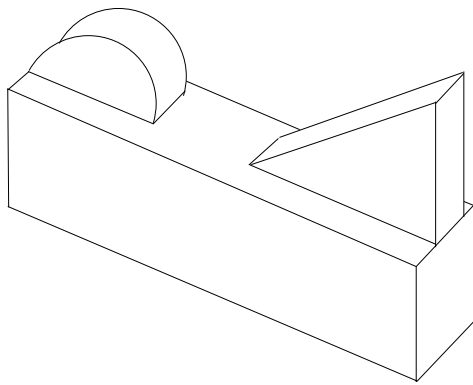
Ce vecteur représente la position de la pince dans le plan. Ce plan est lui même référencé à partir du repère de l'objet :

1. les coordonnées  $x$  et  $y$  de la position de la pince dans le plan de prise,
2. l'orientation  $\theta$  de la pince dans le plan de prise. Elle indique la rotation autour de l'axe  $z$  perpendiculaire au plan de prise.

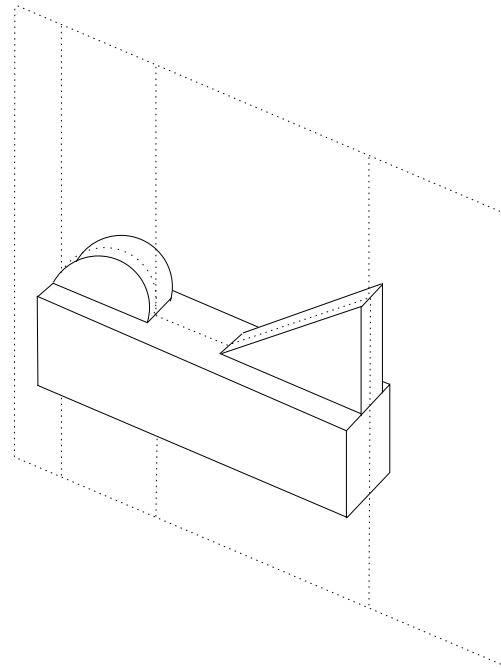
La figure 4.6d représente le plan avec les deux positions de prise indiquées par des vecteurs de pose.

### 4.3.2 Description de l'état de connaissance

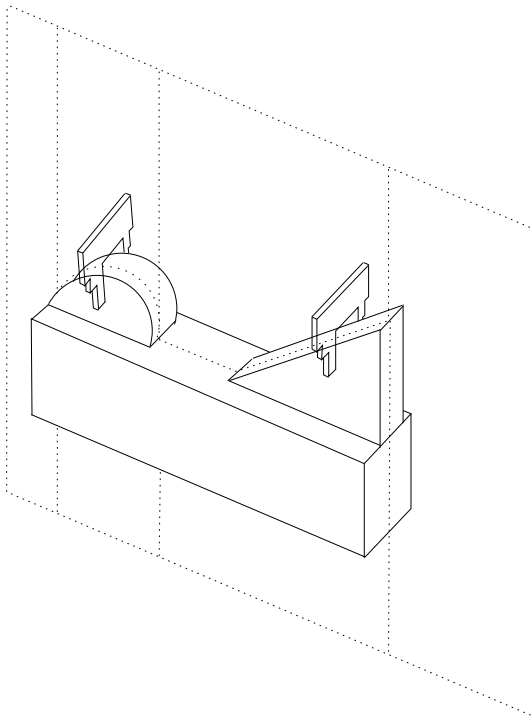
Nous exprimons la diversité des positions de prises sur le plan en utilisant un mélange de distributions de probabilités normales. L'idée est de définir un ensemble discret de lieux de prises



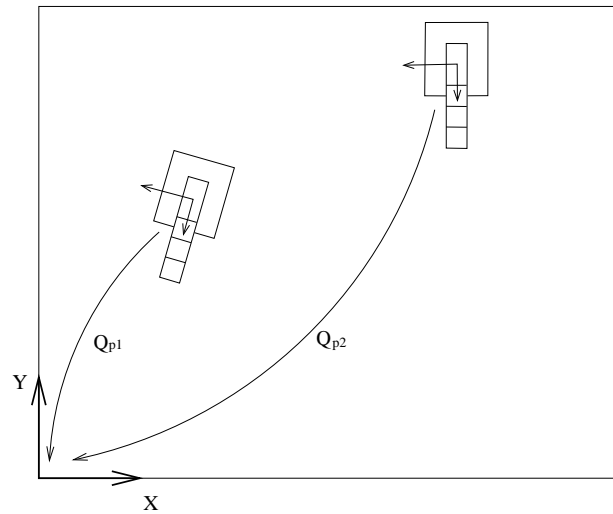
a). Objet



b). Un plan de prise sur l'objet



c). Possibilités de prises de l'objet



d). Plan avec prises

FIG. 4.6 – Objet avec plan de prise et prises.

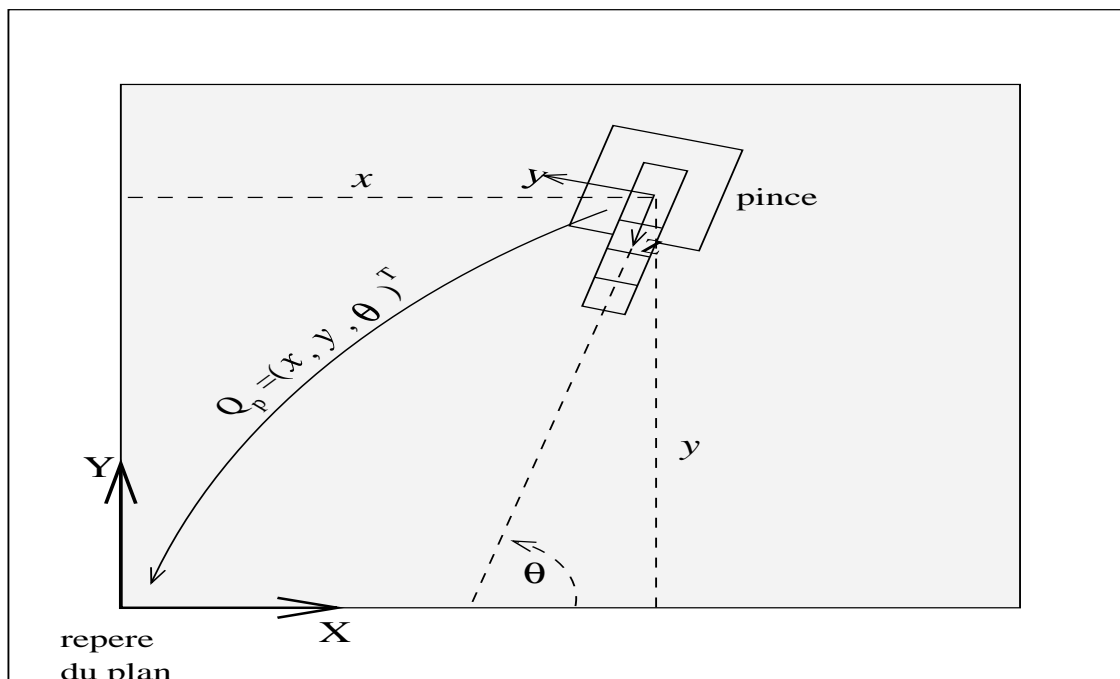


FIG. 4.7 – Plan de prise avec les paramètres de la pose de prise

appelés  $D$  et de définir, pour chacun de ces lieux, une distribution de probabilité sur les paramètres de prise.

Nous utilisons des distributions de probabilités normales univariées car elles facilitent l'apprentissage et l'inférence (voir annexe A.3). Le mélange de distributions de probabilités normales représentant les positions de prises peut s'écrire comme une distribution conjointe :

$$p(\text{prise } D) = p(x \ y \ \theta \ D)$$

où :

- $x$  est une variable probabiliste à valeur dans  $\mathbb{R}$  représentant l'abscisse de la position de prise dans le repère du plan de prise,
- $y$  est une variable probabiliste à valeur dans  $\mathbb{R}$  représentant l'ordonnée de la position de prise dans le repère du plan de prise,
- $\theta$  est une variable probabiliste à valeur dans  $[0, 2\pi]$  représentant l'orientation de la pince dans le plan de prise,
- $D$  est un variable discrète  $[1, 2, \dots, m]$  représentant l'ensemble des lieux de prise.

Un mélange de distributions de probabilités normales univariées est défini par la décomposition suivante :

$$p(x \ y \ \theta \ D) = p(D) p(x|D) p(y|D) p(\theta|D) \quad (4.28)$$

où,

- $p(D)$  - représente la distribution de probabilités sur les lieux de prises.
- $p(x|D)$  - est la distribution de probabilités normale sur  $x$  sachant le lieu de prise.
- $p(y|D)$  - est la distribution de probabilités normale sur  $y$  sachant le lieu de prise.
- $p(\theta|D)$  - est la distribution de probabilités normale sur  $\theta$  sachant le lieu de prise.

Pour instantier les formes paramétriques de chaque facteur de la décomposition, nous utilisons l'algorithme *EM* [DLR77, RW84, MK97] modifiée par [RdTG98]. L'apprentissage est réalisé à partir d'un ensemble de prises valides donné par l'utilisateur.

L'apprentissage des paramètres implique de :

1. définir le nombre  $m$  de composantes du mélange,
2. fournir un ensemble de prises valides et d'utiliser l'algorithme *EM* (décrit dans l'annexe A.3.4) pour calculer les paramètres.

L'ensemble de prises utilisées pour l'apprentissage ainsi que les hypothèses faites pour construire la distribution représentent une connaissance indiquée par  $c_{prise}$ . Pour rester cohérent avec notre formalisme, nous l'ajoutons dans l'expression de la distribution de probabilités conjointe.

$$p(p_{prise} D c_{prise}) = p(x y \theta D c_{prise}).$$

Dans le paragraphe suivant, nous montrons comment choisir une position de prise en utilisant ce mélange.

### 4.3.3 Inférence d'une prise

Le choix d'une position de prise peut se faire de diverses façons. Par exemple, nous pouvons chercher la solution maximisant la probabilité ou choisir une solution par des tirages aléatoires.

Nous avons choisi cette deuxième option car elle est très facile à mettre en oeuvre par la méthode de Gibbs.

Un nombre  $d_0$  est tiré selon la distribution :

$$p(D | c_{prise})$$

Ensuite, nous utilisons l'algorithme décrit en annexe D pour faire les tirages selon chacune des autres distributions :

$$p(x | D = d_0 c_{prise})$$

$$p(y | D = d_0 c_{prise})$$

$$p(\theta | D = d_0 c_{prise})$$

Nous obtenons finalement la position de prise  $p_{prise} = (x, y, \theta)$  sur le repère de l'objet.



La position de prise doit être fournie au bras pour qu'il puisse prendre l'objet, pour cela, il faut connaître la position de l'objet sur le repère du bras et faire la transformation suivante :

$$PosPrise = PosObj * pprise \quad (4.29)$$

cette transformation détermine la position de prise de l'objet  $PosPrise$  dans le repère du bras à partir de la position de l'objet  $PosObj$  dans le repère du bras et de la position de prise de l'objet  $pprise$  dans le repère de l'objet.

La décomposition correspondante s'écrit :

$$p(PosPrise | PosObj) = p(PosObj) p(PosPrise | PosObj) \quad (4.30)$$

où,

- $p(PosObj)$  - représente la distribution de probabilités sur les positions possibles de l'objet dans l'environnement. Cette distribution est uniforme.
- $p(PosPrise | PosObj)$  représente la distribution de probabilités sur les positions possibles de prise de l'objet à partir de la position de l'objet. Cette distribution est un dirac sur la position de prise  $PosPrise$  définie dans l'expression (4.29).

La question probabiliste est directe (c.-à-d., sans inférence) :  $p(PosPrise | PosObj)$

#### 4.3.4 Conclusion

L'utilisation d'un mélange de distributions normales univariées est une méthode simple pour représenter un ensemble de prises sur un plan. Elle peut être facilement étendue à plusieurs plans en rajoutant une autre variable aléatoire pour sélectionner le plan de prise. De plus les méthodes développés pour ce type de modélisation permettent un apprentissage et une sélection simple à mettre en oeuvre.

### 4.4 Conclusion

Dans ce chapitre, nous avons tout d'abord défini un processus pour amener le bras d'un point de départ à un point d'arrivée. La commande est faite en vitesse. C'est-à-dire que l'on détermine, tous les dix millièmes de seconde, les vitesses des six articulations du bras pour qu'il se déplace. Ce programme sera utilisée, comme nous le verrons dans le chapitre suivant, dans des tâches complexes impliquant le mouvement du bras.

Nous avons aussi proposé une méthode probabiliste pour localiser un objet. Cela pose le problème d'inversion géométrique consistant à calculer la position d'un objet par le système de vision stéréoscopique à partir d'un ensemble de points le caractérisant. La programmation probabiliste propose une solution naturelle à ce problème de part sa capacité de fusionner des formes paramétriques et de les inverser lors de l'inférence. La décomposition correspondante à

ce problème (voir eqn. 4.21) est intéressante dans la mesure où elle permet d'exprimer d'une part nos *a priori* sur la position de l'objet et sur la localisation des points  $M_i$  dans l'espace cartésien. Elle permet aussi, d'autre part, d'exprimer le produit des distributions décrivant les contraintes de fermeture connaissant la position de l'objet et de la position du point  $M_i$  dans l'espace cartésien. Ce produit peut être vu comme une manière de fusionner les différentes informations provenant des points  $M_i$ .

Ensuite, l'approche probabiliste de la fusion d'informations présente l'avantage de pouvoir répondre à n'importe quel type de question sur ce modèle. Nous ne sommes pas contraints par un modèle direct ne permettant de répondre qu'à certaines questions. Dans la décomposition, toutes les variables jouent mathématiquement le même rôle et les notions même de modèle direct et inverse n'ont plus vraiment de sens.

En pratique, c'est typiquement ce que l'on fait pour résoudre ce problème de localisation. Lors de la décomposition, à partir de la donnée de la position de l'objet et d'un point  $M_i$ , on spécifie la contrainte de fermeture correspondante ainsi que sa vraisemblance. Ensuite, lors de l'inférence, on cherche la "meilleure" position de l'objet à partir de l'ensemble des contraintes de fermeture.

Un autre intérêt majeur de l'approche est la prise en compte des incertitudes liées à la précision des caméras du système.

D'autre part, le problème de localisation d'un objet a une solution unique qui se trouve généralement dans une petite zone d'un espace de grande dimension. Par conséquent, la recherche de cette solution est réalisée en utilisant un algorithme génétique couplé avec des concepts de température.

Dans la dernière partie de ce chapitre, nous nous sommes servis de cette méthode pour déterminer la position de prise d'un objet à partir de la position de l'objet, c'est-à-dire la position à laquelle doit se trouver la pince pour saisir un objet. Ce programme utilise une forme paramétrique originale : la mixture de distributions normales univariées. Les intérêts de cette forme paramétrique sont qu'elle est facile à construire et à utiliser, et qu'elle a un grand pouvoir d'expression. En plus, elle s'intègre de façon naturelle dans un programme probabiliste.

Enfin, ce programme a permis de mettre en évidence la capacité de réutiliser le résultat d'une question dans un autre programme probabiliste, puisqu'on calcule la position de prise de l'objet en utilisant le résultat du programme de calcul de la position de l'objet.



# Chapitre 5

## Programmation bayésienne de robots

Dans ce chapitre, nous analysons comment la tâche de “pick and place” peut être décomposée sous forme d’opérations élémentaires. A partir de là, nous étudions le programme probabiliste permettant le séquençage de ces opérations.

Nous détaillons les programmes probabilistes permettant au programme de description du “pick and place” de s’interfacer avec les programmes probabilistes interagissant directement avec le robot.

Dans la section 5.3.1, nous présentons l’architecture de contrôle permettant l’intégration et la coordination de l’ensemble des programmes. Cette tâche met en jeu le bras manipulateur, le système de vision et le robot cartésien. Le bras est commandé par inférence. Tous les centième de seconde, l’information fournie par les divers capteurs et l’état actuel du système sont pris en compte pour inférer une distribution de probabilité sur les vitesses des articulations et la commande de la pince. Cette inférence met en jeu quelques centaines de variables et des intégrales imbriquées sur les variables intermédiaires.

Étant donnée la complexité de l’inférence, nous devons faire des choix sur l’estimation des intégrales. Nous décrivons ces choix et nous analysons nos résultats.

### 5.1 Gestion d’activités

Nous allons décrire une opération combinée de saisie et de dépose d’un objet (“pick and place”). Cette opération nécessite l’enchaînement de plusieurs tâches ou activités élémentaires qui seront effectuées par le bras. Ces tâches seront réalisées en fonction des capteurs et des dernières activités réalisées. La méthode utilisée est probabiliste. Elle est dite “inverse” car nous allons nous appuyer sur notre formalisme pour spécifier les lectures capteur auxquelles il faut s’attendre lorsqu’on réalise une activité donnée. Ce mode de spécification est relativement peu usité en robotique. En effet, on a plutôt tendance à spécifier l’inverse : à savoir ce qu’il faut faire étant

données les lectures capteur. La capacité de l'inférence probabiliste d'inverser des problèmes complexes rend possible ce mode de spécification original. La méthode de programmation inverse a été proposée par [Leb99]. Nous l'avons étendue à notre système pour en démontrer la généralité.

Nous allons le détailler suivant la méthodologie que nous avons adoptée depuis le début de ce document.

### 5.1.1 Description du problème

Nous rappelons la tâche de “pick and place” décrite dans la section 3.3. Elle consiste à saisir un objet avec le bras manipulateur et à le déposer à une position précise dans l'environnement : typiquement sur une table dont la position est connue.

Pour cela, le bras manipulateur enchaîne plusieurs opérations élémentaires. Initialement, le bras est immobile en attendant de l'information sur la position de l'objet à saisir. Quand il connaît la position de cet objet, il s'en approche pour le saisir. Ensuite, il ralentit pour positionner la pince. Puis, il s'arrête pour saisir l'objet en fermant la pince. Il se dirige vers la position où il faut poser l'objet. Il ralentit pour positionner la pince, puis s'arrête pour poser l'objet en ouvrant sa pince. Finalement, il vient se replacer dans une position d'attente.

Le problème se décompose en plusieurs activités élémentaires, plusieurs positions géométriques de référence et plusieurs capteurs d'état qui déterminent si le bras se trouve ou non à une de ces positions de référence.

#### 5.1.1.1 Positions géométrique de référence

Dans cette section, nous présentons les 7 positions de référence. Ces positions sont définies dans l'espace de travail du bras.

L'action de *prendre un objet* nécessite la définition de trois positions dans l'espace de travail du bras :

1. La *position de prise* appelée *PosPrise* détermine où prendre l'objet. Un mouvement proche de cette position doit être contraint.
2. La *position d'approche* appelée *PosApp* se déduit de la *position de prise*. Elle permet au bras d'approcher l'objet librement sans prendre en compte les contraintes géométriques de la prise.
3. La *position de distance avec l'objet* appelée *PosElop* détermine l'endroit auquel le bras doit aller après avoir pris l'objet. Cette position peut être similaire ou égale à la *position d'approche*.

En utilisant les positions précédentes, nous décrivons la prise de l'objet ainsi : quand la position de l'objet est connue, le bras va à la *position d'approche*, ensuite il va doucement à la

*position de prise*. Quand il y est, il prend l'objet. Autrement dit, le bras s'arrête et ferme la pince. Après avoir saisi l'objet, le bras s'éloigne doucement, car il est proche de la table où l'objet était posé. Pour bien gérer l'activité après la prise, il va à la *position de distance avec l'objet*.

Quand vient le moment de déposer l'objet, les activités sont semblables, ainsi il faut définir trois nouvelles positions :

1. la *position d'approche avec l'objet* appelée *PosAppp* est une position sûre et proche de la position où le bras doit déposer l'objet,
2. la *position de pose* appelée *PosPos* représente l'endroit où le bras doit déposer l'objet ; le mouvement proche de cette position doit être contraint,
3. la *position de distance* appelée *PosElo* est la position à laquelle le bras doit se diriger après avoir déposé l'objet, elle est en général une position libre de contraintes à partir de laquelle il peut de nouveau partir librement.

Nous définissons une septième position : *position base*. Elle représente l'endroit où le bras attend l'information qui va démarrer la prise ou la pose de l'objet.

### 5.1.1.2 Comportement

Le problème de "pick and place" nécessite plusieurs activités élémentaires. Nous les décrivons de façon intuitive en utilisant la figure 5.1, ainsi :

1. si la position de l'objet n'est pas connue, le bras va à la base ; il y reste tant que l'information sur la position de l'objet n'est pas connue.
2. dès que la position de l'objet est connue, le bras bouge vers la position d'approche.
3. en arrivant à la position d'approche, le bras ralentit et bouge vers la position de prise.
4. en arrivant à la position de prise, le bras s'arrête et ferme la pince pour saisir l'objet.
5. après avoir saisi l'objet, il se dirige vers la position de distance avec l'objet.
6. une fois à la position de distance avec l'objet, si la position de pose est connue, il se dirige vers la position d'approche avec l'objet ; sinon, il reste à la base.
7. quand le bras arrive à la position d'approche avec l'objet, il se dirige vers la position de pose.
8. quand il est à la position de pose, il s'arrête et ouvre la pince pour déposer l'objet.
9. après avoir déposé l'objet, il s'éloigne en se dirigeant vers la position de distance.
10. en arrivant à la position de distance, il peut aller saisir un autre objet, s'il y a déjà l'information sur la position du nouvel objet, sinon il doit aller à la base pour attendre.

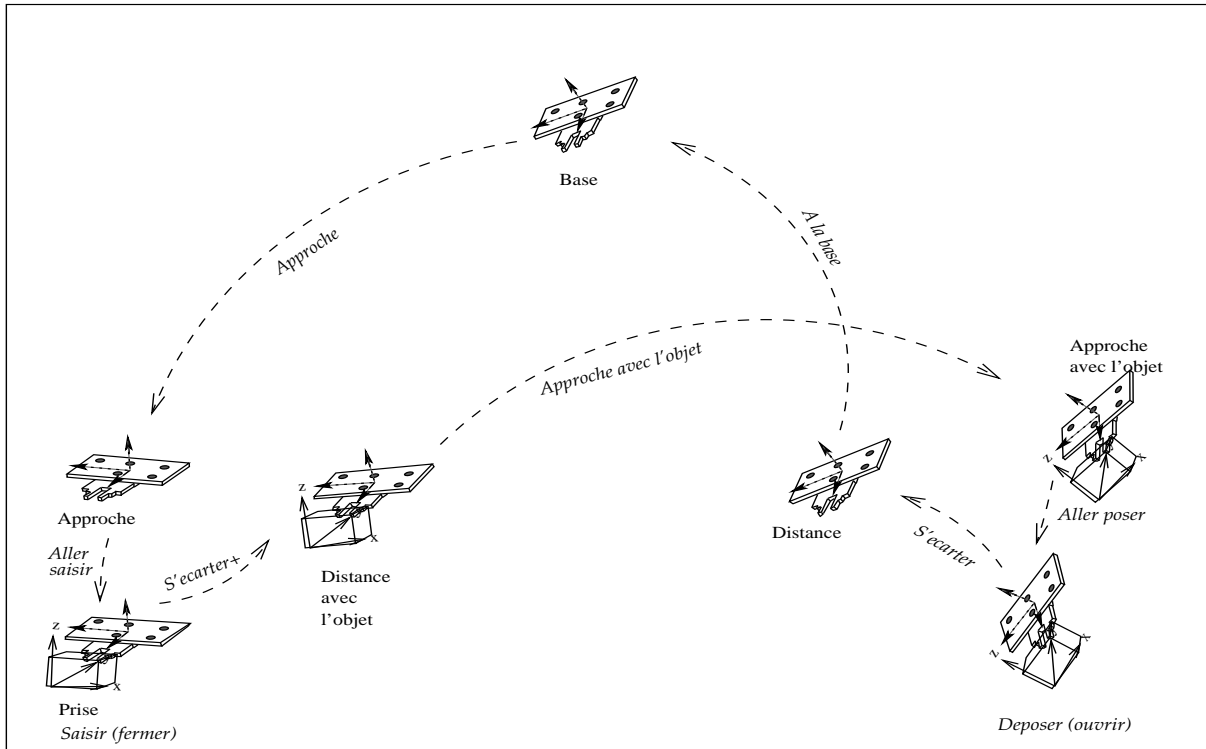


FIG. 5.1 – Comportement du bras

### 5.1.1.3 Capteurs d'état pour les positions géométriques

Nous voyons clairement que le passage d'une activité élémentaire à la suivante a lieu lorsque le bras se trouve à une position de référence. Nous avons donc besoin d'un ensemble de capteurs d'états pour savoir si le bras se trouve à une position physique de référence donnée. Chaque variable d'état est booléenne et représente le fait que le bras se trouve ou non au voisinage d'une position de référence.

1.  $S_{base}$  est vraie quand le bras est à la *position base*.
2.  $S_{app}$  est vraie quand le bras est à la *position d'approche*.
3.  $S_{prise}$  est vraie si le bras est à la *position de prise*. C'est-à-dire que le bras peut saisir l'objet.
4.  $S_{elop}$  est vraie quand le bras est à la *position de distance avec l'objet*.
5.  $S_{apppp}$  est vraie quand le bras se trouve à la *position d'approche avec l'objet*.
6.  $S_{dep}$  est vraie quand le bras est à la *position de pose*.
7.  $S_{elo}$  est vraie à la *position de distance*.

De plus nous avons besoin de 4 variables d'état supplémentaires pour avoir des informations sur l'objet ou l'état de la pince :

1.  $S_{posobjet}$  est vraie quand la position de l'objet est connue.

2.  $S_{postable}$  est vraie quant la position de pose est connue.
3.  $S_{pince}$  vraie si la pince est fermée, fausse si la pince est ouverte.
4.  $S_{objprise}$  vraie si le bras porte l'objet, sinon fausse.

#### 5.1.1.4 Activités élémentaires

Pour finir, nous définissons l'ensemble d'activités élémentaires nécessaires à la conduite du "pick and place" :

1. *Attente*. C'est l'activité du bras quand il est à la position base ou quand l'information fournie par les divers capteurs n'est pas suffisante pour prendre une décision. Sa fonction est d'arrêter le bras,
2. *À la base*, le bras va vers la *position base*,
3. *Approche*, le bras va vers la *position d'approche*,
4. *Aller saisir*, le bras se dirige à la *position de prise*,
5. *Saisir*, le bras est à la *position de prise* et il est en train de fermer sa pince pour prendre l'objet. Durant cette activité, les articulations du bras ne bougent pas. Étant donné que la pince ne sera fermée que pour prendre l'objet, cette activité est équivalent à fermer la pince (*fermer*),
6. *S'écarter avec l'objet*, le bras porte l'objet et il se dirige vers la *position de distance avec l'objet*,
7. *Approche avec l'objet* le bras transporte l'objet vers la *position d'approche avec l'objet*,
8. *Aller déposer*, le bras porte l'objet vers la *position de pose*,
9. *Ouvrir*, le bras ouvre sa pince pour déposer l'objet cependant, il peut arriver que la pince soit fermée sans avoir l'objet. Il faut alors l'ouvrir avant d'aller prendre l'objet.
10. *S'écarter*, le bras bouge vers la *position de distance*, cette activité doit être accomplie après avoir placé l'objet.

De cette description du "pick and place", nous avons un ensemble d'activités et de capteurs que nous utilisons dans les paragraphes suivants pour aborder la gestion d'activités de façon probabiliste.

#### 5.1.2 Variables

Les variables pertinentes du problème sont décrites ainsi :

- $A$  est une variable discrète, elle indique l'action élémentaire actuelle. Elle peut prendre une valeur parmi les dix suivantes :

$$\left\{ \begin{array}{l} \textit{Attente}, \textit{À la base}, \textit{Approche}, \textit{Aller saisir}, \textit{Fermer}, \textit{S'écarter avec l'objet}, \\ \textit{Approche avec l'objet}, \textit{Aller déposer}, \textit{Ouvrir}, \textit{S'écarter} \end{array} \right\}$$



- $A_{-i}$  est une variable discrète, elle représente l'action au temps  $-i$ . Elle permet de suivre l'historique des événements. Elle a le même domaine de valeur que la variable  $A$ .
- $S_j$  est la variable booléenne associée à un capteur d'état  $j$ .

$$j = \left\{ \begin{array}{l} base, app, prise, elop, appp, \\ dep, elo, posobjet, postable, pince, objprise \end{array} \right\}$$

Avec ces variables nous pouvons définir l'expression :

$$p(A A_{-1} \dots A_{-n} S_1 \dots S_K) \quad (5.1)$$

représentant la distribution de probabilité conjointe où  $n$  est le nombre d'événements dans le passé qui sont pris en compte pour faire l'inférence probabiliste et  $K$  le nombre de capteurs d'état.

### 5.1.3 Décomposition

Pour exprimer la dépendance entre l'activité présente et les activités passées nous avons choisi la décomposition suivante :

$$p(A A_{-1} \dots A_{-n} S_1 \dots S_K) = p(A) \prod_{i=1}^n p(A_{-i}|A) \prod_{j=1}^K p(S_j|A) \quad (5.2)$$

Les actions du bras dans le passé sont représentées par les variables,

$$A_{-1} \dots A_{-n}$$

où  $n$  est le nombre d'activités qui ont été déjà faites par le bras et qui sont prises en compte pour faire l'inférence probabiliste. Autrement dit, la mémoire à court terme du bras est représentée par cet ensemble de variables.

Pour définir la décomposition, nous avons fait l'hypothèse que, sachant l'activité actuelle, les valeurs lues sur les capteurs d'état sont indépendantes des actions du passé et ne dépendent que de l'action actuelle.

À partir de la décomposition (expression 5.2), nous allons construire les formes paramétriques associées à ses facteurs.

## 5.1.4 Formes paramétriques

D'après les programmes bayésiens décrits précédemment, nous définissons les formes paramétriques associées à chacun des facteurs de la décomposition. La différence, cette fois, est le type de distributions utilisées. Nous allons en effet utiliser des distributions explicitement décrites par un tableau de probabilité.

- $p(A) = \textit{Uniforme}$ . indique le fait qu'il n'y a pas d'*a priori* sur la valeur de l'activité actuelle,
- $p(A_{-i}|A)$ . La probabilité de l'activité  $A_{-i}$  au temps  $-i$  sachant l'activité actuelle  $A$  est représentée dans le tableau (5.1). Nous détaillons la construction de cette forme paramétrique dans la section suivante.
- $p(S_j|A)$  représente la probabilité du capteur d'état  $j$  sachant l'activité actuelle  $A$ . Ces distributions de probabilité sont décrites dans les tableaux (5.2), (5.3), (5.4) et (5.5). Nous détaillons la construction des tableaux dans la section 5.1.4.2.

Nous allons maintenant commenter certaines valeurs des distributions  $p(A_{-1}|A)$  et  $p(S_j|A)$ .

### 5.1.4.1 Construction de $p(A_{-1}|A)$

La construction du tableau (5.1) a été faite pour respecter le comportement décrit dans la figure 5.1 et en suivant les principes suivants :

1. une transition interdite implique une probabilité nulle,
2. une transition possible implique une valeur de probabilité  $\varepsilon$  (modulable) établie en fonction du comportement attendu et des résultats expérimentaux,
3. la stabilité. Elle modélise le fait que l'activité à l'instant précédent est identique à l'activité actuelle,
4. Le tableau est normalisé par colonne.

Exemple : la distribution de probabilité  $p(A_{-1} | [A = \textit{Approche}])$  est décrite dans la troisième colonne du tableau (5.1). Nous l'interprétons ainsi :

1. Les activités interdites sont :
  - $p([A_{-1} = \textit{Aller saisir}] | [A = \textit{Approche}]) = 0$  - car l'activité actuelle est l'*approche*, cela implique que le bras n'est pas encore passé par la position d'approche : position garantissant la direction de mouvement du bras pour arriver à la position de prise,
  - $p([A_{-1} = \textit{Saisir}] | [A = \textit{Approche}]) = 0$  - pour les mêmes raisons,
  - $p([A_{-1} = \textit{S'écarter}] | [A = \textit{Approche}]) = 0$  - pour les mêmes raisons,
  - $p([A_{-1} = \textit{Approche avec l'objet}] | [A = \textit{Approche}]) = 0$  - pour les mêmes raisons,
  - $p([A_{-1} = \textit{Aller déposer}] | [A = \textit{Approche}]) = 0$  - pour les mêmes raisons,
  - $p([A_{-1} = \textit{Ouvrir}] | [A = \textit{Approche}]) = 0$  - pour les mêmes raisons.

Activité Passée $A_{-1}$	Activité Actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche +l'objet	Aller déposer	Ouvrir	S'écarter
<b>Attente</b>	0.80	0.15	0.10	0.01	0.01	0.01	0.10	0.01	0.01	0.00
<b>À la base</b>	0.06	0.75	0.10	0.01	0.01	0.01	0.10	0.01	0.01	0.00
<b>Approche</b>	0.01	0.01	0.79	0.04	0.00	0.00	0.00	0.00	0.00	0.00
<b>Aller saisir</b>	0.01	0.01	0.00	0.94	0.04	0.00	0.00	0.00	0.00	0.00
<b>Saisir (Fermer)</b>	0.02	0.01	0.00	0.00	0.94	0.10	0.00	0.00	0.00	0.00
<b>S'écarter+</b>	0.03	0.01	0.00	0.00	0.00	0.88	0.10	0.00	0.00	0.00
<b>App+objet</b>	0.01	0.01	0.00	0.00	0.00	0.00	0.70	0.05	0.00	0.00
<b>Aller déposer</b>	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.93	0.04	0.00
<b>Ouvrir</b>	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.94	0.06
<b>S'écarter</b>	0.03	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.94

TAB. 5.1 – Distribution pour l'activité dans le temps  $-1$  en sachant l'activité actuelle :  $p(A_{-1}|A)$ .

2. Les activités possibles avant l'activité *Approche* sont *Attente*, *Aller à la base* et *S'écarter*. Les deux premières sont les activités à faire en cas de problème. Cela est indiqué par :

$$\begin{aligned} p([A_{-1} = \textit{Attente}] | [A = \textit{Approche}]) &= 0.1 \\ p([A_{-1} = \textit{Aller à la base}] | [A = \textit{Approche}]) &= 0.1 \\ p([A_{-1} = \textit{S'écarter}] | [A = \textit{Approche}]) &= 0.01 \end{aligned}$$

indiquant la plausibilité de la transition de l'activité *attente* vers l'activité *approche*. L'activité *S'écarter* peut avoir lieu avant l'activité *Approche* car cela implique que le bras vient de laisser un objet quand il a déjà la position d'un autre. En conséquence il doit s'approcher pour le prendre. La valeur 0.01 indique cette possibilité.

3. La stabilité implique le fait de rester dans la même tâche, cela est indiqué par :

$$p([A_{-1} = \textit{Approche}] | [A = \textit{Approche}]) = 0.79.$$

4. Étant donné qu'il s'agit de probabilités :

$$\sum_{A_{-1}} p(A_{-1} | [A = \textit{Approche}]) = 1.$$

#### 5.1.4.2 Construction de $p(S_j|A)$

La construction des formes paramétriques correspondantes aux capteurs d'état dépend de l'activité. Nous allons étudier leurs distributions sachant l'activité actuelle.

Les tableaux (5.2), (5.3), (5.4) et (5.5) représentent les distributions de probabilités des capteurs d'état en sachant l'activité actuelle.

Pour construire les distributions de probabilité des capteurs d'état, nous avons considéré que les transitions des activités du bras ont en général lieu dans les positions physiques préalablement décrites. À savoir, les positions de base, d'approche, de prise, de distance avec l'objet, d'approche avec l'objet, de pose et de distance. Les principes de construction sont les suivants :

1. si le capteur abstrait est impliqué dans la transition vers l'activité actuelle, la valeur de probabilité vraie est importante,
2. si le capteur n'est pas impliqué dans la transition, on considère que sa valeur la plus probable est fausse,
3. pour le cas des activités alternatives comme : *aller à la base et attente* on n'a pas d'*a priori* alors, on assigne la même probabilité pour les deux valeurs de la variable (distribution uniforme),

$S_{base}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche +l'objet	Aller déposer	Ouvrir	S'écarter
<b>faux</b>	0.2	0.9	0.4	0.9	0.9	0.9	0.4	0.9	0.9	0.9
<b>vrai</b>	0.8	0.1	0.6	0.1	0.1	0.1	0.6	0.1	0.1	0.1

a). Capteur de la *position base*,  $p(enbase | A)$ .

$S_{app}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche l'objet	Aller déposer	Ouvrir	S'écarter
<b>faux</b>	0.5	0.5	0.9	0.2	0.9	0.9	0.9	0.9	0.9	0.9
<b>vrai</b>	0.5	0.5	0.1	0.8	0.1	0.1	0.1	0.1	0.1	0.1

b). Capteur de la *position d'approche*,  $p(enapproche | A)$ .

$S_{prise}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche +l'objet	Aller déposer	Ouvrir	S'écarter
<b>faux</b>	0.5	0.5	0.9	0.999	0.001	0.4	0.9	0.9	0.9	0.9
<b>vrai</b>	0.5	0.5	0.1	0.001	0.999	0.6	0.1	0.1	0.1	0.1

c). Capteur de la *position de prise*,  $p(enprise | A)$ .

TAB. 5.2 – Formes paramétriques pour les capteurs de : position de base, position d'approche et position de prise

$S_{elop}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche +l'objet	Aller déposer	Ouvrir	S'écarter
<b>faux</b>	0.5	0.5	0.9	0.9	0.9	0.99	0.1	0.9	0.9	0.9
<b>vrai</b>	0.5	0.5	0.1	0.1	0.1	0.01	0.9	0.1	0.1	0.1

a). Capteur pour *position de distance avec l'objet*,  $p(\text{endistplus} | A)$ .

$S_{appp}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche +l'objet	Aller déposer	Ouvrir	S'écarter
<b>faux</b>	0.5	0.5	0.9	0.9	0.9	0.9	0.9	0.01	0.9	0.9
<b>vrai</b>	0.5	0.5	0.1	0.1	0.1	0.1	0.1	0.99	0.1	0.1

b). Capteur pour la *position d'approche avec l'objet*,  $p(\text{enapprocheplus} | A)$ .

$S_{dep}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche l'objet	Aller déposer	Ouvrir	S'écarter
<b>faux</b>	0.5	0.5	0.9	0.9	0.9	0.9	0.9	0.9	0.001	0.4
<b>vrai</b>	0.5	0.5	0.1	0.1	0.1	0.1	0.1	0.1	0.999	0.6

c). Capteur pour la *position de pose*,  $p(\text{enpose} | A)$ .

TAB. 5.3 – Formes paramétriques pour les capteurs de : positions de distance avec l'objet, position d'approche avec l'objet et position de pose de l'objet

$S_{elo}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche +l'objet	Aller déposer	Ouvrir	S'écarter
<b>faux</b>	0.5	0.5	0.4	0.9	0.9	0.9	0.9	0.9	0.9	0.999
<b>vrai</b>	0.5	0.5	0.6	0.1	0.1	0.1	0.1	0.1	0.1	0.001

a). Capteur de la *position de distance*,  $p(endistance | A)$ .

$S_{posobjet}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche +l'objet	Aller déposer	Ouvrir	S'écarter
<b>faux</b>	0.7	0.6	0.0001	0.0001	0.0	0.5	0.5	0.5	0.5	0.5
<b>vrai</b>	0.3	0.4	0.9999	0.9999	1.0	0.5	0.5	0.5	0.5	0.5

b). Capteur indiquant si la position de l'objet est connue,  $p(posobjet | A)$ .

$S_{postable}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche +l'objet	Aller déposer	Ouvrir	S'écarter
<b>faux</b>	0.7	0.6	0.5	0.5	0.5	0.5	0.0	0.0001	0.0001	0.2
<b>vrai</b>	0.3	0.4	0.5	0.5	0.5	0.5	1.0	0.9999	0.9999	0.8

c). Capteur indiquant si la position pour placer l'objet est connue,  $p(postable | A)$ .

TAB. 5.4 – Distributions pour : le capteur de la position de distance, le capteur d'information sur la connaissance de la position de l'objet et le capteur de la position pour déposer l'objet

$S_{objprise}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche +l'objet	Aller déposer	Ouvrir	S'écarter
<b>faux</b>	0.5	0.5	1.0	1.0	1.0	0.0	0.0	0.0	0.1	1.0
<b>vrai</b>	0.5	0.5	0.0	0.0	0.0	1.0	1.0	1.0	0.9	0.0

a). Distribution du capteur indiquant si le bras porte l'objet,  $p(objprise | A)$ .

$S_{pince}$	Activité actuelle A									
	Attente	À la base	Approche	Aller saisir	Saisir (Fermer)	S'écarter +l'objet	Approche +l'objet	Aller déposer	Ouvrir	S'écarter
<b>OUVERTE</b>	0.5	0.5	0.9999	0.9999	0.9999	0.0001	0.0001	0.0001	0.0001	0.6
<b>FERMÉE</b>	0.5	0.5	0.0001	0.0001	0.0001	0.9999	0.9999	0.9999	0.9999	0.4

b). Distribution pour l'état de la pince,  $p(pince | A)$ .

TAB. 5.5 – Distributions pour : le capteur indiquant si le bras porte l'objet et le capteur de l'état de la pince



4. les distribution de probabilité sont décrites par colonne ainsi, la somme de chaque colonne est l'unité.

Exemple : la distribution de probabilité  $p(S_{app} | A)$  décrite dans le tableau (5.2b), nous l'interprétons ainsi :

1. l'activité à faire pour le bras après l'activité *approche* est *aller saisir* (voir figure 5.1), étant donné que la transition à lieu à la position d'approche, nous l'indiquons avec :

$$p([S_{app} = vrai] | [A = Aller Saisir]) = 0.8$$

2. quand le capteur  $S_{app}$  n'est pas impliqué dans la transition d'une activité vers une autre, on considère que sa valeur la plus probable est fausse, cela est indique par une valeur de probabilité de 0.9.

$$p([S_{app} = faux] | [A = Aller Déposer]) = 0.9$$

3. Étant données que les activités à faire en cas de problèmes n'ont pas d'*a priori* sur la valeur du capteur,

$$p(S_{app} | [A = Attente]) = Uniforme = \frac{1}{2}$$

Quant aux autres capteurs ( $S_{posobj}$ ,  $S_{postable}$  et  $S_{pince}$ ), ils doivent avoir une probabilité  $\epsilon$  modulable étant donnée l'activité actuelle. Par exemple :

$$p([S_{objprise} = vrai] | [A = Approche]) = 1$$

car cette activité doit se faire uniquement quand on connaît la position de l'objet.

### 5.1.5 Inférence

Pour définir le comportement du bras, il faut choisir l'activité à un instant donné à l'aide de l'information suivante :

- les dernières activités du bras,
- les valeurs actuelles des capteurs.

Nous posons la question suivante :

*Quelle est la probabilité de l'activité actuelle sachant les dernières activités et les valeurs actuelles des capteurs ?*

La représentation de cette question se fait à partir de l'expression (5.2) de la décomposition et du théorème de Bayes,

$$p(A | A_{-1} \dots A_{-n} S_1 \dots S_K C_{activité}) \propto p(A) \prod_{i=1}^n p(A_{-i}|A) \prod_{j=1}^K p(S_j|A) \quad (5.3)$$

Cette expression est facile à calculer car on connaît toutes les distributions de probabilité. La difficulté réside alors dans le choix d'une solution dans l'espace de solutions.

### 5.1.6 Solution

Le choix d'une solution se fait par tirage aléatoire. Cette méthode de choix nous semble adéquate pour laisser au système la possibilité de changer de stratégie entre deux pas de calcul et notamment elle permet de choisir et de changer entre des solutions équiprobables.

Pour faire le tirage aléatoire sur la distribution de probabilité représentant l'espace des solutions, nous devons :

1. Calculer la probabilité de chacune des activités avec l'expression (5.3).
2. Faire le tirage aléatoire sur la fonction de répartition construite dans le pas précédent pour choisir une activité.

### 5.1.7 Résultat

La distribution de probabilité générée par la question (expression 5.3) est en générale très piquée. C'est-à-dire qu'elle détermine clairement l'activité à faire. Dans le cas de gestion d'activités, cela est un avantage car le bras ne fait pas de mauvais choix d'activité le conduisant à refaire des activités déjà faites.

Quand le bras est à la base et la position de l'objet n'est pas connue, les activités choisies sont "attente" ou "aller à la base". La première est l'activité naturellement attendue. "Aller à la base" pourrait s'interpréter comme un mauvais choix cependant, il n'a pas d'incidence car le bras ne bouge pas puisqu'il est déjà à la base.

Les expériences avec la gestion de tâches nous ont amené à considérer trois activités dans le passé, car si le nombre est faible, le bras a tendance à revenir sur les activités déjà faites et quand le nombre d'activités considéré est plus grand, le système a tendance à favoriser l'activité actuelle et à y rester trop longtemps.

### 5.1.8 Conclusion

Notre façon de construire les tableaux représentant les formes paramétriques a démontré son efficacité, le choix d'activité correspond au choix qu'on attend naturellement. En plus, il est rela-

tivement simple d'ajouter de nouvelles activités ou de nouveaux capteurs. Ils sont pris en compte tout de suite par le processus d'inférence. Cette caractéristique d'intégrer des informations nouvelles est naturelle dans notre approche.

La programmation inverse nous permet de réfléchir sur l'information fournie par un capteur et non sur la façon de l'utiliser, cela correspond mieux à la description physique des capteurs. Le processus d'inférence se charge de l'inversion. C'est-à-dire, étant donnée les lectures-capteur et l'état du système, le processus d'inférence détermine l'action.

La construction des formes paramétriques de façon explicite est un labeur ardu qui peut s'aborder de façon automatique, c'est-à-dire que les formes paramétriques soient apprises en utilisant des techniques d'apprentissage comme celle décrite dans l'annexe [A.3](#).

## 5.2 Distribution de probabilités entre positions et actions

La gestion d'activités décrite dans la section précédente définit le séquençement de plusieurs activités dépendantes d'un groupe de capteurs d'état. Dans cette section nous décrivons dans un premier temps la façon de calculer les positions de référence. Ensuite nous décrivons le processus pour calculer les valeurs des capteurs d'état.

Dans le troisième paragraphe, nous définissons la position des buts intermédiaires à atteindre pour chaque activité. Ces buts intermédiaires vont nous servir pour passer d'une activité élémentaire à la suivante. Ils correspondent en fait aux positions de référence associées à chaque action élémentaire.

Pour finir, nous définissons la commande de la pince à partir de l'activité élémentaire courante.

### 5.2.1 Calcul des positions de référence

Les positions de référence ont été décrites dans la section [5.1.1.1](#). Le cas particulier de la position de prise et de la position de pose a été traité dans la section [4.3](#). Dans cette section, la position de prise est choisie à partir de la position de l'objet par des tirages aléatoires sur la distribution  $p(PosPrise | PosObj c_{prise})$ , la position de pose est calculée de la même façon sur la distribution  $p(PosPos | PosTable PosPrise c_{pose})$ .

Dans le prochain paragraphe, nous décrivons la représentation des positions de référence liés à la prise et à la pose de l'objet. Elles sont définies dans le plan de prise de l'objet.

La position de d'approche doit être proche de la position de prise, elle est déduite de celle-ci par :

$$PosApp = PosPrise + AP$$

avec,  $AP = (d_{AP} * \cos(\Theta), d_{AP} * \sin(\Theta), 0)^T$  où,  $d_{AP}$  est une distance constante dépendant de l'objet et de la pince.

Nous voulons définir la position d'approche par une distribution de probabilité uniforme représentant sa plage de valeurs. Cette plage dépend de la taille des doigts de la pince et de la forme de l'objet Ainsi, la forme paramétrique correspondante s'écrit :

$$p(PosApp | PosPrise) = Uniforme[PosPrise + AP - \Delta_{AP}, PosPrise + AP + \Delta_{AP}]. \quad (5.4)$$

avec  $\Delta_{AP} = (d_{\Delta_{AP}} * \cos(\Theta), d_{\Delta_{AP}} * \sin(\Theta), 0)^T$  où,  $d_{\Delta_{AP}}$  est une distance définissant la plage de valeurs de la position d'approche. Elle dépend de l'objet et de la pince.

La figure 5.2 représente la plage de valeurs d'une position d'approche définie à partir d'une position de prise donnée en considérant les distances  $d_{AP}$  et  $d_{\Delta_{AP}}$ . En fait, la distribution uniforme définit une région ayant les limites suivantes :

- la plage de valeurs pour l'abscisse de la position d'approche est,

$$[PosPrise.x + (d_{AP} - d_{\Delta_{AP}}) * \cos(\Theta), PosPrise.x + (d_{AP} + d_{\Delta_{AP}}) * \cos(\Theta)],$$

- la plage de valeurs pour l'ordonnée de la position d'approche est,

$$[PosPrise.y + (d_{AP} - d_{\Delta_{AP}}) * \sin(\Theta), PosPrise.y + (d_{AP} + d_{\Delta_{AP}}) * \sin(\Theta)]$$

Nous utilisons la même représentation pour les autres positions de référence. Ainsi, la forme paramétrique de :

- la position d'éloignement avec l'objet est,

$$p(PosElop | PosPrise) = Uniforme[PosPrise + ELP - \Delta_{ELP}, PosPrise + ELP + \Delta_{ELP}],$$

- la position d'approche à la table avec l'objet est,

$$p(PosApppp | PosPos) = Uniforme \left[ \begin{array}{l} PosPos + APPP - \Delta_{APPP}, \\ PosPos + APPP + \Delta_{APPP} \end{array} \right],$$

- la position d'éloignement de la table est,

$$p(PosElo | PosPos) = Uniforme[PosPos + EL - \Delta_{EL}, PosPos + EL + \Delta_{EL}].$$

Dans le paragraphe suivant, nous définissons les formes paramétriques pour les capteurs d'état basés sur les positions précédemment décrites.

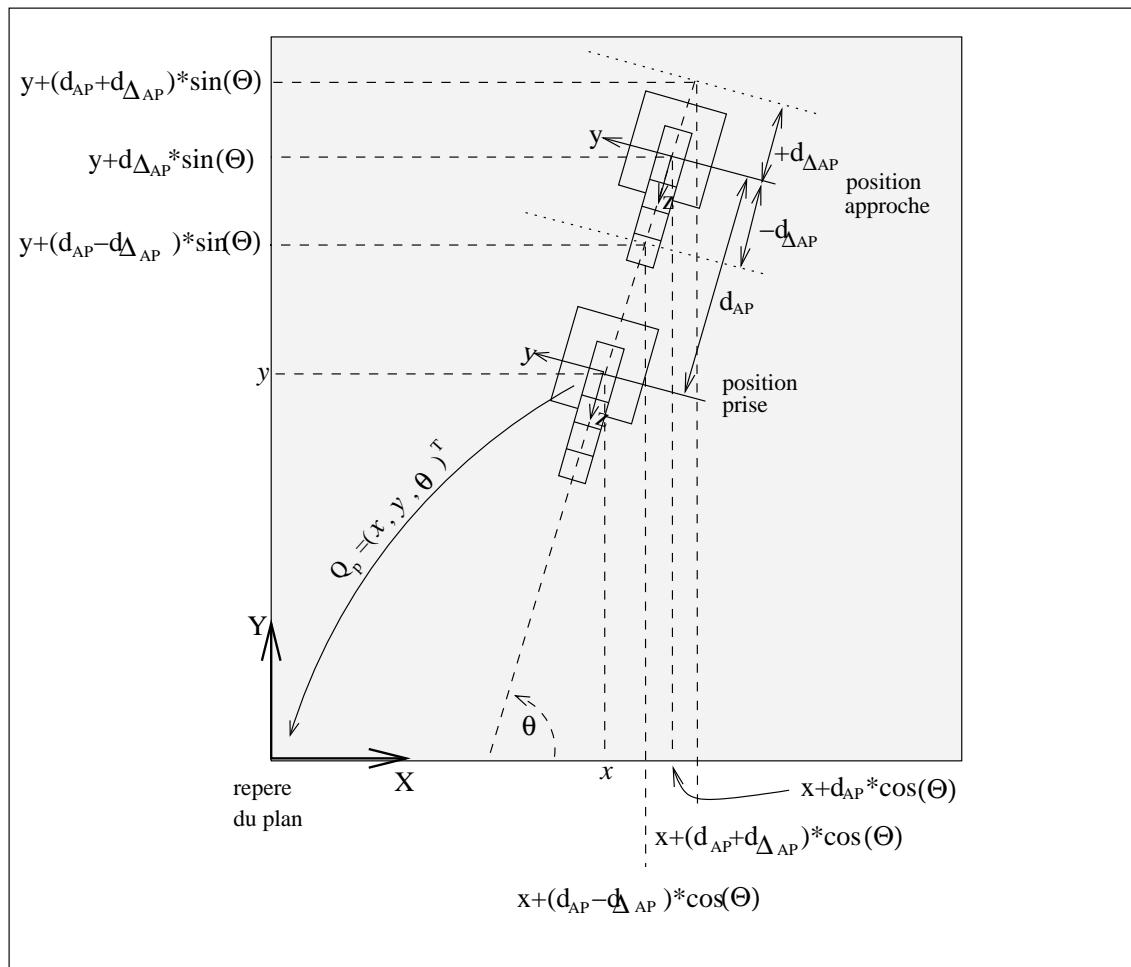


FIG. 5.2 – Position d'approche.

## 5.2.2 Calcul des capteurs d'état

L'ensemble des capteurs d'état utilisé pour gérer la gestion de tâches nécessaires au processus de "pick and place" a été défini dans la section 5.1.1.3. Chaque capteur d'état est défini à partir d'une position associée, que nous appelons sa position de référence, et de la position du bras.

Pour calculer la valeur d'un capteur de position, nous utilisons la distribution de probabilité suivante :

$$p([S_r = vrai] | Pos_r X) = \begin{cases} 1 & \text{si } \|Pos_r - X\| < \epsilon_r, \\ 0 & \text{sinon.} \end{cases}$$

Le capteur  $S_r = vrai$  si, la distance entre la position de référence  $Pos_r$ , correspondant au capteur  $S_r$ , et la position  $X$  du bras est plus petite que l'erreur  $\epsilon_r$  permise.

Suivant ce principe nous pouvons définir :

$$\begin{aligned}
& p(S_{base} | Base X), \\
& p(S_{app} | PosApp X), \\
& p(S_{prise} | PosPrise X), \\
& p(S_{elop} | PosElop X), \\
& p(S_{apppp} | PosApppp X), \\
& p(S_{dep} | PosPos X), \\
& p(S_{elo} | PosElo X).
\end{aligned}$$

Ces distributions établissent des liens entre l'espace de travail du robot et le processus de gestion d'activités nécessaire à la tâche de "pick and place". Elles définissent les valeurs des capteurs d'état utilisés pour repérer les positions de référence.

Par exemple, pour une position de référence  $PosApp = posapp$ , une position du bras  $X = x$  et une erreur permise  $\epsilon_{PosApp} = erreur$  nous avons

$$p([S_{PosApp} = vrai] | [PosApp = posapp] [X = x]) = \begin{cases} 1 & \text{si } \|posapp - x\| < \epsilon_{PosApp}, \\ 0 & \text{sinon.} \end{cases}$$

### 5.2.3 Calcul de la position but

Dans le paragraphe précédent, nous avons établi des liens entre l'espace de travail (typiquement les positions de référence) et les capteurs d'état. Maintenant, nous établissons des liens dans le sens inverse où, étant donnée une activité, il faut définir une position de référence à atteindre pour passer à l'activité suivante. Par exemple, si l'activité actuelle est *Approcher* alors le but est *PosApp*. C'est en effet la position à atteindre pour passer de l'activité *Approcher* à l'activité *Prise*.

La position but a été définie dans la section 4.1.5. Elle représente la position vers laquelle le bras se dirige. Nous calculons sa valeur avec la forme paramétrique suivante :

$$p(But | A) = \delta_{(c(A))}(But)$$

où la fonction  $c(A)$  est définie par :

$$\begin{aligned}
c(A = \text{\textit{À la base}}) &= Base \\
c(A = \text{\textit{Approcher}}) &= PosApp \\
c(A = \text{\textit{Aller Saisir}}) &= PosPrise \\
c(A = \text{\textit{Fermer}}) &= PosPrise \\
c(A = \text{\textit{S'écarter avec l'objet}}) &= PosElop \\
c(A = \text{\textit{Approcher avec l'objet}}) &= PosApppp
\end{aligned}$$

$$\begin{aligned}
c(A = \textit{Aller déposer}) &= \textit{PosPos} \\
c(A = \textit{Ouvrir}) &= \textit{PosPos} \\
c(A = \textit{S'écarter}) &= \textit{PosElo}
\end{aligned}$$

impliquant que l'activité détermine la position à laquelle se dirige le bras.

### 5.2.4 Calcul de la commande de la pince

De la même façon que dans le paragraphe précédent, la commande de la pince est directement liée à certaines actions dans le processus de gestion. Nous établissons ce lien avec :

$$p(MPince | A) = \delta_{c(A)}(MPince). \quad (5.5)$$

Le dirac est défini ainsi :

$$\begin{aligned}
c(A = \textit{Fermer}) &= \textit{fermer la pince} \\
c(A = \textit{Ouvrir}) &= \textit{ouvrir la pince}
\end{aligned}$$

Cette forme paramétrique implique que l'activité est liée à la commande physique de la pince.

### 5.2.5 Conclusion

Dans cette section nous avons défini les formes paramétriques nous permettant de lier les positions aux activités nécessaires à la gestion du comportement du bras. Nous avons établi les liens entre les positions de référence et des abstractions de ces positions : les capteurs d'état. Nous avons aussi établi les liens directs existants entre les activités et les commandes du bras et de la pince.

Dans la section suivante, ces formes paramétriques et l'ensemble de processus probabilistes définis précédemment sont intégrés dans une seule distribution de probabilités nous permettant de conduire le bras à réaliser une tâche de "pick and place".

## 5.3 Intégration pour programmation du "Pick and Place"

Dans cette section, nous allons spécifier un comportement complexe en utilisant la programmation bayésienne. La tâche consiste à saisir puis à déposer un objet repéré par la vision stéréoscopique. Les activités impliquées dans cette opération seront gérées par l'information fournie par les capteurs et par l'état du système.

### 5.3.1 “Pick and Place”

Le problème de “pick and place” implique d’interpréter l’information fournie par le système de vision stéréoscopique de façon à conduire le bras à prendre l’objet et ensuite amener cet objet sur la table. Nous citons les processus probabilistes déjà décrits pour cette tâche et les questions probabilistes correspondantes à chaque processus.

1. La calibration de caméras (annexe B.1). Ce processus calcule la matrice de projection d’une caméra à partir des points dans l’espace et de leurs projections sur l’image de la caméra. La question résolvant le problème est :

$$p(P | Z_1 \dots Z_L B_{g_1} \dots B_{g_L} c_{cam}),$$

où  $P$  est la matrice de projection de la caméra,  $Z_i$  représente un point caractéristique sur le repère de l’objet et  $B_{g_i}$  représente la projection de ce point dans l’image de la caméra.

2. La localisation de points dans l’espace avec le système de vision (annexe B.2.2) consiste à calculer un point dans l’espace à partir de ses projections sur les caméras du système de vision stéréoscopique. Le problème est posé par la question suivante :

$$p(M | P_g P_d B_g B_d c_{VS}),$$

où  $M$  représente le point dans l’espace,  $B_g$  est la projection de ce point dans la caméra gauche et  $B_d$  est la projection du point dans la caméra droite.  $P_g$  et  $P_d$  sont les matrices de projections des caméras gauche et droite du système de vision.

3. La localisation d’objets par le système de vision (section 4.2). À partir des points de référence ( $Z_i$  correspondants à la description géométrique de l’objet) et leurs positions ( $M_i$ ) estimées par le système de vision, la position de l’objet est estimée. La question correspondante s’écrit :

$$p(PosObj | K_1 \dots K_L M_1 \dots M_L Z_1 \dots Z_L c_{loc\_Obj}).$$

4. Le modèle de saisie (section 4.3) consiste à déterminer la position à laquelle doit se trouver la pince pour saisir un objet. Cette position de prise est choisie par des tirages aléatoires sur la mixture de distributions de probabilités normales qui la représente. La question correspondante s’écrit :

$$p(PosPrise | PosObj c_{prise}).$$

De façon similaire pour la position physique de pose :

$$p(PosPos | PosPrise PosObj c_{pose}).$$

5. La gestion d’activités (section 5.1) consiste à gérer le processus de “pick and place” au moyen d’un ensemble d’activités. Pour déterminer l’activité, on pose la question suivante :

$$p(A | A_{-1} A_{-2} A_{-3} S_{base} \dots S_{objprise} c_{activité}).$$



6. Les positions d’approche et de distance avec l’objet (section 5.2.1) sont représentées par :

$$p(PosApp | PosPrise)$$

$$p(PosElop | PosPrise).$$

7. Les positions d’approche avec l’objet et de distance sachant la position de pose sont décrites par :

$$p(PosAppp | PosPos)$$

$$p(PosElo | PosPos)$$

8. Les capteurs d’état sont fonctions des positions qui leur sont associées et de la position du bras dans l’espace de travail (section 5.2.2). Ils sont décrits par :

$$p(S_r | Pos_r X).$$

9. la position but (section 5.2.3) est décrite par :

$$p(But | A).$$

10. la commande de la pince (section 5.2.4) est représentée par :

$$p(MPince | A).$$

11. Le modèle de mouvement (section 4.1) consiste à calculer la vitesse de commande des articulations du bras à partir de la position actuelle du bras, du but courant à atteindre et de la dernière vitesse commandée. La question représentant le problème s’écrit :

$$p(\dot{\Theta} | \Theta \dot{\Theta}_{-1} But E c_{mov}).$$

Dans les paragraphes suivants, nous décrivons le programme probabiliste correspondant au “pick and place”. En suivant notre approche, nous commençons par la définition des variables pertinentes.

### 5.3.2 Variables

Les variables pertinentes pour le problème de “pick and place” sont celles de l’ensemble impliqués. Ce sont les suivantes :

- $Z_i$ , vecteur de pose représentant un point de référence placé sur l’objet ou sur la table (avec  $i = 1 \dots N$  où,  $L$  points appartiennent à l’objet et  $N - L$  à la table),
- $B_{g_i}$ , projection du point de référence  $Z_i$  sur l’image de la caméra gauche,
- $B_{d_i}$ , projection du point de référence  $Z_i$  sur l’image de la caméra droite,
- $P_g$ , matrice de projection de la caméra gauche,

- $P_d$ , matrice de projection de la caméra droite,
- $M_i$ , vecteur de pose représentant le point de référence  $Z_i$  sur la paire stéréoscopique,
- $K_i$ , représente la condition de fermeture du cycle cinématique correspondant au point de référence  $M_i$  (localisation d'un objet avec la vision stéréoscopique en utilisant la CAO-Robotique, voir section 4.2),
- $PosObj$ , vecteur de pose représentant la *position de l'objet*,
- $PosTable$ , similaire à  $PosObj$ . Il indique la *position de la table* où l'objet doit être placé,
- $PosPrise$ , vecteur de pose représentant la *position de prise* de l'objet,
- $PosApp$ , *position d'approche* à l'objet,
- $PosElop$ , *position de distance avec l'objet*,
- $PosPos$ , *position de pose* de l'objet sur la table,
- $PosAppp$ , *position d'approche avec l'objet* à la table. Le bras porte l'objet,
- $PosElo$ , *position de distance* de la table après avoir déposé l'objet,
- $\Theta$ , position articulaire actuelle du bras robot,
- $X$ , position cartésienne actuelle du bras robot,
- $S_{base}$ , représente le capteur abstrait de la *position Base*,
- $S_{app}$ , représente le capteur abstrait de la *position d'approche* à l'objet,
- $S_{prise}$ , représente le capteur abstrait de la *position de prise* de l'objet,
- $S_{elop}$ , représente le capteur abstrait de la *position de distance avec l'objet*,
- $S_{appp}$ , représente le capteur abstrait de la *position d'approche avec l'objet*,
- $S_{dep}$ , représente le capteur abstrait de *position de pose*,
- $S_{elo}$ , représente le capteur abstrait de la *position de distance*,
- $S_{pince}$ , représente l'état de la pince ; *vraie* si la pince du bras est *ouverte* et *fausse* sinon,
- $S_{objprise}$ , elle est *vraie* si le bras porte l'objet et *fausse* sinon,
- $A_{-1}$ , représente l'activité du bras au temps  $t - 1$ ,
- $A_{-2}$ , représente l'activité du bras au temps  $t - 2$ ,
- $A_{-3}$ , représente l'activité du bras au temps  $t - 3$ ,
- $A$ , représente l'activité actuelle du bras,
- $But$ , c'est la position, en coordonnées cartésiennes, vers laquelle le bras se dirige,
- $MPince$ , représente la commande de la pince ; les actions sont *ouvrir* et *fermer*,
- $E$ , représente l'erreur permise pour la commande de la vitesse cartésienne du bras,
- $\dot{\Theta}_{-1}$ , vitesse articulaire au temps  $t - 1$ ,
- $\dot{\Theta}$ , vitesse articulaire actuelle.

Le nombre de variables impliquées dans le problème est donc fonction du nombre de points de référence utilisés pour représenter l'objet et la table. Si on considère 10 points de référence pour l'ensemble de l'objet et la table, le nombre de variables impliquées est de 207. Ce nombre est obtenu ainsi :

- 30 pour les points de référence de l'objet et de la table (les  $Z_i$ ) car chaque point de référence à trois coordonnées dans l'espace,
- 20 pour les projections des points de référence sur la caméra gauche (les  $B_{g_i}$ ), car chaque point sur l'image a deux coordonnées,
- 20 pour les projections des points de référence sur la caméra droite (les  $B_{d_i}$ ),

- 11 pour la matrice de projection de la caméra gauche ( $P_g$ ),
- 11 pour la matrice de projection de la caméra droite ( $P_d$ ),
- 30 pour les points de référence fournis par la vision stéréoscopique (les  $M_i$  sont des points dans l'espace, alors chacun a trois coordonnées),
- 10 pour les cycles de fermeture (les  $K_i$ ) car il y en a un pour chaque point de référence,
- 6 pour la position de l'objet ( $PosObj$ ) dans l'espace, elle est 6-dimensionnelle car il y a trois coordonnées de translation et trois d'orientation,
- 6 pour la position de la table ( $PosTable$ ),
- 3 pour la position de prise ( $PosPrise$ ) de l'objet, car cette position est définie dans un plan de prise. Ainsi, elle a deux coordonnées de translation et une d'orientation,
- 3 pour la position d'approche à l'objet ( $PosApp$ ), similaire à  $PosPrise$ ,
- 3 pour la position d'éloignement avec l'objet ( $PosElop$ ),
- 3 pour la position de pose ( $PosPos$ ) de l'objet,
- 3 pour la position d'approche avec l'objet ( $PosAppp$ ),
- 3 pour la position d'éloignement ( $PosElo$ ) de la table,
- 6 pour la position articulaire ( $\Theta$ ) car, le bras robot a six articulations,
- 6 pour la position cartésienne ( $X$ ) du bras robot,
- 9 pour les capteurs. Elles sont indiquées par :  $S_{base}$ ,  $S_{app}$ ,  $S_{prise}$ ,  $S_{elop}$ ,  $S_{appp}$ ,  $S_{dep}$ ,  $S_{elo}$ ,  $S_{pince}$ ,  $S_{objprise}$ ,
- 4 pour les activités. Elles sont : l'activité actuelle ( $A$ ) et les activités dans les temps passés ( $A_{-1}$ ,  $A_{-2}$  et  $A_{-3}$ ),
- 6 pour la position but ( $But$ ),
- 1 pour l'action de la pince ( $MPince$ ),
- 1 pour la différence entre les vitesses ( $E$ ),
- 6 pour la vitesse articulaire antérieure ( $\dot{\Theta}_{-1}$ ) et,
- 6 pour la vitesse articulaire actuelle ( $\dot{\Theta}$ ).

Notons que la complexité du problème dépend aussi du domaine de chaque variable.

Comme nous le savons, la distribution de probabilités conjointe porte sur toutes les variables impliquées et il faut choisir sur cette distribution une décomposition représentant notre connaissance sur le problème de "pick and place".

### 5.3.3 Décomposition

Le processus de "pick and place" est un processus complexe impliquant les processus probabilistes cités dans la section 5.3.1. Le choix de la décomposition du "pick and place" est lié aux hypothèses d'indépendances existantes entre les processus impliqués. Nous citons les hypothèses :

- la calibration d'une caméra dépend des points de référence et de leurs projections ,
- la localisation d'un point dépend des calibrations des caméras du système de vision et de ses projections sur les caméras,

$$\begin{aligned}
& p \left( \begin{array}{c} Z_1 \dots Z_N \\ B_{g1} \dots B_{gN} \\ B_{d1} \dots B_{dN} \\ P_g \ P_d \\ M_1 \dots M_N \\ K_1 \dots K_N \\ PosObj \ PosTable \\ PosPrise \ PosApp \\ PosElop \ PosPos \\ PosAppp \ PosElo \\ \Theta \ X \\ S_{base} \\ S_{app} \ S_{prise} \ S_{elop} \\ S_{appp} \ S_{dep} \ S_{elo} \\ S_{pince} \ S_{objprise} \\ A_{-1} \ A_{-2} \ A_{-3} \ A \\ MPince \\ But \ E \ \dot{\Theta}_{-1} \ \dot{\Theta} \end{array} \right) = \left[ \begin{array}{l} \prod_{i=1}^N p(Z_i) \\ \prod_{i=1}^N p(B_{g_i}) \\ \prod_{i=1}^N p(B_{d_i}) \\ p \left( P_g \ \left| \begin{array}{c} Z_1 \dots Z_L \\ B_{g1} \dots B_{gL} \end{array} \right. \ c_{cam-g} \right) \\ p \left( P_d \ \left| \begin{array}{c} Z_1 \dots Z_L \\ B_{d1} \dots B_{dL} \end{array} \right. \ c_{cam-d} \right) \\ \prod_{i=1}^N p(M_i \mid P_g \ P_d \ B_{g_i} \ B_{d_i} \ c_{VS}) \\ \prod_{i=1}^N p(K_i) \\ p \left( PosObj \ \left| \begin{array}{c} K_1 \dots K_L \\ M_1 \dots M_L \end{array} \right. \ c_{loc-Obj} \right) \\ p \left( PosTable \ \left| \begin{array}{c} K_{L+1} \dots K_N \\ M_{L+1} \dots M_N \\ Z_{L+1} \dots Z_N \end{array} \right. \ c_{loc-Table} \right) \\ p(PosPrise \mid PosObj \ c_{prise}) \\ p(PosPos \mid PosTable \ PosPrise \ c_{placer-objet}) \\ p(PosApp \mid PosPrise) \\ p(PosElop \mid PosPrise) \\ p(PosAppp \mid PosPos) \\ p(PosElo \mid PosPos) \\ p(\Theta) \\ p(X \mid \Theta) \\ p(S_{base} \mid X) \\ p(S_{app} \mid PosApp \ X) \\ p(S_{prise} \mid PosPrise \ X) \\ p(S_{elop} \mid PosElop \ X) \\ p(S_{appp} \mid PosAppp \ X) \\ p(S_{dep} \mid PosPos \ X) \\ p(S_{elo} \mid PosElo \ X) \\ p(S_{pince}) \\ p(S_{objprise}) \\ p(A_{-1}) \ p(A_{-2}) \ p(A_{-3}) \\ p(A \mid A_{-1} \ A_{-2} \ A_{-3} \ S_{base} \ \dots \ S_{objprise} \ c_{activité}) \\ p(MPince \mid A) \\ p(But \mid A) \\ p(\dot{\Theta}_{-1}) \\ p(E) \\ p(\dot{\Theta} \mid \Theta \ \dot{\Theta}_{-1} \ But \ E \ c_{mov}) \end{array} \right) \end{aligned} \tag{5.6}$$

- la position d’un objet dépend de sa description géométrique en terme des points caractéristiques et du calcul de la position de ces points par le système de vision,
- la position de prise dépend de la position de l’objet,
- la position de pose dépend de la position de prise et de la position de la table,
- la position d’approche et la position de distance avec l’objet dépendent de la position de prise,
- la position d’approche avec l’objet et la position de distance dépendent de la position de pose,
- la position cartésienne du bras dépend de sa position articulaire,
- les capteurs abstraits de position dépendent de la position cartésienne du bras et de leurs positions de référence,
- l’activité actuelle dépend des activités passées et des capteurs abstraits,
- la commande de la pince dépend de l’activité actuelle,
- la position du but dépend de l’activité actuelle (voir section 5.2.3),
- la vitesse articulaire dépend de la position articulaire, de la dernière vitesse et de la position but,

La décomposition choisie est représentée par l’expression (5.6).

Nous décrivons dans la section suivante les formes paramétriques associées aux facteurs de la décomposition.

### 5.3.4 Formes paramétriques

Nous décrivons maintenant les formes paramétriques associées aux facteurs de la décomposition (voir aussi 5.3.1).

1. Le point de référence  $Z_i$  peut être placé n’importe où sur l’objet ou sur la table :

$$p(Z_i) = \textit{Uniforme}.$$

2. Les projections du point de référence  $Z_i$  sur les images gauche et droite suivent des distributions normales :

$$\begin{aligned} p(B_{g_i}) &= \textit{Normale}(b_{g_i}, \Sigma_{g_i}), \\ p(B_{d_i}) &= \textit{Normale}(b_{d_i}, \Sigma_{d_i}). \end{aligned}$$

où  $b_{g_i}$  et  $b_{d_i}$  représentent les coordonnées des projections sur les plans focaux des caméras gauche et droite où les  $\Sigma$ 's représentent les matrices de covariance associées à la précision de chacune des caméras (voir annexe B.1.9).

3. Les formes paramétriques pour les matrices de projection sont obtenues du processus d’inférence décrit dans la section B.1, par les questions :

$$p(P_g | Z_1 \dots Z_L B_{g_1} \dots B_{g_L} c_{cam\_g}),$$

$$p(P_d | Z_1 \dots Z_L B_{d_1} \dots B_{d_L} c_{cam\_d}).$$

Nous avons  $L$  points de référence en utilisant l'objet pour faire la calibration des caméras.

4. La position du point de référence  $M_i$  repérée par la vision stéréoscopique suit une distribution de probabilités définie par la question (voir section B.2.2) :

$$p(M_i | P_g P_d B_{g_i} B_{d_i} c_{VS}).$$

5. Nous n'avons aucun *a priori* sur la condition de fermeture de chaque cycle cinématique correspondant à chaque point de référence :

$$p(K_i) = \text{Uniforme}$$

6. La distribution de probabilités décrivant la position de l'objet est représentée avec la question (section 4.2) :

$$p(PosObj | K_1 \dots K_L M_1 \dots M_L Z_1 \dots Z_L c_{loc\_Obj}).$$

7. Le cas de la position de la table est similaire au cas de la position de l'objet :

$$p(PosTable | K_{L+1} \dots K_N M_{L+1} \dots M_N Z_{L+1} \dots Z_N c_{loc\_Tab}).$$

8. La distribution de probabilités pour la position de prise est donnée par la question :

$$p(PosPrise | PosObj c_{prise}).$$

9. La forme paramétrique pour la position de pose de l'objet sur la table est représentée de façon similaire à la description précédente :

$$p(PosPos | PosTable PosPrise c_{placer\_objet}).$$

10. La position d'approche à l'objet est déduite de la position de prise. La forme paramétrique associée est une distribution uniforme représentant sa plage de valeurs (voir section 5.2.1) :

$$p(PosApp | PosPrise) = \text{Uniforme}[PosPrise + AP - \Delta_{AP}, PosPrise + AP + \Delta_{AP}]. \quad (5.7)$$

11. La position d'éloignement avec l'objet est similaire à la position d'approche :

$$p(PosElop | PosPrise) = Uniforme[PosPrise + ELP - \Delta_{ELP}, PosPrise + ELP + \Delta_{ELP}].$$

12. La position d'approche à la table avec l'objet et la position d'éloignement de la table sont déduites, de la position de pose de l'objet sur la table, de façon similaire à  $PosApp$  :

$$p(PosAppp | PosPos) = Uniforme \left[ \begin{array}{l} PosPos + APPP - \Delta_{APPp}, \\ PosPos + APPP + \Delta_{APPp} \end{array} \right],$$

$$p(PosElo | PosPos) = Uniforme[PosPos + EL - \Delta_{EL}, PosPos + EL + \Delta_{EL}].$$

13. Nous n'exprimons pas de connaissance préalable sur la position articulaire du bras :

$$p(\Theta) = Uniforme.$$

14. La position cartésienne du bras est considérée sans incertitude (voir section 4.1.7) :

$$p(X | \Theta) = \delta_{(\theta)}(X).$$

15. La distribution de probabilités décrivant les capteurs de position s'écrit (voir 5.2.2) :

$$p(S_r | Pos_r X) = \begin{cases} 1 & \text{si } \|Pos_r - X\| < \epsilon_r, \\ 0 & \text{sinon.} \end{cases}$$

Nous utilisons  $r$  pour définir les divers capteurs, les positions qu'ils testent et l'erreur permise pour chaque position testée.

16. Étant donné que nous n'avons pas d'*a priori* sur la valeur du capteur physique de la pince :

$$p(S_{pince}) = Uniforme.$$

17. Puisque nous n'avons pas d'*a priori* sur le capteur (physique) indiquant que le bras porte l'objet :

$$p(S_{objprise}) = Uniforme.$$

18. Puisque nous n'avons pas d'*a priori* sur les activités dans les temps passés  $-1$ ,  $-2$  et  $-3$  :

$$p(A_{-1}) = Uniforme,$$

$$p(A_{-2}) = Uniforme,$$

$$p(A_{-3}) = Uniforme.$$

19. La distribution de probabilités représentant l'activité actuelle est décrite dans la section 5.1 par la question :

$$p(A | A_{-1} A_{-2} A_{-3} S_{base} \dots S_{objprise} c_{activité}). \quad (5.8)$$

20. L'action de la pince est un dirac décrit dans la section 5.2.4. Ainsi ;

$$p(MPince | A) = \delta_{(c)}(MPince). \quad (5.9)$$

21. Le but représente la position à laquelle doit se diriger le bras, il dépend de l'activité actuelle et nous le considérons sans incertitude (voir section 5.5) :

$$p(But | A) = \delta_{(A)}(But).$$

22. La dernière vitesse articulaire peut être n'importe laquelle dans la plage de vitesses autorisées pour les articulations (voir section 4.1.7) :

$$p(\dot{\Theta}_{-1}) = Uniforme.$$

23. L'erreur permise pour la vitesse cartésienne (section 4.1.7) est une distribution normale :

$$p(E) = Normale(0, \sigma_E).$$

24. La distribution de probabilités représentant la vitesse des articulations du bras est décrite (dans la section 4.1.4) par :

$$p(\dot{\Theta} | \Theta \dot{\Theta}_{-1} But E c_{mov}). \quad (5.10)$$

Maintenant, nous considérons l'ensemble des variables, la décomposition et l'ensemble des formes paramétriques comme représentant toute la connaissance que nous avons sur le problème. Dorénavant, cette connaissance sera notée  $c_{pick\_and\_place}$ . À partir de cette connaissance, nous allons inférer la vitesse des articulations du bras et la commande de la pince.

### 5.3.5 Commande par inférence probabiliste

Étant donnée la description du problème (la décomposition, et les formes paramétriques associées aux facteurs de la décomposition), il est possible d'inférer la distribution de probabilités représentant la vitesse des articulations du bras ( $\dot{\Theta}$ ) et la commande de la pince ( $MPince$ ) en connaissant : la description géométrique de l'objet et de la table (les  $Z_i$ ), les projections des points de référence sur les caméras du système de vision (les  $B_{g_i}$  et les  $B_{d_i}$ ), la position articulaire du bras ( $\Theta$ ), l'état de la pince ( $S_{pince}$ ), l'information indiquant si la pince porte l'objet ( $S_{objprise}$ ),



les dernières activités ( $A_{-1}$ ,  $A_{-2}$  et  $A_{-3}$ ), les conditions garantissant la cohérence du système (les  $K_i$  impliquant les contraintes géométriques des problèmes de localisation et  $E$  établissant l'erreur permise pour la vitesse de déplacement du bras dans l'espace de travail).

Cette distribution de probabilités est représentée par l'expression suivante :

$$p \left( \begin{array}{c|c} \dot{\Theta} MPince & \begin{array}{l} Z_1 \dots Z_N B_{g_1} \dots B_{g_N} B_{d_1} \dots B_{d_N} \\ K_1 = vrai \dots K_N = vrai \\ S_{pince} S_{objprise} \\ A_{-1} A_{-2} A_{-3} \\ E = 0 \Theta \dot{\Theta}_{-1} \end{array} \\ \hline & c_{pick\_and\_place} \end{array} \right) \quad (5.11)$$

Elle pose un problème d'inférence qui doit être résolu dans un temps court pour rester dans le cadre défini par les hypothèses sur le contrôle de mouvement du bras (section 4.1.5).

Cette expression nécessite plusieurs processus d'inférence, elle les intègre dans une inférence plus général en réduisant la conduite d'une tâche complexe à une seule question probabiliste. Cette façon de commander le bras est tout à fait originale. Il faut fournir l'information disponible et poser la question correspondante.

La section suivante présente la solution au problème posé et les choix que nous faisons pour arriver à cette solution.

### 5.3.6 Commande du système

La commande du bras consiste à calculer une solution pour le problème posé par l'expression (5.11) tous les centièmes de seconde. Cela implique une marginalisation, sur l'ensemble des variables intermédiaires, représentée par l'expression (5.12). Ce calcul se fait en quatre temps.

1. Notre premier problème est de trouver la position des points caractéristiques décrivant l'objet à saisir à partir de l'information du système de vision. La distribution de probabilités conjointe représentant ce problème et la décomposition correspondante sont construites en considérant la description de la localisation d'un point avec le système de vision (eqn. B.29) et la description correspondante à la calibration de la caméra (eqn. B.13). Elle s'écrit :

$$p \left( \begin{array}{c|c} P_g P_d & \begin{array}{l} Z_1 \dots Z_L \\ B_{g_1} \dots B_{g_n} \\ B_{d_1} \dots B_{d_n} \end{array} \\ \hline M_1 \dots M_n & \end{array} \right) = \left[ \begin{array}{c} p \left( P_g \left| \begin{array}{l} Z_1 \dots Z_L \\ B_{g_1} \dots B_{g_L} \end{array} \right. c_{cam-g} \right) \\ p \left( P_d \left| \begin{array}{l} Z_1 \dots Z_L \\ B_{d_1} \dots B_{d_L} \end{array} \right. c_{cam-d} \right) \\ \prod_{i=1}^L p \left( M_i \left| \begin{array}{l} P_g P_d \\ B_{g_i} B_{d_i} \end{array} \right. c_{VS} \right) \end{array} \right]$$

$$\begin{aligned}
& p \left( \begin{array}{c} \dot{\Theta} \text{ MPince} \\ \left( \begin{array}{l} Z_1 \dots Z_N \\ B_{g1} \dots B_{gN} \\ B_{d1} \dots B_{dN} \\ K_1 = \text{vrai} \\ \vdots \\ K_N = \text{vrai} \\ S_{pince} \ S_{objprise} \\ A_{-1} \ A_{-2} \ A_{-3} \\ E = 0 \\ \Theta \ \dot{\Theta}_{-1} \\ C_{pick\_and\_place} \end{array} \right) \end{array} \right) \\
& \propto \int_{\text{But}} \sum_A \left[ \begin{array}{c} p(\dot{\Theta} \mid \Theta \ \dot{\Theta}_{-1} \ \text{But} \ E = 0 \ c_{mov}) \\ \left[ \begin{array}{c} p(\text{But} \mid A) \\ p(\text{MPince} \mid A) \\ \left[ \begin{array}{c} p \left( A \mid \begin{array}{l} A_{-1} \ A_{-2} \ A_{-3} \\ S_{base} \dots S_{objprise} \\ C_{activité} \end{array} \right) \\ \int_X \left[ \begin{array}{c} p(X \mid \Theta) \\ p(S_{base} \mid X) \\ p(S_{prise} \mid X) \\ p(S_{app} \mid X) \\ p(S_{elop} \mid X) \\ p(S_{dep} \mid X) \\ p(S_{appp} \mid X) \\ p(S_{elo} \mid X) \end{array} \right] d_X \end{array} \right] \end{array} \right] d_{\text{But}} \end{array} \right] \quad (5.12)
\end{aligned}$$

Si nous voulons calculer les mêmes points de référence en considérant l'incertitude sur les matrices de projection, nous posons la question :

$$p(M_1 \dots M_n \mid Z_1 \dots Z_L B_{g_1} \dots B_{g_n} B_{d_1} \dots B_{d_n}).$$

Elle est calculée en marginalisant sur les variables manquante (les matrices de projection  $P_g$  et  $P_d$ ),

$$\begin{aligned} p \left( M_1 \dots M_n \mid \begin{array}{l} Z_1 \dots Z_L \\ B_{g_1} \dots B_{g_n} \\ B_{d_1} \dots B_{d_n} \end{array} \right) &\propto \int_{P_g} \int_{P_d} p(M_1 \dots M_n \mid P_g P_d B_{g_1} \dots B_{g_n} B_{d_1} \dots B_{d_n}) dP_d dP_g \\ &\propto \int_{P_g} \int_{P_d} \left[ \begin{array}{l} p \left( P_g \mid \begin{array}{l} Z_1 \dots Z_L \\ B_{g_1} \dots B_{g_n} \end{array} c_{cam-g} \right) \\ p \left( P_d \mid \begin{array}{l} Z_1 \dots Z_L \\ B_{d_1} \dots B_{d_n} \end{array} c_{cam-d} \right) \\ \prod_{i=1}^L p \left( M_i \mid \begin{array}{l} P_g P_d \\ B_{g_i} B_{d_i} \end{array} c_{VS} \right) \end{array} \right] dP_d dP_g \end{aligned}$$

Cette dernière expression représente la distribution de probabilités des points  $M_i$ .

2. Dans un deuxième temps, nous cherchons la position de l'objet à saisir à partir de la position des points qui le caractérisent :

$$p(PosObj \mid K_1 \dots K_L M_1 \dots M_L Z_1 \dots Z_L c_{loc-Obj}) = p(PosObj) p(M_1 \dots M_L) p(K_1 \dots K_L)$$

Cette expression correspond à l'expression (4.21). Les points de référence ( $Z_i$ ) sont considérés constants, en conséquence ils ne sont pas dans l'expression (4.21). Cela ne veut pas dire qu'ils ne participent pas au calcul.

En considérant l'information disponible dans le problème de "pick and place", l'inférence sur la position de l'objet est faite à partir de sa description (les points de référence) et leurs projections sur les caméras de système de vision. Ainsi, la question correspondante s'écrit,

$$p(PosObj \mid [K_1 = vrai] \dots [K_L = vrai] [Z_1] \dots [Z_L] c_{loc-Obj})$$

avec les  $K_i = vrai$  pour prendre en compte les contraintes géométriques de la localisation stéréoscopique.

Pour répondre à cette question, il faut marginaliser sur les variables intermédiaires. Ce sont : les points de référence calculés par le système de vision (les  $M_i$ ) et les matrices de calibration ( $P_g$  et  $P_d$ ) des caméras du système de vision. Cette marginalisation est représentée par l'expression (5.19).

L'expression (5.20) représente l'inférence de la position de la table à partir de la position des points qui la caractérisent, elle est similaire à l'inférence précédente.

3. Dans un troisième temps, nous déterminons la valeur de chacun des capteurs d'état à partir de la position du bras et des différentes positions de référence.

Pour chaque capteur d'état, ce calcul implique une marginalisation sur une ou plusieurs positions, ainsi que l'utilisation de questions intermédiaires permettant d'introduire ces positions.

Regardons, par exemple, le calcul de la valeur du capteur d'état  $S_{prise}$  à partir de la position du bras (voir expression (5.13)). Ce calcul est réalisé par l'intermédiaire de la question incluant la position de l'objet et celle de prise de l'objet, ainsi qu'en marginalisant sur ces deux variables.

Il en est de même pour le calcul de la valeur des autres capteurs d'état.

En effet :

- le calcul de la valeur du capteur d'état  $S_{app}$  à partir de la position du bras (expression (5.14)), utilise des questions intermédiaires sur la position de l'objet, la position de prise, la position d'approche et une marginalisation sur ces trois variables,
  - le calcul de la valeur du capteur d'état de distance avec l'objet (expression (5.15)) utilise des questions intermédiaires sur la position de l'objet, la position de prise, la position de distance avec l'objet et une marginalisation sur ces trois variables,
  - le calcul de la valeur du capteur d'état de pose de l'objet (expression (5.16)) utilise des questions intermédiaires sur la position de l'objet, la position de la table, la position de prise, la position de pose de l'objet sur la table et une marginalisation sur ces quatre variables,
  - le calcul de la valeur du capteur d'état d'approche avec l'objet (expression (5.17)) utilise des questions intermédiaires sur la position de l'objet, la position de la table, la position de prise, la position de pose sur la table, la position d'approche et une marginalisation sur ces cinq variables,
  - le calcul de la valeur du capteur d'état de position de distance (expression (5.18)) utilise des questions intermédiaires sur la position physique de l'objet, la position physique de la table, la position de prise de l'objet, la position physique de pose de l'objet et sur la table et la position physique de distance et une marginalisation sur ces cinq variables.
4. Finalement dans un quatrième temps, nous résolvons la question permettant de commander le système (expression (5.11)). Pour cela, nous déterminons tout d'abord la distribution de probabilités sur les activités élémentaires à partir de questions intermédiaires incluant les capteurs d'états et d'une sommation sur ces variables. Puis, nous procédons à une sommation sur les activités élémentaires pour calculer l'état de la pince et le but à atteindre. Enfin, cette position à atteindre est utilisée pour déterminer la vitesse des six articulations du bras.

En pratique ce calcul doit être fait rapidement (environ tous les centièmes de seconde). C'est la principale difficulté de l'approche. Pour conduire les calculs dans le temps désiré, nous sommes amenés à faire des approximations sur l'estimation des sommes.

### 5.3.6.1 Résolution d'intégrales

Pour l'estimation des intégrales, nous avons choisi trois types d'approximation :

1. L'une consiste à approcher  $\int f(x)dx$  par :  $\max_x f(x)$ .
2. L'autre consiste à approcher  $\int f(x)dx$  par une méthode de Monte-Carlo.
3. Nous considérons un troisième cas dans lequel les deux cas précédents sont mélangés.

Supposons une expression de la forme :

$$\int_m f(m) \int_y g(y) h(y, m) d_y d_m \quad (5.21)$$

où, l'intégrale intérieure peut être estimée par la méthode de Monte-Carlo ainsi :

$$\int_y g(y) h(y, m) d_y \approx \frac{1}{N} \sum_{i=1}^N h(y_i^*, m), \quad (5.22)$$

avec  $y_i^*$  généré par des tirages aléatoires en suivant la fonction  $g(y)$ .

Nous avons considéré une approximation plus grossière de cette intégrale en ne prenant qu'une valeur pour  $y_i^*$  ( $N = 1$ ). Nous choisissons la valeur de  $y$  maximisant,

$$\max_y g(y) = g(\hat{y}). \quad (5.23)$$

L'approximation de l'intégrale devient ;

$$\int_y g(y) h(y, m) d_y \approx h(\hat{y}, m). \quad (5.24)$$

En considérant cette approximation, nous pouvons réécrire l'expression (5.21) :

$$\int_m f(m) \int_y g(y) h(y, m) d_y d_m \approx \int_m f(m) h(\hat{y}, m) d_m. \quad (5.25)$$

$$p(S_{prise} | X) \propto \int_{PosPrise} \left[ \int_{PosObj} \left[ \begin{array}{l} p(S_{prise} | X PosPrise) \\ p(PosPrise | PosObj c_{prise}) \\ p \left( PosObj \mid \begin{array}{l} K_1 = vrai \dots K_L = vrai \\ Z_1 \dots Z_L c_{loc\_Obj} \end{array} \right) \end{array} \right] d_{PosObj} \right] d_{PosPrise} \quad (5.13)$$

$$p(S_{app} | X) \propto \int_{PosApp} \left[ \int_{PosPrise} \left[ \int_{PosObj} \left[ \begin{array}{l} p(S_{app} | X PosApp) \\ p(PosApp | PosPrise) \\ p(PosPrise | PosObj c_{prise}) \\ p \left( PosObj \mid \begin{array}{l} K_1 = vrai \\ \vdots \\ K_L = vrai \\ Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \right) \end{array} \right] d_{PosObj} \right] d_{PosPrise} \right] d_{PosApp} \quad (5.14)$$

$$p(S_{elop} | X) \propto \int_{PosElop} \left[ \int_{PosPrise} \left[ \int_{PosObj} \left[ \begin{array}{l} p(S_{elop} | X PosElop) \\ p(PosElop | PosPrise) \\ p(PosPrise | PosObj c_{prise}) \\ p \left( PosObj \mid \begin{array}{l} K_1 = vrai \\ \vdots \\ K_L = vrai \\ Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \right) \end{array} \right] d_{PosObj} \right] d_{PosPrise} \right] d_{PosElop} \quad (5.15)$$

$$\begin{aligned}
p(S_{dep} | X) &\propto \int_{PosPos} \int_{PosPrise} \int_{PosTable} \left[ \begin{array}{l} p(S_{dep} | X PosPos) \\ p(PosPos | PosPrise PosTable) \\ \int_{PosObj} \left[ \begin{array}{l} p(PosPrise | PosObj c_{prise}) \\ p \left( \begin{array}{l} PosObj \\ \left. \begin{array}{l} K_1 = vrai \\ \vdots \\ K_L = vrai \\ Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \right) \\ \end{array} \right] d_{PosObj} \\ p \left( \begin{array}{l} PosTable \\ \left. \begin{array}{l} K_{L+1} = vrai \\ \vdots \\ K_N = vrai \\ Z_{L+1} \dots Z_N \\ c_{loc\_Table} \end{array} \right) \end{array} \right) \end{array} \right] d_{PosTable} d_{PosPrise} d_{PosPos}
\end{aligned} \tag{5.16}$$

$$\begin{aligned}
p(S_{app} | X) &\propto \int_{PosApp} \int_{PosPos} \int_{PosPrise} \int_{PosTable} \left[ \begin{array}{l} p(S_{app} | X PosApp) \\ p(PosApp | PosPos) \\ p(PosPos | PosPrise PosTable) \\ \int_{PosObj} \left[ \begin{array}{l} p(PosPrise | PosObj c_{prise}) \\ p \left( \begin{array}{l} PosObj \\ \left. \begin{array}{l} K_1 = vrai \\ \vdots \\ K_L = vrai \\ Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \right) \\ \end{array} \right] d_{PosObj} \\ p \left( \begin{array}{l} PosTable \\ \left. \begin{array}{l} K_{L+1} = vrai \\ \vdots \\ K_N = vrai \\ Z_{L+1} \dots Z_N \\ c_{loc\_Table} \end{array} \right) \end{array} \right) \end{array} \right] d_{PosTable} d_{PosPrise} d_{PosPos} d_{PosApp}
\end{aligned} \tag{5.17}$$

$$\begin{aligned}
p(S_{elo} | X) \propto & \int_{PosElo} \left[ \begin{array}{c} p(S_{elo} | X, PosElo) \\ p(PosElo | PosPos) \end{array} \right] \\
& \int_{PosPos} \left[ \begin{array}{c} p(PosPos | PosPrise, PosTable) \\ p(PosPrise | PosObj, c_{prise}) \\ p(PosObj | \begin{array}{l} K_1 = vrai \\ \vdots \\ K_L = vrai \\ Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array}) \\ p(PosTable | \begin{array}{l} K_{L+1} = vrai \\ \vdots \\ K_N = vrai \\ Z_{L+1} \dots Z_N \\ c_{loc\_Table} \end{array}) \end{array} \right] d_{PosObj} \\
& d_{PosTable} d_{PosPrise} d_{PosPos} d_{PosElo} \quad (5.18)
\end{aligned}$$



$$p \left( \begin{array}{c|c} \text{PosObj} & \begin{array}{l} K_1 = \text{vrai} \\ \vdots \\ K_L = \text{vrai} \\ Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \end{array} \right) \propto \int_{M_1} \dots \int_{M_L} \left[ \begin{array}{c} p \left( \begin{array}{c|c} \text{PosObj} & \begin{array}{l} K_1 = \text{vrai} \dots K_L = \text{vrai} \\ M_1 \dots M_L Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \end{array} \right) \\ \int_{P_g} \int_{P_d} \left[ \begin{array}{c} p \left( \begin{array}{c|c} P_g & \begin{array}{l} Z_1 \dots Z_L \\ B_{g1} \dots B_{gL} \\ c_{cam\_g} \end{array} \end{array} \right) \\ p \left( \begin{array}{c|c} P_d & \begin{array}{l} Z_1 \dots Z_L \\ B_{d1} \dots B_{dL} \\ c_{cam\_d} \end{array} \end{array} \right) \\ \prod_{i=1}^L p \left( \begin{array}{c|c} M_i & \begin{array}{l} P_g P_d \\ B_{g_i} B_{d_i} \\ c_{VS} \end{array} \end{array} \right) \end{array} \right] d_{P_d} d_{P_g} \end{array} \right] d_{M_L} \dots d_{M_1} \quad (5.19)$$

$$p \left( \begin{array}{c|c} \text{PosTable} & \begin{array}{l} K_{L+1} = \text{vrai} \\ \vdots \\ K_N = \text{vrai} \\ Z_{L+1} \dots Z_N \\ c_{loc\_Table} \end{array} \end{array} \right) \propto \int_{M_{L+1}} \dots \int_{M_N} \left[ \begin{array}{c} p \left( \begin{array}{c|c} \text{PosTable} & \begin{array}{l} K_{L+1} = \text{vrai} \\ \vdots \\ K_N = \text{vrai} \\ M_{L+1} \dots M_N \\ Z_{L+1} \dots Z_N \\ c_{loc\_Table} \end{array} \end{array} \right) \\ \int_{P_g} \int_{P_d} \left[ \begin{array}{c} p \left( \begin{array}{c|c} P_g & \begin{array}{l} Z_{L+1} \dots Z_N \\ B_{gL+1} \dots B_{gN} \\ c_{cam\_g} \end{array} \end{array} \right) \\ p \left( \begin{array}{c|c} P_d & \begin{array}{l} Z_{L+1} \dots Z_N \\ B_{dL+1} \dots B_{dN} \\ c_{cam\_d} \end{array} \end{array} \right) \\ \prod_{i=L+1}^N p \left( \begin{array}{c|c} M_i & \begin{array}{l} P_g P_d \\ B_{g_i} B_{d_i} \\ c_{VS} \end{array} \end{array} \right) \end{array} \right] d_{P_d} d_{P_g} \end{array} \right] d_{M_N} \dots d_{M_{L+1}} \quad (5.20)$$

En estimant la solution de cette intégrale par la méthode de Monte-Carlo :

$$\int_m f(m) h(\hat{y}, m) d_m \approx \frac{1}{M} \sum_{i=1}^M f(m_i^*), \quad (5.26)$$

où,  $m_i^*$  est généré par des tirages aléatoires en suivant la distribution  $h(\hat{y}, m)$ .

L'expression (5.26) correspond à une estimation de l'intégrale (5.21) où l'intégrale intérieure a été estimée par (5.23). Avec cette estimation, l'intégrale extérieure est approchée par la méthode de Monte-Carlo.

### 5.3.6.2 Solution

Étant donnée l'expression (5.12), le processus de marginalisation génère trop d'opérations à cause des intégrales imbriquées (chaque variable intermédiaire doit être intégrée pour la sortir de la décomposition). Pour arriver à gérer le problème, pour chacune des 4 parties du calcul, nous sommes amenés à simplifier les intégrales :

**Notation pour indiquer la méthode d'obtention d'une valeur pour une variable :**

– **maximisation** :  $\hat{y}$

– **tirage aléatoire** :  $y^*$

1. dans la première partie nous présentons l'estimation de la distribution de probabilité de la position physique de l'objet. L'expression (5.19) est similaire à l'expression (5.21) avec :

$$\begin{aligned} f(m) &= p \left( \text{PosObj} \left| \begin{array}{l} K_1 = \text{vrai} \dots K_L = \text{vrai} \\ M_1 \dots M_L Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \right. \right), \\ g(y) &= p \left( P_g \left| \begin{array}{l} Z_1 \dots Z_L \\ B_{g1} \dots B_{gL} \end{array} \right. c_{cam\_g} \right) p \left( P_d \left| \begin{array}{l} Z_1 \dots Z_L \\ B_{d1} \dots B_{dL} \end{array} \right. c_{cam\_d} \right), \\ h(y, m) &= \prod_{i=1}^L p \left( M_i \left| \begin{array}{l} P_g P_d \\ B_{gi} B_{di} \end{array} \right. c_{VS} \right). \end{aligned}$$

L'intégrale intérieure :

$$\int_{P_g} \int_{P_d} \left[ \begin{array}{l} p \left( P_g \left| \begin{array}{l} Z_1 \dots Z_L \\ B_{g1} \dots B_{gL} \end{array} \right. c_{cam\_g} \right) \\ p \left( P_d \left| \begin{array}{l} Z_1 \dots Z_L \\ B_{d1} \dots B_{dL} \end{array} \right. c_{cam\_d} \right) \\ \prod_{i=1}^L p \left( M_i \left| \begin{array}{l} P_g P_d \\ B_{gi} B_{di} \end{array} \right. c_{VS} \right) \end{array} \right] d_{P_d} d_{P_g} \quad (5.27)$$

peut être estimée par une méthode de Monte-Carlo (eqn. 5.22), c'est-à-dire que l'approximation est représentée par :

$$\frac{1}{N} \sum_{j=1}^N \prod_{i=1}^L p(M_i | P_{g_j}^* P_{d_j}^* B_{g_i} B_{d_i})$$

avec,  $P_{g_j}^*$  et  $P_{d_j}^*$  générées par des tirages aléatoires en suivant les distributions :

$$\begin{aligned} p(P_g | Z_1 \dots Z_L B_{g_1} \dots B_{g_L} c_{cam\_g}), \\ p(P_d | Z_1 \dots Z_L B_{d_1} \dots B_{d_L} c_{cam\_d}). \end{aligned}$$

Étant donné que :

- la solution analytique de l'intégrale est trop complexe ou infaisable,
- l'approximation par la méthode de Monte-Carlo est un processus lourd,
- la distribution de probabilités est uni-modale,

Nous avons pris la décision de l'estimer en ne prenant qu'une seule valeur,  $N = 1$  (cas représenté par l'expression 5.23). C'est-à-dire que l'on considère les valeurs maximisant les distributions,

$$\begin{aligned} \max_{P_g} p(P_g | Z_1 \dots Z_L B_{g_1} \dots B_{g_L} c_{cam\_g}) &= p(\hat{P}_g | Z_1 \dots Z_L B_{g_1} \dots B_{g_L} c_{cam\_g}), \\ \max_{P_d} p(P_d | Z_1 \dots Z_L B_{d_1} \dots B_{d_L} c_{cam\_d}) &= p(\hat{P}_d | Z_1 \dots Z_L B_{d_1} \dots B_{d_L} c_{cam\_d}). \end{aligned}$$

Les matrices de projection  $\hat{P}_g$  et  $\hat{P}_d$  sont les matrices résultants du processus de maximisation (annexe B.1). Elles représentent les paramètres de calibration des caméras gauche et droite respectivement.

Étant données  $\hat{P}_g$  et  $\hat{P}_d$ , l'approximation de l'intégrale (5.27) peut se faire comme il est indiqué en (5.24) :

$$\int_{P_g} \int_{P_d} \left[ \begin{array}{c} p \left( P_g \left| \begin{array}{c} Z_1 \dots Z_L \\ B_{g_1} \dots B_{g_L} \end{array} c_{cam\_g} \right. \right) \\ p \left( P_d \left| \begin{array}{c} Z_1 \dots Z_L \\ B_{d_1} \dots B_{d_L} \end{array} c_{cam\_d} \right. \right) \\ \prod_{i=1}^L p \left( M_i \left| \begin{array}{c} P_g P_d \\ B_{g_i} B_{d_i} \end{array} c_{VS} \right. \right) \end{array} \right] dP_d dP_g \approx \prod_{i=1}^L p \left( M_i \left| \begin{array}{c} \hat{P}_g \hat{P}_d \\ B_{g_i} B_{d_i} \end{array} c_{VS} \right. \right) \quad (5.28)$$

Étant donnée cette estimation, nous réécrivons l'expression (5.19) :

$$\int_{M_1} \dots \int_{M_L} \left[ p \left( \begin{array}{c} PosObj \\ K_1 = vrai \dots K_L = vrai \\ M_1 \dots M_L Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \right) \right] d_{M_L} \dots d_{M_1},$$

$$\prod_{i=1}^L p \left( M_i \mid \begin{array}{c} \hat{P}_g \hat{P}_d \\ B_{g_i} B_{d_i} \\ c_{VS} \end{array} \right)$$

elle est similaire à l'expression (5.24) estimée par la méthode de Monte-Carlo (expression 5.26). De façon similaire, l'estimation de l'expression précédente a la forme :

$$p \left( \begin{array}{c} PosObj \\ K_1 = vrai \\ \vdots \\ K_L = vrai \\ Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \mid \begin{array}{c} K_1 = vrai \\ \vdots \\ K_L = vrai \\ M_{1j}^* \dots M_{Lj}^* \\ Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \right) \approx \frac{1}{N} \sum_{j=1}^N p \left( \begin{array}{c} PosObj \\ K_1 = vrai \\ \vdots \\ K_L = vrai \\ M_{1j}^* \dots M_{Lj}^* \\ Z_1 \dots Z_L \\ c_{loc\_Obj} \end{array} \right) \quad (5.29)$$

avec  $M_{ij}^*$  généré par des tirages aléatoires sur  $p \left( M_{ij} \mid \begin{array}{c} \hat{P}_g \hat{P}_d \\ B_{g_i} B_{d_i} \\ c_{VS} \end{array} \right)$ ,  
et  $i = 1 \dots L$ .

Les  $M_{ij}^*$  peuvent être générés indépendamment car il existe une hypothèse d'indépendance mutuelle étant données les matrices de projection des caméras de la paire stéréoscopique et les projections de points de référence sur les images des caméras (annexe B.2.2).

L'expression (5.29) représente la distribution de probabilités pour la position de l'objet.

2. Dans la deuxième partie, nous présentons l'estimation de la distribution de probabilité pour la position physique de prise de l'objet. Cela implique l'intégration sur la position physique de l'objet indiqué par l'expression :

$$\int_{PosObj} \left[ p(PosPrise \mid PosObj c_{prise}) \right]$$

$$p \left( \begin{array}{c} PosObj \\ K_1 = vrai \dots K_L = vrai \\ Z_1 \dots Z_L c_{loc\_Obj} \end{array} \right) d_{PosObj}$$

approchée comme l'expression (5.24), c'est-à-dire que nous prenons la valeur maximisant la probabilité car, il s'agit d'un problème de calibration impliquant une solution unique ;

$$\max_{PosObj} p \left( \begin{array}{c} PosObj \\ K_1 = vrai \\ \vdots \\ K_L = vrai \\ c_{loc\_Obj} \end{array} \right) = p \left( \begin{array}{c} \widehat{PosObj} \\ K_1 = vrai \\ \vdots \\ K_L = vrai \\ c_{loc\_Obj} \end{array} \right)$$

le processus est décrit dans la section 4.2, il implique l'intégration des  $M_i$  par la méthode de Monte-Carlo (voir équation 5.29) dans chaque pas de l'algorithme de maximisation (voir section 4.2).

En prenant l'approximation de la position physique de l'objet  $\widehat{PosObj}$ , la distribution de probabilités de la position physique de prise s'écrit,

$$p(PosPrise | \widehat{PosObj} c_{prise}) \quad (5.30)$$

3. La troisième partie est dédiée à l'estimation de la distribution de probabilité des capteurs d'états de position.

L'intégrale sur la position physique de prise (équation 5.13) est estimée de façon spéciale, car au lieu de prendre la valeur de la position de prise maximisant (5.30), nous prenons des valeurs de  $PosPrise^*$  générées par des tirages aléatoires avec (5.30). Ainsi, la distribution de probabilité représentant le capteur d'état de la position physique de prise indiquée,

$$p(S_{prise} | X PosPrise)$$

est estimée par :

$$p(S_{prise} | X PosPrise^*).$$

ou,  $PosPrise^*$  représente la valeur de la position physique de prise obtenue par des tirages aléatoires.

Étant donnée la position actuelle  $X$  du bras et la position physique de référence correspondante au capteur  $Pos_r^*$ , la distribution de probabilité de chaque capteur d'état en fonction de la distance entre les deux positions est :

$$p(S_r | X Pos_r) = \begin{cases} 1 & \text{si } \|X - Pos_r\| < \varepsilon_r, \\ 0 & \text{sinon.} \end{cases}$$

où  $r$  a été établi dans la description des capteurs. La distance  $\varepsilon_r$  établit l'erreur acceptée entre la position du bras et la position correspondante au capteur  $Pos_r$ .

Exemple ; l'expression pour la probabilité du capteur de prise étant donnée la position cartésienne  $X$  (dirac calculée par l'expression 4.1) du bras et la position liée au capteur de prise  $PosPrise^*$  (obtenue par des tirages aléatoires) est :

$$p(S_{prise} | X PosPrise^*).$$

Alors,  $p([S_{prise} = vrai] | [X = x] [PosPrise = PosPrise^*]) = 1$  si  $\|x - PosPrise^*\| < \varepsilon$ . Ce modèle est utilisé pour tous les capteurs d'états de position.

4. Dans un quatrième temps, nous choisissons la tâche actuelle au moyen d'un tirage aléatoire sur la distribution définie par la question :

$$p(A | A_{-1} A_{-2} A_{-3} S_{base} \dots S_{objprise} C_{activité}).$$

La sommation sur le *but* ne se fait pas car sa distribution de probabilités est très proche d'un dirac. Ainsi, la valeur du but est définie par l'activité, laquelle est choisie par des tirages aléatoires.

Enfin, l'inférence de la vitesse des moteurs du bras se fait en posant la question,

$$p(\dot{\Theta} | \Theta \dot{\Theta}_{-1} But E = 0 c_{mov}).$$

La solution prise est celle maximisant la distribution. Les détails de cette inférence se trouvent dans la section 4.1.

L'action physique de la pince dépend de l'action d'état.

### 5.3.7 Conclusion

Dans cette section, nous avons présenté l'architecture de contrôle permettant au système robotique de réaliser la tâche complexe de "pick and place". Cette architecture a pour but d'intégrer et de coordonner l'ensemble des programmes probabilistes décrits dans ce document. Nous avons expliqué comment les différentes descriptions et questions venaient s'intégrer dans l'architecture. Nous avons montré que la description intégratrice (correspondante à l'architecture) représentant le "pick and place" nous permet de poser le problème sur une seule question probabiliste. Cette question met en jeu le bras manipulateur, le système de vision stéréoscopique et le robot cartésien. Le but étant de commander le bras par inférence.

Résoudre ces questions revient à intégrer sur les variables intermédiaires. Nous avons présenté des alternatives pour l'approximation de ces intégrales, et justifié leur intérêt par rapport au but visé.

Pour illustrer le résultat de notre travail, nous présentons dans la section suivante une expérimentation sur le système robotique.

## 5.4 Expérimentation

Dans nos expérimentations, nous avons constaté un déplacement souple du robot suivant la séquence prévue. Quelque rares fois, le bras a effectué des petits mouvements aléatoires sans conséquence. Ces mouvements sont dus à la précision demandée sur la position de prise. Mais de façon générale, la prise d'objets se passe sans problème.

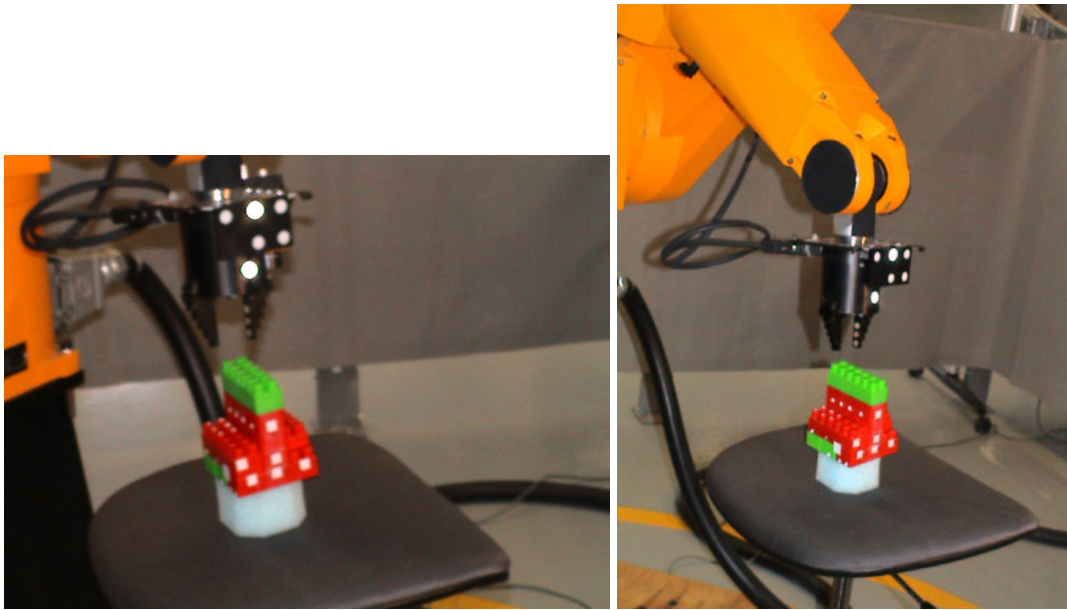


FIG. 5.3 – Bras en position d’approche

Une fois l’objet saisi, le robot se rend à une position sûre nommée la position d’éloignement avec l’objet. Cette position est proche de la position de prise pour garantir que le robot ne prend pas des trajectoires aléatoires induisant des risques de collision avec la table.

Nous présentons les principales étapes de la manipulation. Le “pick” démarre quand la position de l’objet est connue.

- la figure 5.3 montre le bras en position d’approche. La position d’approche n’est pas trop proche de l’objet de façon à garantir que le robot a une liberté de mouvement mais en même temps, elle est suffisamment proche de l’objet pour garantir que le bras se déplace en suivant une droite pour arriver à la position de prise de l’objet.
- la figure 5.4 montre une prise de l’objet (l’action correspondante est *saisir*) avec des points de vues différents.
- la figure 5.5 montre une position sûre nommée *position d’éloignement avec l’objet*.
- la partie gauche de la figure 5.6 montre le bras bougeant vers la position d’approche avec l’objet (action *approche avec l’objet*). Ce déplacement est libre de contraintes et se fait à la vitesse maximale autorisée.
- la partie droite de la figure 5.6 montre le bras en position d’approche avec l’objet. Il est prêt pour aller déposer l’objet.
- la partie gauche de la figure 5.7 montre le bras en mouvement vers la position de pose de l’objet (action *aller déposer*).
- la partie droite de la figure 5.7 illustre l’ouverture de la pince. Cette action se fait avec le robot immobile.
- finalement, le bras se dirige vers la position d’éloignement (action *s’écarter*), le mouve-



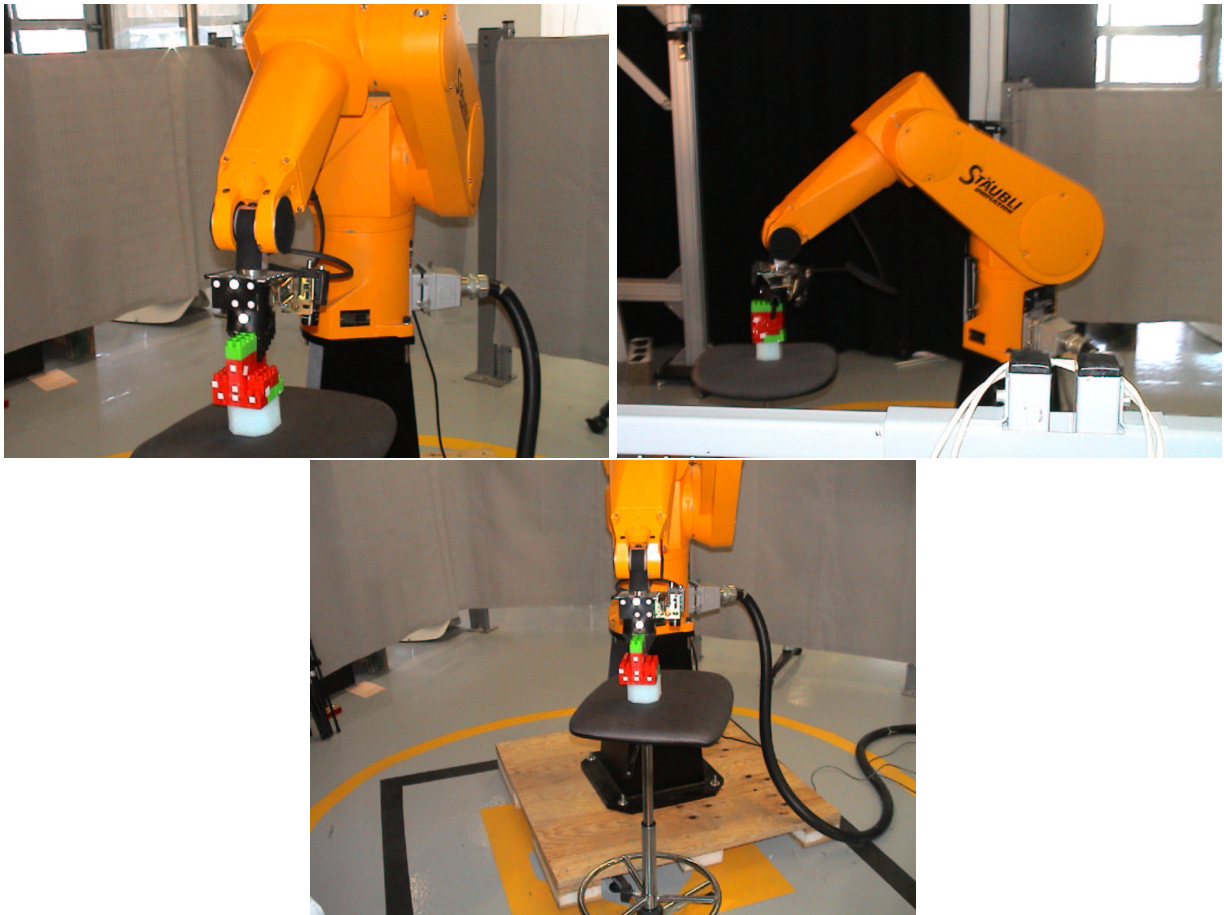


FIG. 5.4 – Bras en position de prise de l'objet

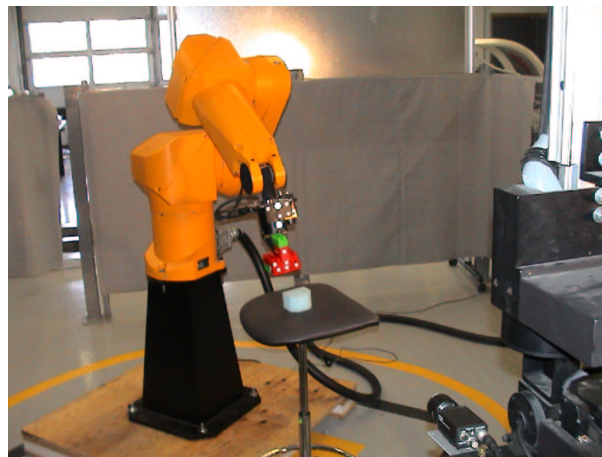


FIG. 5.5 – Bras en position de distance avec l'objet.



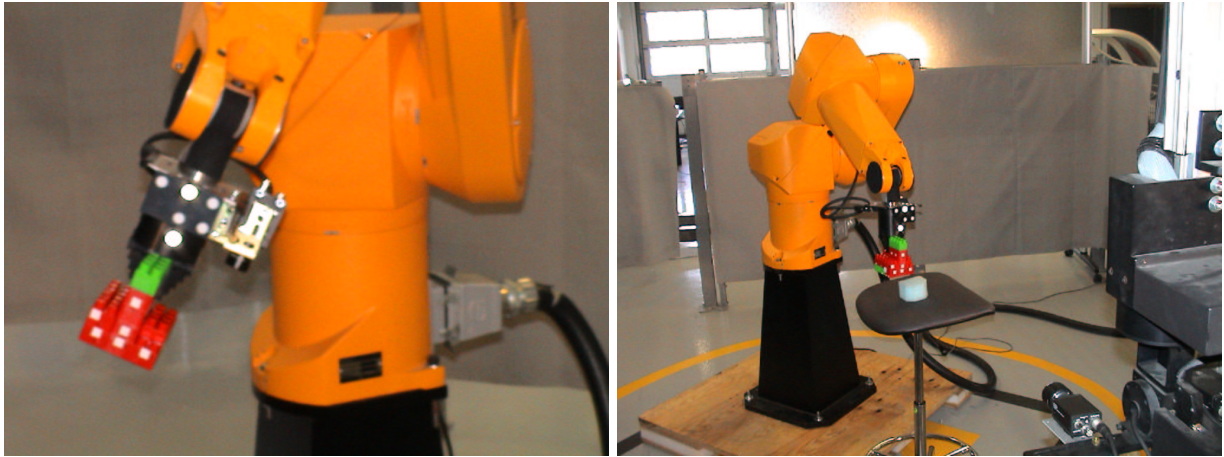


FIG. 5.6 – Bras vers et en position d’approche avec l’objet.



FIG. 5.7 – Place de l’objet.

ment du robot est lent car il peut faire tomber l’objet (figure 5.8).

- pour finir, nous montrons aussi le modèle numérique du bras robot (figure 5.9) utilisé pour mettre au point notre système.

## 5.5 Conclusion

Nous avons défini un programme probabiliste qui permet très facilement de définir comment les opérations élémentaires (telles qu’aller saisir l’objet ou s’écarter avec l’objet) s’enchaînent dans le “pick and place”. Pour cela, nous faisons appel à deux propriétés de la programmation bayésienne :

1. pour effectuer le “pick and place”, le bras manipulateur enchaîne des opérations élémentaires. Pour cela, les formes paramétriques modélisent des dépendances temporelles entre



FIG. 5.8 – Bras en position de distance.

les activités élémentaires, sur un horizon plus ou moins long. Ceci est possible par l'utilisation de variables dynamiques, associées aux actions élémentaires, qui jouent le rôle de mémoire. Le programme probabiliste construit avec ce type de dépendances temporelles est en fait un automate probabiliste, qui décrit la séquence d'opérations élémentaires à effectuer pour réaliser la tâche ;

2. chaque opération élémentaire abstraite est spécifiée en fonction des divers capteurs d'état et des dernières activités réalisées. Cette vision du problème est dite "inverse" car nous spécifions les lectures capteur auxquelles il faut s'attendre lorsqu'on réalise une activité élémentaire donnée. Ce mode de spécification est relativement peu usité en robotique. En effet, on a plutôt tendance à spécifier l'inverse : à savoir ce qu'il faut faire étant données les lectures capteur.

Dans la troisième section, nous avons détaillé les processus bayesiens nécessaires à l'interaction entre la tâche de "pick and place" et les autres programmes qui sont en contact direct avec le monde physique.

Enfin, nous avons présenté l'architecture permettant la réalisation de la tâche de "pick and place". Elle sert à intégrer et coordonner l'ensemble des processus probabilistes que nous avons présentés tout au long de ce document. Nous avons vu que la description intégratrice représentant le "pick and place" nous permet de poser le problème sur une seule question probabiliste. Nous avons présenté des alternatives pour l'approximation des intégrales : nous avons décrit trois méthodes d'intégration originales qui permettent en fonction des contraintes sur le temps de calcul et de la connaissance sur les distributions attendues de propager au mieux l'incertitude.

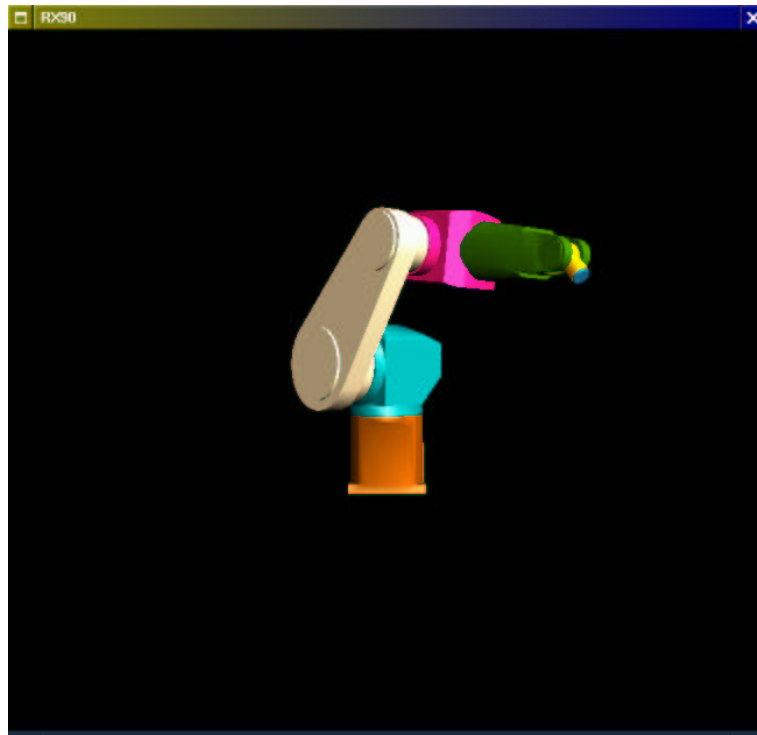


FIG. 5.9 – Simulation de mouvement du bras.

Nous voyons que les techniques mises en oeuvre pour réduire le temps calcul de l'expression (5.12) peuvent s'interpréter comme un choix d'architecture pour un contrôleur de robot. Nous pouvons aussi voir que cette architecture dépend de la spécification de l'interdépendance des processus. Dans notre cas, elle a été établie par la décomposition et les formes paramétriques. On a la possibilité de changer l'architecture en changeant de choix ou en changeant les distributions de probabilités. Notre méthode peut donc s'adapter aux caractéristiques de l'environnement et aux dispositifs de calcul.

# Chapitre 6

## Conclusion

Dans ce document, nous nous sommes intéressés à utiliser la programmation bayésienne pour réaliser une tâche de saisie et de déplacements d'objets avec un système robotique. Le système robotique utilisé est composé d'un bras manipulateur équipé d'une pince pour saisir et déplacer les objets et d'un robot cartésien équipé d'un système de vision stéréoscopique pour percevoir l'environnement. Cette tâche complexe consiste à saisir un objet avec le bras manipulateur et à le déposer à une position précise dans l'environnement : typiquement sur une table dont la position est connue.

Pour résoudre ce problème, nous avons utilisé la programmation bayésienne et ses principes. Nous rappelons dans la suite, nos apports :

- Nous avons tout d'abord défini un programme bayésien de planification de trajectoires pour le bras afin qu'il puisse se déplacer dans l'environnement d'une position à une autre.
- Ensuite, nous avons proposé une méthode probabiliste pour localiser un objet à partir d'un ensemble de points caractérisant cet objet dans le repère des caméras. Nous avons montré que les propriétés d'inversion et de fusion de l'approche bayésienne propose une solution naturelle à ce problème. D'autre part, pour ce problème, la solution la plus adéquate est celle maximisant la distribution de probabilité générée par la question. Pour effectuer cette recherche de maximum, nous avons adopté les algorithmes génétiques.
- Enfin, nous nous sommes servis de cette méthode pour déterminer la position de prise d'un objet à partir de la position de l'objet, c'est-à-dire la position à laquelle doit se trouver la pince pour saisir cet objet. Ce programme utilise une forme paramétrique originale : la mixture de distributions normales univariées.
- Puis, nous avons montré comment utiliser la programmation probabiliste pour décrire la séquence d'actions élémentaires composant une tâche complexe et abstraite (dans notre cas, le "pick and place"), ainsi que la manière dont ces actions s'enchaînent. Ce programme repose sur l'utilisation de variables dynamiques associées aux actions élémentaires. Elles jouent le rôle de mémoire, et sont utilisées pour modéliser des dépendances temporelles. Ceci montre qu'on peut aussi décrire facilement un comportement abstrait complexe grâce à la programmation probabiliste. De plus, nous pouvons tout à fait tenir compte de l'incertitude qu'il existe lors de l'enchaînement des actions élémentaires.
- Par la suite, nous avons détaillé les processus bayésiens nécessaires à l'interaction entre la

description de la tâche de “pick and place” et les autres programmes qui sont en contact direct avec le monde physique.

- Pour finir, nous avons conçu et développé une architecture de contrôle, pour le système robotique utilisé, dont le but est d’intégrer et de coordonner les programmes bayesiens composant le “pick and place”. Nous avons montré que cette intégration est possible et aisée. De plus, elle permet de poser le problème sur une seule question probabiliste. Cette question met en jeu le bras manipulateur, le système de vision stéréoscopique et le robot cartésien. Le but étant de commander le bras en temps réel par inférence.

La réponse à cette question implique les problèmes d’intégration sur les variables intermédiaires toujours lié à l’inférence probabiliste. Nous avons présenté des alternatives pour l’approximation de ces intégrales, et justifié leur intérêt par rapport au but visé.

De plus, cette architecture nous a permis de montrer la modularité de l’approche : les opérations de calibration, de localisation, de saisie et d’approche sont décrites séparément. Toutefois les incertitudes liées à chacune de ces opérations se transmettent naturellement aux autres. Cette caractéristique unique du formalisme bayésien permet de ne pas cloisonner artificiellement les calculs. Par exemple, une imprécision dans la localisation se combine à une imprécision dans le modèle de saisie.

Pourtant, notre approche reste compatible avec la notion de modularité qui consiste à diviser un problème en sous-problèmes. Toutefois, dans notre cas, cette décomposition ne repose pas sur une indépendance forte entre les niveaux. La dépendance maintenue permet de propager l’incertitude entre les différents niveaux d’analyse du problème. Cette propagation peut être totalement ignorée en effectuant une décision (typiquement en remplaçant une marginalisation par la valeur maximum) sur les variables intermédiaires, ou partiellement ignorée en effectuant une estimation des intégrations sur les variables intermédiaires. Nous avons introduit une conception originale de la notion d’architecture. Un choix d’architecture revient à un choix de méthodes d’intégration : c’est le fruit d’un compromis entre une résolution exhaustive du problème et la puissance de calcul.

En programmant une application complexe de manipulations d’objets par un bras manipulateur nous avons montré la généralité de l’approche bayésienne. Par la même, nous avons apporté une solution originale à ce problème en utilisant la programmation bayésienne et ses différentes propriétés. La programmation bayésienne de robots nous paraît donc une alternative crédible aux méthodes standards de programmation.

De plus, le résultat obtenu est très satisfaisant. En effet, nous avons réalisé une implantation en vraie grandeur et aussi plusieurs opérations de manipulation à l’aide de ce système. Les mouvements du robot sont souples et le système s’adapte sans problème au changement de contexte.

Notre méthode présente l’avantage de prendre explicitement en compte toutes les incertitudes liées à cette opération : incertitudes sur la géométrie des objets, précision des caméras, précision du robot et de sa commande. En plus, elle propose un mécanisme original de propagation de l’incertitude. Cette propagation peut être complète, partielle ou inhibé.

Dans cette thèse nous franchissons une étape pour la programmation bayésienne de robots en faisant passer le nombre de variables utilisées de quelques dizaines à plusieurs centaines, tout en mettant en oeuvre des méthodes d'intégration originales permettant un contrôle en temps réel d'un bras manipulateur.



# Annexe A

## Bibliographie

Nous présentons, dans cette annexe, quelques travaux liés à l'approche probabiliste, la prise en compte de l'incertitude dans la programmation de robots, l'incertitude géométrique et l'apprentissage.

### A.1 Travaux de Puget

Puget a travaillé sur la programmation automatique de robots manipulateurs [Pug89]. Il propose une modélisation d'actions et de l'environnement prenant en compte les incertitudes de positionnement. Pour planifier une opération d'assemblage, il considère comme indépendants les problèmes de saisie et de transport d'objets.

#### A.1.1 Graphe d'état

La représentation de l'environnement est faite au moyen d'un graphe orienté appelé *graphe d'état* où :

- un nœud représente un repère géométrique associé à la position d'un objet,
- un arc du repère  $R_i$  vers le repère  $R_j$  représente la position de l'objet  $j$  par rapport à l'objet  $i$ .

La matrice homogène de transformation  $T_{ij}$  indique la relation géométrique entre les repères  $j$  et  $i$ . La relation entre les repères  $i$  et  $j$  se fait en inversant la matrice homogène  $T_{ij}$ . Alors,

$$T_{ij} = T_{ji}^{-1}$$

#### A.1.2 Incertitude de position

L'incertitude sur la position d'un objet est associée à l'arc représentant sa relation géométrique. La valeur de l'erreur sur la position ne peut être qu'estimée. Elle est représentée par un vecteur de  $\mathbb{R}^6$  où :

- les trois premières composantes constituent l'erreur de translation,



- les trois dernières composantes constituent l’erreur de rotation (les trois composantes constituent un vecteur représentant l’axe de rotation, son module étant égal à l’angle de rotation).

Le passage de la représentation de l’erreur avec un vecteur :

$$\vec{\varepsilon} = \begin{pmatrix} \tau \\ \omega \end{pmatrix}$$

vers la représentation avec la matrice homogène

$$\varepsilon = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

se fait par les formules suivantes :

$$t = \tau$$

$$R = \cos(\|\omega\|)I_3 + (1 - \cos(\|\omega\|))\frac{\omega}{\|\omega\|} \left( \frac{\omega}{\|\omega\|} \right)^T - \sin(\|\omega\|)\{w\}$$

avec,

$$\{w\} = \begin{pmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{pmatrix}$$

Cette erreur  $\vec{\varepsilon}$  est considérée comme une variable aléatoire multidimensionnelle réelle, donnée par une distribution de probabilité normale sur  $\mathbb{R}^6$  avec une moyenne nulle et une matrice  $6 \times 6$  de covariance  $\Delta$  définie,

$$\forall i, j \in [1, 6], \Delta_{ij} = E(\varepsilon_i \varepsilon_j)$$

où,

- $E$  est l’espérance mathématique et,
- $\varepsilon_i$  est la *i*<sup>gravee</sup> composante de  $\vec{\varepsilon}$ .

### A.1.3 Problèmes causés par l’existence de cycles

Pour calculer la position et l’incertitude d’un repère par rapport à un deuxième repère, il existe un chemin entre les deux repères (dans le graphe d’état). Cependant, il est possible d’avoir plus d’un chemin entre les deux repères, c’est-à-dire qu’il existe des cycles dans le graphe d’état.

Puget a considéré l’existence de cycles comme une contradiction car, selon lui, la valeur de la position d’un repère par rapport à l’autre est différent en fonction du graphe d’état et du chemin choisi. Alors, pour éliminer les contradictions, Puget n’accepte pas de cycles dans le graphe d’état.

## A.1.4 Discussion

Les travaux de Puget se situent dans le cadre général de travaux sur la prise en compte d'incertitude en robotique de manipulation. Tout en se plaçant dans le cadre gaussien, son approche est fondatrice car il introduit explicitement la notion de graphe cinématique probabiliste.

Contrairement à l'approche de Mekhnacha, Puget considère que la présence d'un cycle dans le graphe d'état est représentatif d'une contradiction, il éliminera les arcs formant des cycles. Il ne saisit pas que les cycles sont des contraintes que l'on peut traiter par le raisonnement probabiliste en introduisant justement la notion d'erreur. Dans ce cas, le raisonnement probabiliste géométrique nous dit simplement que certaines erreurs sont possibles et pas d'autres.

## A.2 Travaux de De la Rosa

De la Rosa a travaillé sur la planification de mouvements pour des robots mobiles en présence d'incertitudes. En particulier, il s'intéresse aux incertitudes de positionnement et de déplacement.

Dans son travail, De la Rosa considère un robot ponctuel  $p_c$  naviguant dans l'espace libre d'obstacles  $C_{libre}$ . La position du robot est définie dans l'espace bidimensionnel par :

$$p_c = (x_c, y_c, \theta_c)$$

où,  $x_c$  et  $y_c$  représentent la position du robot et  $\theta_c$  représente son orientation.

L'incertitude sur la position et l'orientation du robot est représentée par  $\rho_c$  et  $\psi_c$  respectivement. Elle est calculée pour la position suivante  $p_{c'}$  ainsi :

$$\begin{aligned} \rho_{c'} &= \rho_c + \delta_\rho \\ \psi_{c'} &= \psi_c + \delta_\psi \end{aligned}$$

avec,

- $p_0$  représentant la position de départ,
- $p_f$  représentant la position du but,
- $p_c$  est la position actuelle,
- $p_{c'}$  est la position suivante,
- $\delta_\rho = \|\vec{\Delta}_{q_t}(p_c, p_f)\| * \tan(\eta)$  représentant l'incrément dans l'incertitude de position,
- $\vec{\Delta}_{q_t}(p_c, p_f)$  représentant le changement de position avec l'objectif d'arriver à la position but  $p_f$ ,
- $\eta$  est un angle bornant la déviation entre toutes les vitesses effectives  $v'$  et toutes les vitesses commandées  $v$  dans le trajet du  $p_0$  au  $p_f$ ; voir la figure A.1 avec la description :
  - a). Cône d'incertitude de contrôle en  $\mathbb{R}^2$  autour de la vitesse  $\vec{v}$ ,
  - b). Arc d'incertitude de position causé par la déviation angulaire de  $\vec{v}$  (l'amplitude  $\|\vec{v}\|$  du déplacement est parfaitement connue),

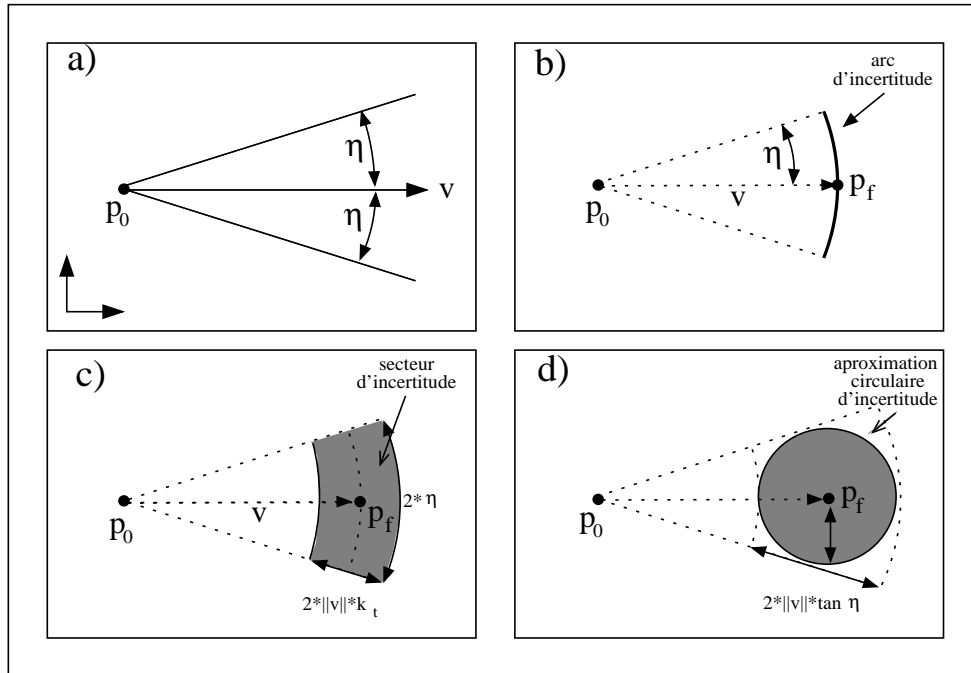


FIG. A.1 – L’incertitude de contrôle et son effet sur l’incertitude de position (tiré de [DLRR96])

- c). Secteur de couronne circulaire d’incertitude de position causé par les imprécisions sur la direction de  $\vec{v}$  et l’amplitude  $\|\vec{v}\|$  du déplacement,
- d). Approximation circulaire de l’incertitude de position due au suivi de  $\vec{v}$ .

On suppose que la déviation maximale  $\eta$  est constante. Alors, l’incrément dans l’incertitude de position est proportionnelle à la distance parcourue suivant la vitesse commandée,

- $\delta_\psi$  représente l’incrément d’incertitude d’orientation, elle augmente linéairement selon le changement de l’orientation  $\theta$  en fonction de la précision sur la rotation du robot, indiquée par un constante.

Au fur et à mesure que le robot bouge, l’incertitude s’agrandit. Pour la réduire, il est nécessaire d’avoir des contacts avec les obstacles.

L’approche de De la Rosa nous paraît intéressante car, il modélise l’incertitude sur le déplacement. Cette incertitude ne ferait que croître si des stratégies de recalage n’étaient mises en place pour combattre cette augmentation. Il s’agit donc d’élaborer des plans d’action permettant au robot de se recalier grâce à ses capteurs durant ses déplacements.

### A.3 Travaux de Ruiz

Ruiz travaille sur l’inférence probabiliste et l’apprentissage à partir d’information incertaine. Il utilise des mélanges de distributions de probabilités normales univariées pour construire les

modèles d'inférence. Les paramètres du mélange de distributions de probabilités sont estimés en utilisant une extension de l'algorithme *expectation maximization (EM)* [MK97].

Nous utilisons les mélanges de distributions normale univariées pour apprendre les positions de prises valables pour un objet en particulier et à partir du mélange, nous choisissons une position de prise de l'objet.

Dans les paragraphes suivants sont présentés les bases mathématiques et les principaux concepts de la méthode d'inférence probabiliste proposée par [RdTG98].

### A.3.1 Fondements

L'outil de base pour construire le mélange de distributions normales est la distribution normale univariée  $\mathcal{N}(\mu, \sigma)$ . Elle représente une densité de probabilité de la variable  $x$  avec les propriétés suivantes :

$$\begin{aligned}\mathcal{N}(\mu, \sigma) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} & \text{si } \sigma > 0 \\ \mathcal{N}(\mu, \sigma) &= \mathcal{N}(\mu, 0) \equiv \delta(x - \mu) & \text{si } \sigma = 0\end{aligned}\quad (\text{A.1})$$

si l'écart type est zéro, la densité de probabilité devient un Dirac. Dans les deux cas,  $\mathcal{N}(\mu, \sigma)$  est une distribution de densité de probabilités en conséquence :

$$\int_x \mathcal{N}(\mu, \sigma) dx = 1.$$

Le produit de deux distributions normales univariées sur la variable  $x$  peut être exprimé [Pap84, Ber85],

$$\mathcal{N}_x(\mu_1, \sigma_1) \cdot \mathcal{N}_x(\mu_2, \sigma_2) = \mathcal{N}_x(\eta, \varepsilon) \cdot \mathcal{N}_{\mu_1}(\mu_2, \sqrt{\sigma_1^2 + \sigma_2^2}) \quad (\text{A.2})$$

avec :

- $\mathcal{N}_x$  indiquant une distribution normale sur la variable  $x$ ,
- $\mathcal{N}_{\mu_1}$  indiquant une distribution normale sur la variable  $\mu_1$ ,
- quand  $\sigma_1 + \sigma_2 > 0$ , la moyenne  $\eta$  et l'écart type  $\varepsilon$  s'écrivent ;

$$\eta = \frac{\sigma_1^2 \mu_2 + \sigma_2^2 \mu_1}{\sigma_1^2 + \sigma_2^2} \quad \varepsilon^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

L'expression (A.2) est utile pour calculer l'intégrale du produit de deux distributions normales univariées sur  $x$  avec  $\sigma_1 + \sigma_2 > 0$ ,

$$\int_x \mathcal{N}(\mu_1, \sigma_1) \cdot \mathcal{N}(\mu_2, \sigma_2) dx = \mathcal{N}_{\mu_1} \left( \mu_2, \sqrt{\sigma_1^2 + \sigma_2^2} \right). \quad (\text{A.3})$$

Pour être consistant, on écrit l'expression pour les cas où  $\sigma_1 = \sigma_2 = 0$  ;

$$\int_x \mathcal{N}(\mu_1, 0) \cdot \mathcal{N}(\mu_2, 0) dx = \mathcal{N}_{\mu_1}(\mu_2, 0) \equiv I\{\mu_1 = \mu_2\}$$

avec  $I\{t\} = 1$  si  $t$  est vrai et  $I\{t\} = 0$  si  $t$  est faux.

### A.3.2 Famille mathématique

Supposons qu'un phénomène aléatoire  $Z$  soit déduit d'une certaine observation ou information subjective  $S$ . Si  $Z$  est décrit par  $p(Z)$  et l'observation est caractérisée par une distribution de probabilités conditionnelle  $p(S|Z)$ , alors la distribution de probabilité conjointe sur l'ensemble des variables impliquées est :

$$p(SZ) = p(S | Z)p(Z) = p(Z | S)p(S)$$

en marginalisant sur  $Z$ ,

$$\begin{aligned} p(Z | S) &= \frac{p(S | Z)p(Z)}{p(S)} \\ &= \frac{p(S | Z)p(Z)}{\int_Z p(S | Z)p(Z)d_z} \end{aligned} \quad (\text{A.4})$$

Le calcul de  $p(Z|S)$  peut être difficile, sauf si  $p(Z)$  et  $p(S|Z)$  sont des distributions de probabilité appartenant à la même *famille mathématique* [Ber85, BS94]. Par exemple, si elles sont des distributions de probabilité normales univariées :

$$p(Z) = \mathcal{N}_Z(r, \sigma_r) \quad (\text{A.5})$$

$$p(S | Z) = \mathcal{N}_S(Z, \sigma_s) \quad (\text{A.6})$$

De l'équation (A.4) avec les expressions (A.5) et (A.6),

$$p(Z | S) = \frac{\mathcal{N}_Z(S, \sigma_s) \cdot \mathcal{N}_Z(r, \sigma_r)}{\int_Z \mathcal{N}_Z(S, \sigma_s) \cdot \mathcal{N}_Z(r, \sigma_r)d_z}$$

en utilisant les expressions (A.2) et (A.3) et en simplifiant, :

$$\begin{aligned} p(Z | S) &= \frac{\mathcal{N}_Z(\eta, \varepsilon) \cdot \mathcal{N}_S\left(r, \sqrt{\sigma_r^2 + \sigma_s^2}\right)}{\mathcal{N}_S\left(r, \sqrt{\sigma_r^2 + \sigma_s^2}\right)} \\ &= \mathcal{N}_Z(\eta, \varepsilon) \end{aligned} \quad (\text{A.7})$$

avec,

$$\eta = \frac{\sigma_r^2 S + \sigma_s^2 r}{\sigma_r^2 + \sigma_s^2} \quad \varepsilon^2 = \frac{\sigma_r^2 \sigma_s^2}{\sigma_r^2 + \sigma_s^2}$$

### A.3.3 Approximation par mixture de distributions normales univariées

Dans la section précédente, nous avons introduit le phénomène aléatoire  $Z$ , nous nous intéressons maintenant aux attributs de  $Z$ . Nous allons considérer deux cas possibles pour  $Z$ . Dans le premier cas,  $Z$  n'a qu'un attribut (cas unidimensionnel), pour le deuxième cas,  $Z$  en a plusieurs. C'est-à-dire que  $Z$  est multidimensionnel.

### A.3.3.1 Cas unidimensionnel

Le phénomène aléatoire  $Z$  a un attribut, il peut être la taille d'une personne, la couleur d'un objet, en général n'importe quelle caractéristique.

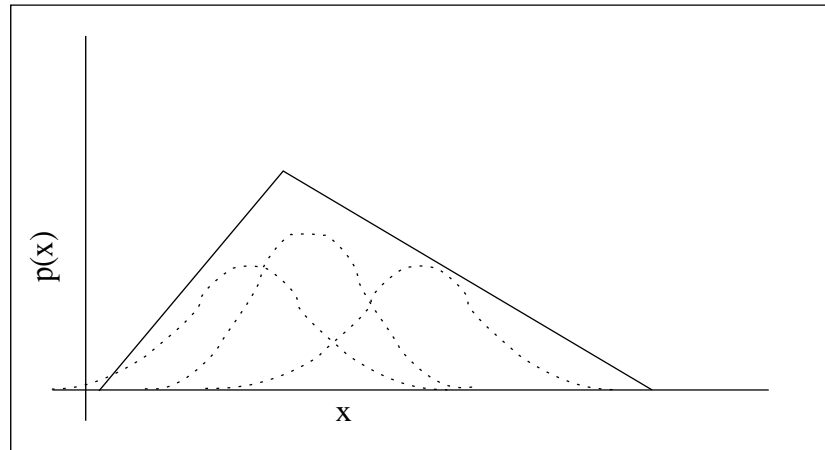


FIG. A.2 – La fonction de densité de probabilité d'un phénomène aléatoire approchée par une mixture de distributions normales univariées.

### Distribution de probabilité pour le phénomène aléatoire

Dans des situations réelles, les attributs du phénomène  $Z$  sont interdépendants entre eux. Dans ce cas, la fonction de densité de probabilité conjointe  $p(Z)$  devient complexe. En conséquence, la densité marginale pour l'inférence probabiliste peut être difficile à calculer. Cependant, on peut exprimer la densité de probabilité comme une mixture de  $m$  composantes factorisées  $C_1, C_2, \dots, C_m$  avec des paramètres indépendants. La figure A.2 montre la fonction de densité de probabilités, d'un phénomène aléatoire, approchée par la mixture de trois distributions de probabilités normales univariées,

$$p(Z) \cong \sum_{i=1}^m p\{C_i\} \mathcal{N}_Z(\mu_i, \sigma_i) \quad (\text{A.8})$$

où  $p\{C_i\}$  représente la probabilité de chaque composante du mélange, elle est aussi un facteur de normalisation.

### Modèle capteur

La distribution de probabilité  $p(S|Z)$  représentant le modèle d'un capteur fournissant une information sur  $Z$  peut aussi être approchée par un mélange de  $R$  distributions normales univariées :

$$p(S|Z) \cong \sum_{r=1}^R \pi_r \mathcal{N}_Z(\eta_r, \epsilon_r), \quad (\text{A.9})$$

les paramètres  $\eta_r$ ,  $\varepsilon_r$  et  $\pi_r$  dépendent explicitement de l'observation  $S$  actuelle.

### Inférence probabiliste

En utilisant l'expression (A.8) pour le phénomène aléatoire et l'expression (A.9) de l'observation, il est possible de réécrire l'expression (A.4) :

$$\begin{aligned} p(Z | S) &\cong \frac{\sum_{i=1}^m p\{C_i\} \mathcal{N}_Z(\mu_i, \sigma_i) \sum_{r=1}^R \pi_r \mathcal{N}_Z(\eta_r, \varepsilon_r)}{\int_Z \sum_{i=1}^m p\{C_i\} \mathcal{N}_Z(\mu_i, \sigma_i) \sum_{r=1}^R \pi_r \mathcal{N}_Z(\eta_r, \varepsilon_r) dZ} \\ &\cong \frac{\sum_{i=1}^m \sum_{r=1}^R p\{C_i\} \pi_r \mathcal{N}_Z(\mu_i, \sigma_i) \mathcal{N}_Z(\eta_r, \varepsilon_r)}{\sum_{i=1}^m \sum_{r=1}^R p\{C_i\} \pi_r \int_Z \mathcal{N}_Z(\mu_i, \sigma_i) \mathcal{N}_Z(\eta_r, \varepsilon_r) dZ} \end{aligned}$$

En utilisant les propriétés (A.2) et (A.3) nous obtenons l'expression pour faire l'inférence probabiliste :

$$p(Z | S) \cong \sum_{i=1}^m \sum_{r=1}^R \Theta_{i,r} \mathcal{N}_Z(\mathcal{V}_{i,r}, \lambda_{i,r}) \quad (\text{A.10})$$

avec,

$$\begin{aligned} \mathcal{V}_{i,r} &= \frac{\sigma_i^2 \eta_r + \varepsilon_r^2 \mu_i}{\sigma_i^2 + \varepsilon_r^2} & \lambda_{i,r} &= \frac{\sigma_i \varepsilon_r}{\sqrt{\sigma_i^2 + \varepsilon_r^2}} \\ \Theta_{i,r} &= \frac{p\{C_i\} \pi_r \mathcal{N}_{\mu_i} \left( \eta_r, \sqrt{\sigma_i^2 + \varepsilon_r^2} \right)}{\sum_{k=1}^m \sum_{l=1}^R p\{C_k\} \pi_l \mathcal{N}_{\mu_k} \left( \eta_l, \sqrt{\sigma_k^2 + \varepsilon_l^2} \right)} \end{aligned}$$

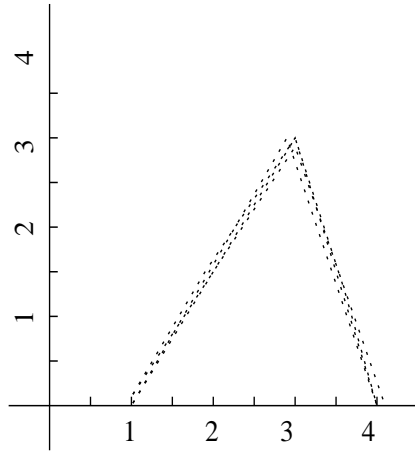
#### A.3.3.2 Cas pluridimensionnel

Le cas où le phénomène aléatoire  $Z$  a plusieurs attributs est le cas que l'on rencontre le plus souvent. Nous allons déduire la mixture de distributions de probabilités normales s'approchant le plus de la fonction de densité de probabilité conjointe du phénomène.

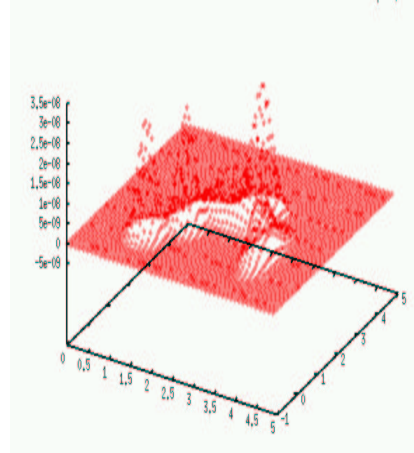
#### Distribution de probabilité pour le phénomène aléatoire

Considérons maintenant un phénomène aléatoire  $Z = (z^1, \dots, z^n)$  ayant  $n$  attributs. Nous allons approcher sa distribution de probabilité  $p(Z)$  avec une mixture de  $m$  distributions normales univariées en prenant l'hypothèse d'indépendance entre les attributs de  $Z$  :

$$\begin{aligned} p(Z) &= f(z^1, \dots, z^n) \\ &\cong \sum_{i=1}^m p\{C_i\} p(Z | C_i) \\ &\cong \sum_{i=1}^m p\{C_i\} \prod_{j=1}^n p(z^j | C_i) \end{aligned} \quad (\text{A.11})$$



a). Phénomène aléatoire



b). Mixture de Distributions normales

FIG. A.3 – Phénomène aléatoire avec deux attributs et mixture de distributions normales univariées pour approcher la distribution de densité de probabilités du phénomène.

Étant donné que la brique de base que nous utilisons est la distribution normale univariée, nous supposons que l'attribut  $j$  de  $Z$  est représenté dans la composante  $i$  de la mixture de distributions normales univariées par :

$$p(z^j | C_i) = \mathcal{N}_{z^j}(\mu_i^j, \sigma_i^j)$$

La mixture de distributions normales univariées approchant la densité de probabilité de  $Z$  s'écrit :

$$p(Z) \cong \sum_{i=1}^m p\{C_i\} \prod_{j=1}^n \mathcal{N}_{z^j}(\mu_i^j, \sigma_i^j). \quad (\text{A.12})$$

La figure A.3a représente un phénomène aléatoire avec deux attributs ; la figure A.3b représente le mélange de distributions normales univariées avec cinq composantes approchant la distribution de probabilité du phénomène, la description du mélange est présentée dans le tableau (A.1).

### Modèle capteur

Puisque dans le cas général  $Z$  a plusieurs attributs, il est possible d'avoir un ou plusieurs capteurs qui peuvent donner des observations sur un ou plusieurs attributs de  $Z$ . En représentant les modèles de ces capteurs avec des mixtures de distributions normales univariées une observation  $S = (s^1, s^2, \dots, s^n)$  de  $Z$  avec  $R$  capteurs s'écrit :

$$p(S | Z) \cong \sum_{r=1}^R p\{S | s_r\} p(s_r | Z)$$



composante		Attribut 1		Attribut 2	
$i$	$p\{C_i\}$	$\mu_i^1$	$\sigma_i^1$	$\mu_i^2$	$\sigma_i^2$
1	0.192464	1.24111	0.168244	0.361667	0.252365
2	0.175715	1.79161	0.159620	1.187410	0.239431
3	0.163943	3.70211	0.142376	0.893658	0.427128
4	0.232731	2.61791	0.501645	1.946710	0.222185
5	0.235147	2.89229	0.209548	2.645900	0.223255

TAB. A.1 – Mixture de normales généralisées avec 2 attributs et 5 composantes.

$$\cong \sum_{r=1}^R p\{S | s_r\} \prod_{j=1}^n p(s_r^j | z^j). \quad (\text{A.13})$$

Si  $p(s_r^j | z^j)$  est une distribution normale univariée ;

$$p(s_r^j | z^j) = \mathcal{N}_{z^j}(s_r^j, \epsilon_r^j),$$

représente l'observation de la caractéristique  $z^j$  fournie par le capteur  $s_r$ . Le modèle capteur s'écrit :

$$p(S | Z) = \sum_{r=1}^R \pi_r \prod_{j=1}^n \mathcal{N}_{z^j}(s_r^j, \epsilon_r^j) \quad (\text{A.14})$$

où  $R$  est le nombre de capteurs et  $n$  le nombre de distributions normales univariées utilisées pour approcher l'information fournie par les capteurs.

### Fonction de densité de probabilité conjointe

L'expression pour la fonction de densité de probabilité conjointe s'écrit :

$$\begin{aligned} p(C_i, Z, s_r, S) &= p\{S | s_r\} p(s_r | Z) p(Z | C_i) p\{C_i\} \\ &= p\{C_i\} p\{S | s_r\} \prod_{j=1}^n p(s_r^j | z^j) p(z^j | C_i) \\ &= p\{C_i\} \pi_r \prod_{j=1}^n \mathcal{N}_{z^j}(s_r^j, \epsilon_r^j) \mathcal{N}_{z^j}(\mu_i^j, \sigma_i^j) \end{aligned} \quad (\text{A.15})$$

Il est convenable de dériver une deuxième expression de l'équation (A.15), pour faciliter le calcul des densités marginales de n'importe quelle variable désirée, ainsi :

$$\begin{aligned} p(s_r^j | z^j) p(z^j | C_i) &= p(s_r^j z^j | C_i) \\ &= p(z^j | s_r^j C_i) p(s_r^j | C_i), \end{aligned}$$

en appliquant les propriétés (A.2) et (A.3) nous avons :

$$\begin{aligned}
\beta_{i,r}^j &= p(s_r^j | C_i) \\
&= \int_{z^j} p(s_r^j | z^j) p(z^j | C_i) dz^j \\
&= \mathcal{N}_{s_r^j}(\mu_i^j, \rho_{i,r}^j)
\end{aligned} \tag{A.16}$$

avec,

$$\rho_{i,r}^j = \sqrt{(\sigma_i^j)^2 + (\varepsilon_r^j)^2}.$$

L'expression  $\beta_{i,r}^j$  représente la probabilité de la  $r^{\text{ième}}$  observation  $s_r^j$  du  $j^{\text{ième}}$  attribut  $z^j$  dans chaque composante  $C_i$ .

En appliquant les propriétés (A.2) et (A.3) on déduit,

$$\begin{aligned}
\Psi_{i,r}^j(z^j) &= p(z^j | s_r^j C_i) \\
&= \frac{p(s_r^j | z^j) p(z^j | C_i)}{p(s_r^j | C_i)} \\
&= \mathcal{N}_{z^j}(\gamma_{i,r}^j, \lambda_{i,r}^j)
\end{aligned} \tag{A.17}$$

avec,

$$\gamma_{i,r}^j = \frac{(\sigma_i^j)^2 s_r^j + (\varepsilon_r^j)^2 \mu_i^j}{(\rho_{i,r}^j)^2} \quad \text{et} \quad \lambda_{i,r}^j = \frac{\sigma_i^j \varepsilon_r^j}{\rho_{i,r}^j}$$

où  $\Psi_{i,r}^j(z^j)$  est l'effet de la  $r^{\text{ième}}$  information élémentaire  $s_r^j$  de  $j^{\text{ième}}$  attribut  $z^j$  sur la composante marginale  $p(z^j | C_i)$  en chaque composante  $C_i$ .

En utilisant les expressions (A.16) et (A.17), l'expression (A.15) peut s'écrire,

$$p(C_i, Z, s_r, S) = p\{C_i\} \pi_r \prod_{j=1}^n \beta_{i,r}^j \Psi_{i,r}^j(z^j) \tag{A.18}$$

elle est l'expression de la mixture de fonctions normales univariées.

## Inférence

Le processus d'inférence probabiliste implique la déduction de certains attributs de  $Z$  à partir de l'information disponible des autres attributs. Pour bien identifier les attributs désirés on va les partager en deux sous-ensembles :  $Z = \{X, Y\}$  où  $X = \{z^d\}$  et  $Y = \{z^0\}$  avec  $z^d$  représentant les attributs désirés et  $z^0$  les autres attributs. En appliquant la même idée sur l'information disponible  $S$  (information fournie par les capteurs). Elle est divisée aussi en deux sous-ensembles :  $S_r = \{s_r^x, s_r^y\}$  où  $s_r^x$  représente la  $r^{\text{ième}}$  observation des attributs désirés  $X$  et indépendants du

modèle  $p(Z)$  ; souvent  $s_r^x$  manque, cependant quand il est présent, il représente un *a priori* sur la valeur des attributs recherchés.  $s_r^y$  représente l'information sur le reste des attributs  $Y$ . Avec cette convention nous pouvons écrire :

$$\begin{aligned} p(ZS) &= p(XYS) \\ &= \sum_{i=1}^m \sum_{r=1}^R p\{C_i\} \pi_r \beta_{i,r} \Psi_{i,r}(X) \Psi_{i,r}(Y) \end{aligned} \quad (\text{A.19})$$

où,

$$\beta_{i,r} = \prod_{j=1}^n \beta_{i,r}^j$$

Les termes  $\Psi_{i,r}(Z)$  sont organisés en accord avec la partition de  $Z = \{X, Y\}$  :

$$\Psi_{i,r}(X) = \prod_d \Psi_{i,r}^d(z^d) \quad \text{et} \quad \Psi_{i,r}(Y) = \prod_0 \Psi_{i,r}^0(z^0)$$

Nous voudrions trouver les attributs inconnus  $X$  en fonction de l'information disponible  $S$ .

Du théorème de Bayes :

$$p(X|S) = \frac{p(XS)}{p(S)}, \quad (\text{A.20})$$

en appliquant la règle de la marginalisation nous avons :

$$\begin{aligned} p(XS) &= \int_y p(XYS) dy \\ &= \sum_{i=1}^m \sum_{r=1}^R p\{C_i\} \pi_r \beta_{i,r} \Psi_{i,r}(X) \end{aligned} \quad (\text{A.21})$$

$$\begin{aligned} p(S) &= \int_Z p(ZS) dZ \\ &= \sum_{i=1}^m \sum_{r=1}^R p\{C_i\} \pi_r \beta_{i,r}, \end{aligned} \quad (\text{A.22})$$

en substituant (A.21) et (A.22) en (A.20) et en appliquant les propriétés (A.2) et (A.3) nous avons l'expression pour l'inférence :

$$p(X | S) = \sum_{i=1}^m \sum_{r=1}^R \alpha_{i,r} \Psi_{i,r}(X) \quad (\text{A.23})$$

avec,

$$\begin{aligned} \alpha_{i,r} &= p\{C_i s_r | S\} \\ &= \frac{p\{C_i\} \pi_r \beta_{i,r}}{\sum_{k=1}^m \sum_{l=1}^R p\{C_k\} \pi_l \beta_{k,l}} \end{aligned}$$

Quand la densité de probabilité conjointe et les fonctions de probabilités sont approchées par le modèle de mixtures avec la structure proposée, le calcul des densités conditionnelles est simplifié car les composantes réduisent l'intégration pluridimensionnelle à la simple combinaison d'intégrales univariées.

### A.3.4 Apprentissage de la fonction de densité de probabilité conjointe

Les paramètres de la mixture de distributions normales univariées sont estimés avec l'algorithme *EM* (Expectation-Maximization) [DLR77, RW84, Tan96], l'idée est de maximiser la probabilité de la base d'apprentissage  $J = \prod_{k=1}^M p(Z^{(k)})$ , cependant la tâche est complexe du point de vue mathématique à cause du produit de sommes imbriquées. Néanmoins, si les composantes générant chaque exemple sont connues,  $J$  peut être réécrit de façon différente pour le maximiser. Au lieu de maximiser toute la densité de probabilité qui est inconnue, nous pouvons itérativement maximiser sa valeur prévue en donnant la base d'apprentissage et les paramètres actuelles de la mixture.

Voici une description succincte de la méthode d'apprentissage (pour une description rigoureux voir [MK97]). Les paramètres du mélange sont des espérances conditionnelles :

$$E_{Z|C_i}\{g(Z)|C_i\} = \int_Z g(Z)p(Z|C_i)dZ \quad (\text{A.24})$$

en particulier,

$$\begin{aligned} p\{C_i\} &= E\{p\{C_i|Z\}\} \\ \mu_i^j &= E\{z^j|C_i\} \\ (\sigma_i^j)^2 &= E\{(z^j - \mu_i^j)^2|C_i\}. \end{aligned}$$

On réécrit l'expression (A.24) en utilisant le théorème de Bayes :

$$\begin{aligned} E_{Z|C_i}\{g(Z)|C_i\} &= \int_Z \frac{g(Z)p\{C_i|Z\}p(Z)}{p\{C_i\}}dZ \\ &= \frac{E_Z\{g(Z)p\{C_i|Z\}\}}{p\{C_i\}} \end{aligned} \quad (\text{A.25})$$

L'algorithme *EM* peut être vu comme une méthode d'actualisation itérative des paramètres de la mixture en utilisant l'expression (A.25) comme une moyenne empirique sur la base d'apprentissage.

Les pas *E* et *M* suivants sont faits jusqu'à ce que la probabilité  $J$  ne s'améliore plus.

- E - le pas d'espérance calcule les probabilités  $q_i^{(k)} \equiv p\{C_i|Z^{(k)}\}$  générées avec le  $k^{\text{ième}}$  exemple et la  $i^{\text{ème}}$  composante du mélange. L'expression utilisée est :

$$p\{C_i\} \leftarrow \frac{p(Z^{(k)}|C_i)p\{C_i\}}{p(Z^{(k)})}$$

- M - le pas de maximisation actualise les paramètres de chaque composante en utilisant toute la base d'apprentissage avec la probabilité  $q_i^{(k)}$  de chaque échantillon. D'abord la probabilité *a priori* de chaque composante :

$$p\{C_i\} \leftarrow \frac{1}{M} \sum_{k=1}^M q_i^{(k)}$$

ensuite, la moyenne et l'écart-type de chaque composante avec ;

$$\mu_i^j \leftarrow \frac{1}{Mp\{C_i\}} \sum_{k=1}^M [q_i z^j]^{(k)}$$

$$(\sigma_i^j)^2 \leftarrow \frac{1}{Mp\{C_i\}} \sum_{k=1}^M [q_i (z^j)^2]^{(k)} - (\mu_i^j)^2$$

Dans toutes les expressions de l'algorithme *EM*,  $^{(k)}$  signifie que l'expression utilise le  $k^{\text{ième}}$  échantillon d'apprentissage.

### A.3.5 Discussion

La méthode d'inférence probabiliste et d'apprentissage consiste à approcher la densité de probabilité conjointe d'un phénomène par une mixture de distributions normales univariées en considérant la mutuelle indépendance entre les attributs du phénomène. Cette mixture est apprise à partir d'exemples en utilisant une version modifiée de l'algorithme *EM*.

La probabilité de l'observation est aussi représentée par des mixtures de distributions normales univariées. La structure mathématique de cette représentation simplifie le calcul informatique car il n'y a pas d'intégration numérique pour faire l'inférence probabiliste.

La méthode peut être bien utilisée pour la reconnaissance d'objets et pour l'approximation de n'importe quelle fonction. Nous pouvons aussi modéliser des phénomènes sur lesquels nous ne connaissons pas les fonctions de densités de probabilité.

À notre avis, l'utilisation de distributions normales univariées présente de nombreux avantages aussi bien sur le plan du calcul que sur celui de la représentation du modèle. Les modèles obtenus sont faciles à gérer et facilement utilisables pour la reconnaissance de forme.

Les remarques que nous avons sur cette approche sont les suivantes :

1. Quand on a besoin d'avoir plus de précision sur l'inférence (comme dans le CAO-Robotique pour faire des inversions cinématiques), cette approche est peu précise car Ruiz a fait l'hypothèse de mutuelle indépendance sur les attributs du phénomène.
2. Il existe toujours le problème de la maximisation car la mixture de distributions est la somme de produits de normales.

## Annexe B

# Calibration et problèmes géométriques

Parmi l'ensemble de capteurs utilisés en robotique, comme les capteurs de force, de proximité ou de température, la vision par ordinateur joue un rôle prédominant [ECR92]. Elle est largement utilisée dans de nombreuses applications robotiques [JRGB94, PGRKB95, MCSP01] touchant à des domaines variés allant du nucléaire à l'espace en passant par les applications maritimes et médicales. Dans cet annexe, nous décrivons les différents programmes probabilistes nécessaires pour localiser un objet avec une paire stéréoscopique constituée de deux caméras noir et blanc.

La localisation de points dans un espace tridimensionnel au moyen d'un système de vision stéréoscopique implique un processus de triangulation ayant lieu entre la paire de caméras impliquées et l'objet visé par le système. La triangulation a besoin de la connaissance des relations géométriques entre les dispositifs impliqués, c'est-à-dire : les deux caméras et l'objet. Ces relations géométriques se trouvent dans les paramètres des caméras, à savoir la taille des points dans l'image, la focal de la caméra, le centre de projection et la position de la caméra dans la scène [HM95]. Le processus de calcul de ces paramètres est appelé la calibration d'une caméra.

Dans l'annexe B.1, nous décrivons la façon probabiliste de calibrer une caméra et nous montrons que ce processus est équivalent à une méthode classique.

Dans l'annexe B.2, nous présentons la calibration stéréoscopique, elle consiste à inférer la transformation géométrique existante entre les deux caméras du système stéréoscopique. Cela est nécessaire pour compléter l'information de la triangulation afin de trouver la position d'un objet. Nous détaillons dans cette section le processus de localisation d'un point dans l'espace étant données ses images dans les caméras du système stéréoscopique. La solution proposée est probabiliste et utilise le programme probabiliste permettant de calibrer une caméra.

L'annexe B.3 présente le processus probabiliste de calibration précédent utilisé pour calculer la transformation permettant de passer du repère du système stéréoscopique au repère du bras manipulateur. Grâce à ce changement de repère, le bras robot peut utiliser l'information du système stéréoscopique pour déterminer la position des objets à saisir.

Nous donnons des résultats expérimentaux dans l'annexe B.4. La dernière section présente la

conclusion de cette annexe.

## B.1 Autocalibration d'une caméra

La localisation d'un point avec un système stéréoscopique implique la connaissance des paramètres géométriques et physiques de la caméra. L'ensemble de ces paramètres se trouvent dans une matrice de projection. Le processus de calcul de cette matrice est nommé "l'autocalibration de la caméra".

Les sections 1, 2, 3, 4 et 5 décrivent la géométrie et les paramètres internes et externes de la caméra comme ils sont décrits dans [HM95].

Les sections 6, 7, 8, 9, 10 et 11 abordent le problème de calibration en utilisant l'approche bayésienne. Enfin nous montrerons que dans le cadre bayésien la résolution de ce problème revient aux méthodes classiquement employées.

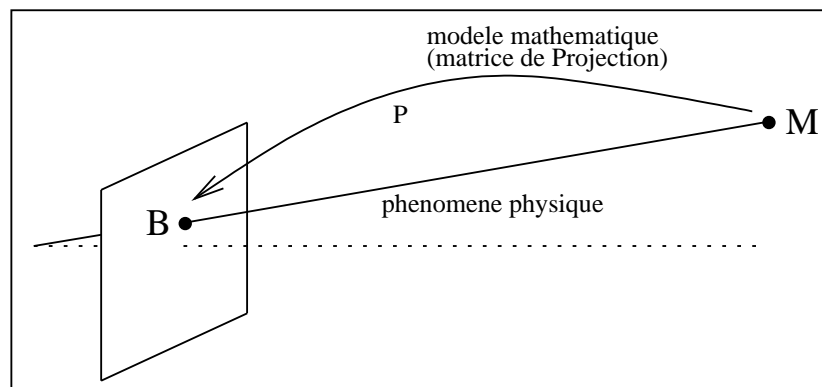


FIG. B.1 – Phénomène physique dans une caméra

### B.1.1 Définition

La figure B.1 représente le phénomène ayant lieu dans la caméra. Le point  $M$  dans l'espace existe physiquement, le point  $B$  est sa projection dans l'image de la caméra. Le modèle mathématique correspondant à ce phénomène est constitué d'une matrice  $P$  dite "de projection" calculant la position du point  $B$  dans l'image étant donné le point  $M$ . La calibration d'une caméra consiste à déterminer cette "matrice de projection".

La matrice de projection permet de calculer la position d'un point dans l'espace sur l'image de la caméra. Le problème contraire, c'est-à-dire, le problème de trouver le points dans l'espace à partir de son image dans la caméra, n'a pas de solution à cause de la réduction d'information

ayant lieu dans la caméra, c'est-à-dire, l'information 3D dans l'espace est représentée en 2D dans la caméra. Néanmoins, le problème est résolu en utilisant deux caméra (voir calibration stéréoscopique dans cette annexe). Ceci implique leurs matrices de projection.

Le modèle d'une caméra se compose de deux transformations inclus dans la matrice de projection :

1. La première est une transformation homogène qui permet de passer des coordonnées d'un point exprimé dans un repère donné : le repère espace, aux coordonnées de ce point dans le repère de la caméra. Elle est connue sous le nom de *Matrice de paramètres extrinsèques*.
2. La deuxième transformation permet de passer de l'expression d'un point dans le repère de la caméra à celle dans le repère de l'image. Cette transformation implique une projection avec changement des unités de mesure. On l'appelle la *Matrice de paramètres intrinsèques* et contient implicitement les paramètres internes de la caméra, à savoir : la focale, le centre de projection et la taille des points.

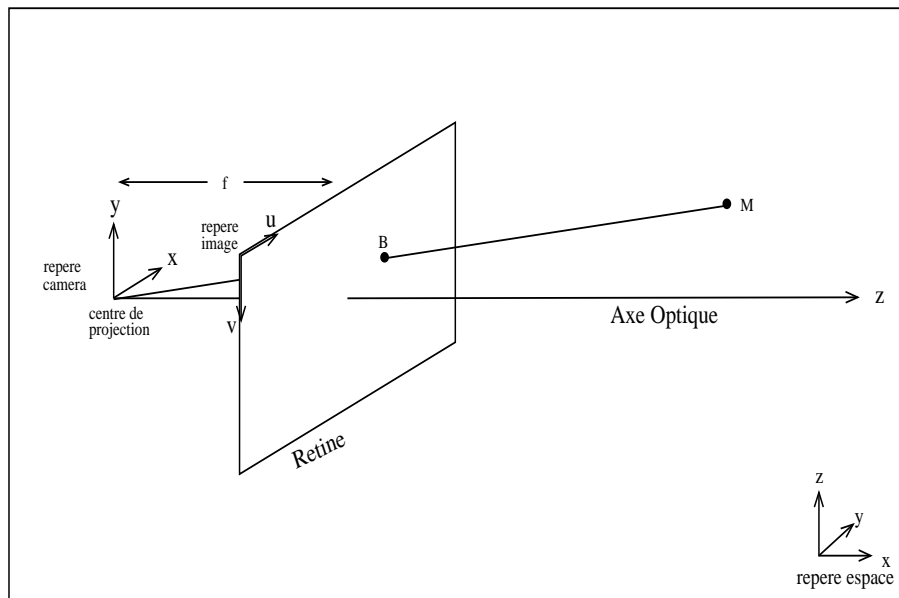


FIG. B.2 – Modèle géométrique d'une caméra

### B.1.2 Transformation Espace - Caméra

Le point  $M$  dans la figure B.2 est un point exprimé dans le repère de l'espace :

$$M = (x, y, z, 1)^T$$

Le même point  $M$  est représenté dans le repère de la caméra par :

$$M_c = (x_c, y_c, z_c, 1)^T$$



La transformation utilisée pour passer du repère espace au repère caméra est une transformation homogène notée  ${}^cT$ . L'expression suivante représente l'opération de transformation de  $M$  à  $M_c$ .

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$M_c = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} M$$

$$M_c = {}^cTM$$

La transformation homogène  ${}^cT$  est connue comme la *matrice de paramètres extrinsèques* de la caméra [HM95]. Elle représente la rotation  $R$  et la translation  $t$  nécessaires pour passer du repère espace au repère caméra.

### B.1.3 Transformation Caméra - Rétine

Nous voulons trouver les coordonnées de la projection  $B_p$  sur la rétine du point  $M_c$ . Le point  $M_c$  est représenté sur le repère de la caméra.

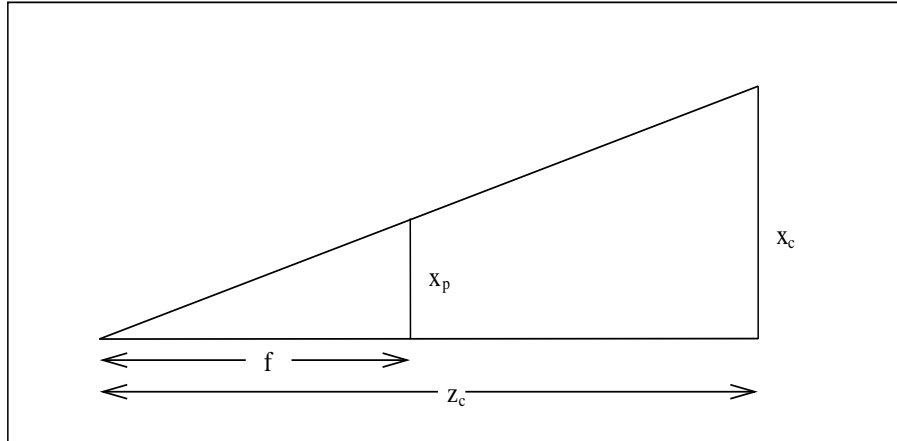


FIG. B.3 – Triangles semblables

Soit  $M_c$  un point exprimé en coordonnées caméra, nous représentons sa projection sur la rétine par :

$$B_p = (x_p, y_p, 1)^T$$

car la projection est définie en 2D.

Nous dénotons par  $B$  la projection de  $M$  en faisant bien la différence entre ces deux points.

La projection  $B_p = (x_p, y_p, 1)^T$  sur la rétine est obtenue en utilisant les propriétés des triangles semblables (figure B.3). Nous avons donc :

$$\frac{x_c}{x_p} = \frac{z_c}{f}$$

$$\frac{y_c}{y_p} = \frac{z_c}{f}$$

alors, il est possible d'établir les égalités suivantes :

$$x_c = \frac{z_c}{f} x_p \quad (\text{B.1})$$

$$y_c = \frac{z_c}{f} y_p \quad (\text{B.2})$$

En posant

$$w = \frac{z_c}{f}$$

nous pouvons écrire la projection de façon matricielle :

$$\begin{pmatrix} wx_p \\ wy_p \\ w \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}$$

$$wB_p = {}^r_c PM_c \quad (\text{B.3})$$

représentant la projection d'un point sur la rétine, où  $w$  est un facteur d'échelle.

### B.1.4 Transformation de la projection sur la rétine à l'image

Le repère de l'image est défini en deux dimensions. La transformation de la rétine à l'image implique une rotation et une translation avec un changement d'unités de mesure (figure B.4). En conséquence, le point  $B = (u, v, 1)$  dans le repère de l'image est trouvé à partir du point  $B_p = (x_p, y_p, 1)$  dans le repère rétine. En fait,  $B$  et  $B_p$  représentent le même point dans des repères différents.

De la figure B.4 représentant la transformation de la rétine à l'image, il est possible de déduire les relations :

$$u = x_p d_u + u_0$$

$$v = -y_p d_v + v_0$$

où :

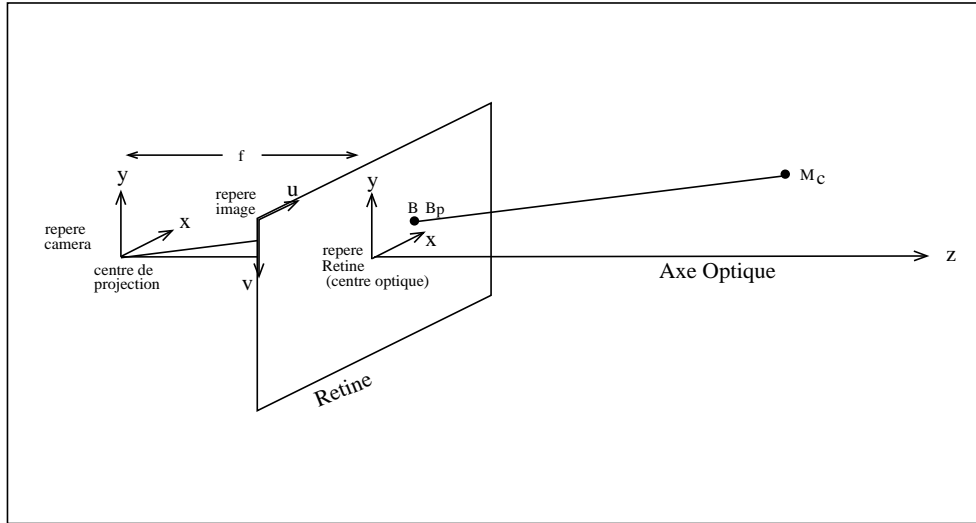


FIG. B.4 – Transformation de la rétine à l'image

- $u_0, v_0$  sont les coordonnées du centre optique de l'image dans le repère image,
- $d_u, d_v$  représentent la taille horizontale et verticale de chaque point dans l'image. En effet les points d'une caméra ne sont pas forcément carrés.

Nous pouvons écrire cette dernière expression sous forme matricielle :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} d_u & 0 & u_0 \\ 0 & -d_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} \quad (\text{B.4})$$

La focale, les coordonnées du centre de projection et la taille des points forment l'ensemble des paramètres intrinsèques de la caméra ; il est souhaitable de les avoir dans la même matrice.

En multipliant la dernière matrice obtenue par la matrice de transformation  ${}^r_cP$  (équation B.3) représentant la projection de la caméra à la rétine, on obtient :

$$K = \begin{pmatrix} d_u & 0 & u_0/f & 0 \\ 0 & -d_v & v_0/f & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} = \begin{pmatrix} d_u & 0 & u_0 \\ 0 & -d_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix}$$

Il ne faut pas oublier que la matrice  ${}^r_cP$  représente une projection à un facteur d'échelle  $w$  près. Pour établir la projection nous devons utiliser ce facteur  $w$  et écrire :

$$wB = KM_c \quad (\text{B.5})$$

Puisque la matrice  $K$  est établie avec un facteur d'échelle, il est possible de la multiplier par

la focale  $f$  sans conséquence :

$$K = \begin{pmatrix} d_u f & 0 & u_0 & 0 \\ 0 & -d_v f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

La matrice  $K$  est connue comme la *matrice de paramètres intrinsèques* [HM95], elle s'écrit :

$$K = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{B.6})$$

avec :  $\alpha_u = d_u f$  et  $\alpha_v = -d_v f$ .

Elle nous permet de calculer la projection  $B = (u, v, 1)$  sur l'image du point  $M_c$  (sur le repère caméra) à un facteur d'échelle près.

La projection est une transformation d'un espace 3D sur un espace 2D. Cela implique une réduction de l'information disponible. En fait, dans ce calcul on perd la coordonnée  $Z_c$ . Par conséquent, on est incapable de faire le calcul inverse. C'est-à-dire qu'à partir d'un point sur l'image il n'est pas possible de connaître la position du point dans l'espace avec une seule caméra.

### B.1.5 Matrice de projection

Le produit de la matrice de paramètres intrinsèques et de la matrice de paramètres extrinsèques est une matrice connue comme la *matrice de projection* [HM95], elle contient tous les paramètres nécessaires du calcul de la projection d'un point dans l'espace sur un point de l'image ;

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$P = K {}^cT \quad (\text{B.7})$$

La projection  $B$  (dans l'image de la caméra) du point  $M$  (dans l'espace sur le repère du monde) est calculée ainsi :

$$wB = PM \quad (\text{B.8})$$

avec,

$$B = (u, v, 1)$$

où  $w$  est le facteur d'échelle.

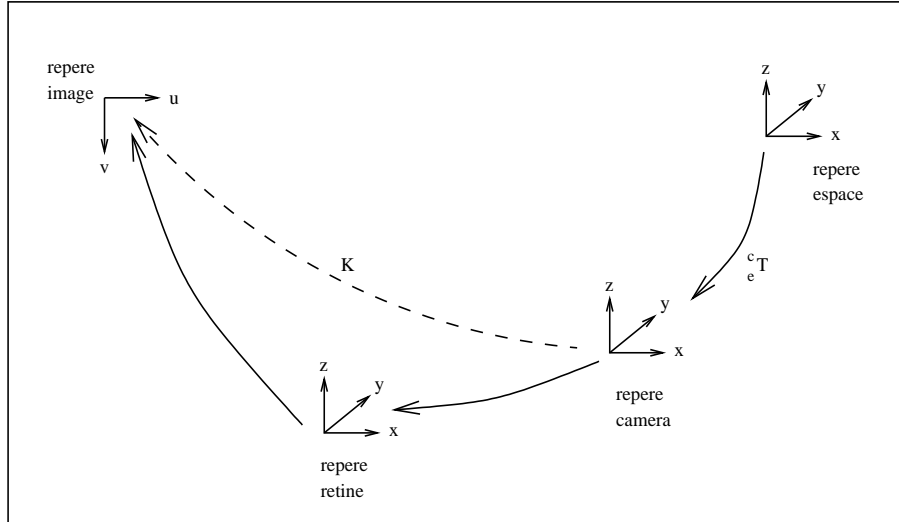


FIG. B.5 – Transformations géométriques et projection dans une caméra

Les transformations impliquées dans la matrice de projection sont représentées dans la figure B.5. Les paramètres extrinsèques sont représentés par la transformation  ${}^cT_e$  et la matrice de paramètres intrinsèques est représentée par  $K$ , elle contient une projection du repère caméra au repère rétine et la transformation de la rétine à l'image.

Dans les sections suivantes nous allons calculer la matrice de projection  $P$ . Puisque la calibration du système stéréoscopique (voir section B.2.1) a besoin des paramètres extrinsèques des caméras impliquées, nous montrons la façon d'obtenir la matrice de paramètres extrinsèques de la caméra étant donnée la matrice de projection [HM95].

Nous pouvons écrire la matrice de projection comme :

$$P = \begin{pmatrix} \mathbf{p}_1 & p_{14} \\ \mathbf{p}_2 & p_{24} \\ \mathbf{p}_3 & p_{34} \end{pmatrix} \quad (\text{B.9})$$

avec  $\mathbf{p}_i = (p_{i1} \ p_{i2} \ p_{i3})$ .

Les paramètres intrinsèques et extrinsèques de la caméra sont calculés à partir de la matrice de projection, ainsi :

$$\begin{aligned} \mathbf{r}_3 &= \mathbf{p}_3 \\ u_0 &= \mathbf{p}_1 \cdot \mathbf{p}_3 \\ v_0 &= \mathbf{p}_2 \cdot \mathbf{p}_3 \\ \alpha_u &= -\|\mathbf{p}_1 \wedge \mathbf{p}_3\| \end{aligned}$$

$$\begin{aligned}
\alpha_v &= -\|\mathbf{p}_2 \wedge \mathbf{p}_3\| \\
\mathbf{r}_1 &= 1/\alpha_u(\mathbf{p}_1 - u_0\mathbf{p}_3) \\
\mathbf{r}_2 &= 1/\alpha_v(\mathbf{p}_2 - v_0\mathbf{p}_3) \\
t_x &= 1/\alpha_u(m_{14} - u_0m_{34}) \\
t_y &= 1/\alpha_v(m_{24} - v_0m_{34}) \\
t_z &= m_{34}
\end{aligned} \tag{B.10}$$

Avec ces paramètres, on peut reconstruire la matrice de paramètres intrinsèques et la matrice de paramètres extrinsèques de la caméra. Cette dernière matrice s'écrivant :

$$T = \begin{pmatrix} \mathbf{r}_1 & t_x \\ \mathbf{r}_2 & t_y \\ \mathbf{r}_3 & t_z \\ \mathbf{0} & 1 \end{pmatrix} \tag{B.11}$$

Nous venons de présenter le modèle mathématique d'une caméra. Les diverses transformations impliquées dans sa géométrie ont été décrites. Dans la suite du chapitre nous, allons les utiliser pour aborder le problème de calibration d'une caméra de façon probabiliste.

## B.1.6 Calibration

De la partie précédente de cette section, nous connaissons la géométrie d'une caméra et nous savons que la calibration d'une caméra consiste à estimer la matrice de projection  $P$  en connaissant les coordonnées de plusieurs points dans l'espace et les coordonnées de leurs projections sur l'image. Nous utilisons une grille de calibration (figure B.6) pour repérer les points dans l'espace qui seront utilisés pour la calibration de la caméra.

## B.1.7 Variables

La figure B.7 représente l'approche probabiliste pour la calibration d'une caméra. Toutes les variables impliquées dans le problème y sont représentées :

- $P$  - représente la matrice de projection de la caméra,
- $M_i$  - le point dans l'espace avec  $i = 1 \dots n$ ,
- $B_i$  - est la projection sur l'image du point  $M_i$ .

La fonction de distribution de probabilité conjointe sur ces variables représentera toute l'information connue sur le problème de la calibration. Elle s'écrit ainsi :

$$p(P M_1 \dots M_n B_1 \dots B_n).$$

Il nous faut maintenant utiliser nos connaissances préalables pour choisir une décomposition de cette distribution qui nous permettra de la calculer effectivement en tout point.

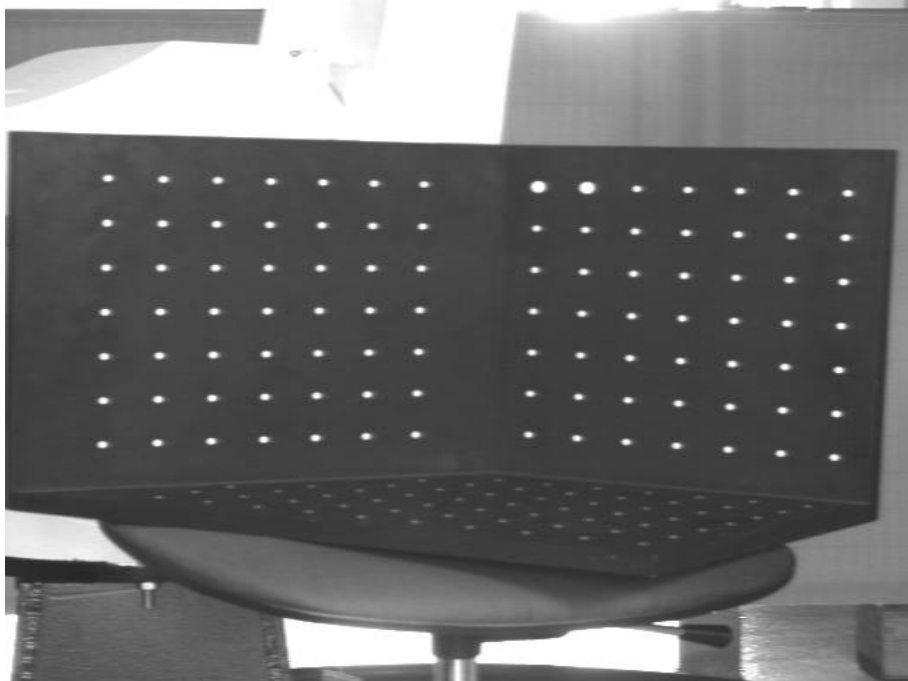


FIG. B.6 – Grille de calibration

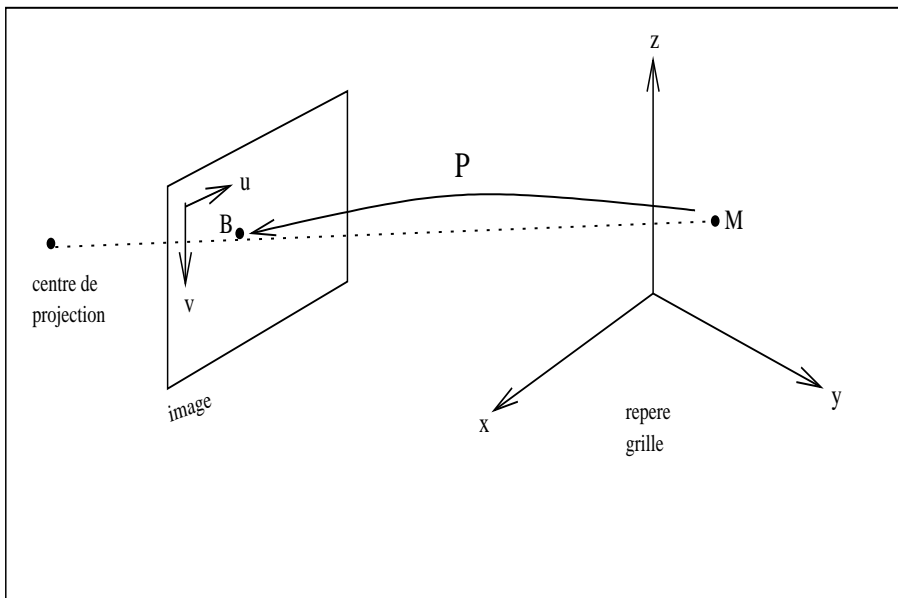


FIG. B.7 – Calibration probabiliste

## B.1.8 Décomposition

Du modèle géométrique (figure B.2) et mathématique (expression B.8) de la caméra, nous établissons les hypothèses suivantes :

1. La position du point  $M_i$  dans l'espace et la matrice de projection  $P$  sont mutuellement indépendants.
2. La projection  $B_i$  dépend du point  $M_i$  et de la matrice de projection  $P$ .

La décomposition choisie correspondante au modèle et hypothèses est :

$$p(P M_1 \dots M_n B_1 \dots B_n) = p(P) \prod_{i=1}^n p(M_i) p(B_i | PM_i). \quad (\text{B.12})$$

## B.1.9 Formes paramétriques

Les formes paramétriques associées aux éléments de la décomposition complètent notre description du système.

- $p(P) = \text{Uniforme}$  - la matrice de projection a une distribution uniforme car nous n'avons aucun *a priori* sur sa valeur,
- $p(M_i) = \text{Uniforme}$  - la distribution uniforme sur la localisation du point  $M_i$  exprime le fait qu'il n'y a pas de restriction sur sa position,
- $p(B_i) = \text{Normale}(\bar{B}_i[P, M_i], \Sigma)$  - représente une distribution normale bivariée où :  
 $\bar{B}_i = (\bar{u}, \bar{v}, 1)^T \propto PM_i$  représente la projection estimée du point  $M_i$  avec la matrice de projection  $P$ .  
 $\Sigma$  représente la matrice de covariance de la caméra ;

$$\Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$

Si on considère que les écarts type des coordonnées  $u$  et  $v$  de la projection sont égaux et que les deux coordonnées sont mutuellement indépendantes, la distribution normale s'écrit :

$$\text{Normale}(\bar{B}_i[P, M_i], \Sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2} \left[ \left( \frac{u_i - \bar{u}_i}{\sigma} \right)^2 + \left( \frac{v_i - \bar{v}_i}{\sigma} \right)^2 \right]}$$

La décomposition et les formes paramétriques ont établi un cadre de connaissance  $c_{cam}$  avec lequel, nous pouvons poser diverses questions sur les variables impliquées. L'une de ces questions correspond au problème de la calibration de la caméra.

## B.1.10 Inférence

Nous connaissons les coordonnées des points dans l'espace  $M_i$  et de leurs projections  $B_i$ , en plus nous avons déjà choisi une décomposition et les formes paramétriques associées, alors nous



pouvons poser la question :

*Quelle est la probabilité de la matrice de projection  $P$  sachant les coordonnées des points  $M_i$  dans l'espace et de leurs projections  $B_i$  sur l'image de la caméra ?*

De façon mathématique :

$$p(P \mid M_1 \dots M_n B_1 \dots B_n c_{cam}). \quad (\text{B.13})$$

Pour faire l'inférence probabiliste, nous appliquons le théorème de Bayes sur la distribution de probabilité représentée par l'expression (B.12). Nous obtenons :

$$p\left(P \mid \begin{array}{c} M_1 \dots M_n \\ B_1 \dots B_n \end{array} c_{cam}\right) \propto p(P) \prod_{i=1}^n p(M_i) p(B_i \mid PM_i) \quad (\text{B.14})$$

représentant l'espace des solutions pour la matrice de projection  $P$ . Il nous faut choisir une solution dans cette espace.

### B.1.11 Solution

Étant donné qu'il s'agit d'un problème de calibration impliquant une solution unique, la solution semblant la plus adéquate est celle maximisant la probabilité suivante :

$$\begin{aligned} P_{solution} &= \max_P p(P \mid M_1 \dots M_n B_1 \dots B_n c_{cam}) \\ &= \max_P p(P) \prod_{i=1}^n p(M_i) p(B_i \mid PM_i) \end{aligned}$$

Nous pouvons simplifier l'expression en prenant en compte les formes paramétriques. Puisque les distributions uniformes ne contribuent pas à la maximisation, on peut écrire :

$$\begin{aligned} \max_P \left( P \mid \begin{array}{c} M_1 \dots M_n \\ B_1 \dots B_n \end{array} c_{cam} \right) &= \max_P \prod_{i=1}^n p(B_i \mid PM_i) \\ &= \max_P \prod_{i=1}^n \text{Normale}(\bar{B}_i[P, M_i], \Sigma) \\ &= \max_P \prod_{i=1}^n \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2} \left[ \left( \frac{u_i - \bar{u}_i}{\sigma} \right)^2 + \left( \frac{v_i - \bar{v}_i}{\sigma} \right)^2 \right]} \end{aligned} \quad (\text{B.15})$$

Étant donné que le logarithme de la distribution normale est strictement croissant, il est possible de le maximiser :

$$\max_P \left( P \mid \begin{array}{c} M_1 \dots M_n \\ B_1 \dots B_n \end{array} c_{cam} \right) = \max_P \ln \left( \prod_{i=1}^n \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2} \left[ \left( \frac{u_i - \bar{u}_i}{\sigma} \right)^2 + \left( \frac{v_i - \bar{v}_i}{\sigma} \right)^2 \right]} \right)$$

$$\begin{aligned}
&= \max_P \sum_{i=1}^n \left( \ln \left( \frac{1}{2\pi\sigma^2} \right) - \frac{1}{2} \left( \left( \frac{u_i - \bar{u}}{\sigma} \right)^2 + \left( \frac{v_i - \bar{v}}{\sigma} \right)^2 \right) \right) \\
&= \min_P \sum_{i=1}^n \left( (u_i - \bar{u})^2 + (v_i - \bar{v})^2 \right) \tag{B.16}
\end{aligned}$$

L'expression  $(u_i - \bar{u})^2 + (v_i - \bar{v})^2$  représente l'erreur entre la projection  $B_i$  sur l'image et son estimation  $\bar{B}_i$ . Le problème revient à la minimisation de cette erreur. Pour le minimiser, nous considérons le modèle de la caméra et nous posons le problème équivalent suivant :

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} wu \\ wv \\ w \end{pmatrix} \tag{B.17}$$

où,

- $PM$  représente l'estimation à une échelle  $w$  de la projection sur l'image du point  $M$ ,
- $B$  représente la projection de  $M$ .

Nous avons trois égalités qui sont :

$$p_{11}x + p_{12}y + p_{13}z + p_{14} = wu \tag{B.18}$$

$$p_{21}x + p_{22}y + p_{23}z + p_{24} = wv \tag{B.19}$$

$$p_{31}x + p_{32}y + p_{33}z + p_{34} = w \tag{B.20}$$

Si l'on divise l'équation (B.18) et l'équation (B.19) entre l'équation (B.20),

$$\frac{p_{11}x + p_{12}y + p_{13}z + p_{14}}{p_{31}x + p_{32}y + p_{33}z + p_{34}} = u$$

$$\frac{p_{21}x + p_{22}y + p_{23}z + p_{24}}{p_{31}x + p_{32}y + p_{33}z + p_{34}} = v$$

en suite,

$$xp_{11} + yp_{12} + zp_{13} + p_{14} - xup_{31} - yup_{32} - zup_{33} = up_{34} \tag{B.21}$$

$$xp_{21} + yp_{22} + zp_{23} + p_{24} - xvp_{31} - yvp_{32} - zvp_{33} = vp_{34} \tag{B.22}$$

Pour chaque point et sa projection on a deux équations avec 12 inconnues  $p_{ij}$ . Si la grille de calibration a  $n$  points, on a  $2n$  équations. Pour résoudre ce système on a 11 inconnues car on

considère  $p_{34} = 1$  pour éliminer la solution triviale. Ainsi, le système devient :

$$\begin{pmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -x_1 u_1 & -y_1 u_1 & -z_1 u_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -x_1 v_1 & -y_1 v_1 & -z_1 v_1 \\ & & & & & & \vdots & & & & & \\ x_n & y_n & z_n & 1 & 0 & 0 & 0 & 0 & -x_n u_n & -y_n u_n & -z_n u_n \\ 0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & -x_n v_n & -y_n v_n & -z_n v_n \end{pmatrix} \begin{pmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ \vdots \\ u_n \\ v_n \end{pmatrix} \quad (\text{B.23})$$

de façon simplifiée, on peut le voir ainsi :

$$Ax = b$$

Les éléments de la matrice  $A$  sont tirés des points dans l'espace  $M_i = (x_i, y_i, z_i, 1)$  et leurs projections tandis que les éléments de la matrice  $b$  sont tirés de leurs projections  $B_i = (u_i, v_i, 1)$ .

Étant donné que le système d'équations (B.23) est un système d'équations linéaire sur-déterminé qui n'a pas de solution, nous cherchons les valeurs de  $p_{ij}$  minimisant la distance entre la projection  $B$  et son estimation  $\tilde{B}$ . La méthode utilisée est celle du moindre carré avec la solution :

$$x = (A^T A)^{-1} A^T b$$

où,  $(A^T A)^{-1} A^T$  est la matrice pseudo-inverse de la matrice  $A$ .

La solution du système d'équations rend les paramètres de la matrice de projection  $P$ . C'est-à-dire la calibration de la caméra.

### B.1.12 Conclusion

Dans cette annexe, nous avons défini un programme probabiliste réalisant la calibration d'une caméra à partir d'un ensemble de points dans l'espace et de leurs projections sur l'image de la caméra. Nous montrons que ce programme est une généralisation de la solution classique pour l'autocalibration d'une caméra. Les deux processus sont équivalents dans le cas où l'inférence est faite par la méthode des moindres carrés avec des *a priori* gaussiens. Notons que ces *a priori* ne sont, de fait, pas valides notamment, parce qu'ils ne respectent pas les contraintes aux limites sur l'image.

Pour sortir du cadre "gaussien" et se rapprocher d'un modèle plus fin de la caméra nous pourrions conserver la même décomposition et changer les formes paramétriques. En imposant par

exemple une probabilité nulle de trouver un point en dehors de l'image. Cette information de bon sens n'est pas prise en compte dans les méthodes de calibration, elle apporte cependant une information supplémentaire au procédé de calibration.

Dans l'annexe suivante nous présentons la calibration du système stéréoscopique impliquant les calibrations des caméras le composant.

## B.2 Vision stéréoscopique

La vision stéréoscopique présente l'avantage de permettre la perception de la profondeur dans l'espace. Dans notre cas, cette capacité devient importante car nous en aurons besoin pour localiser les objets.

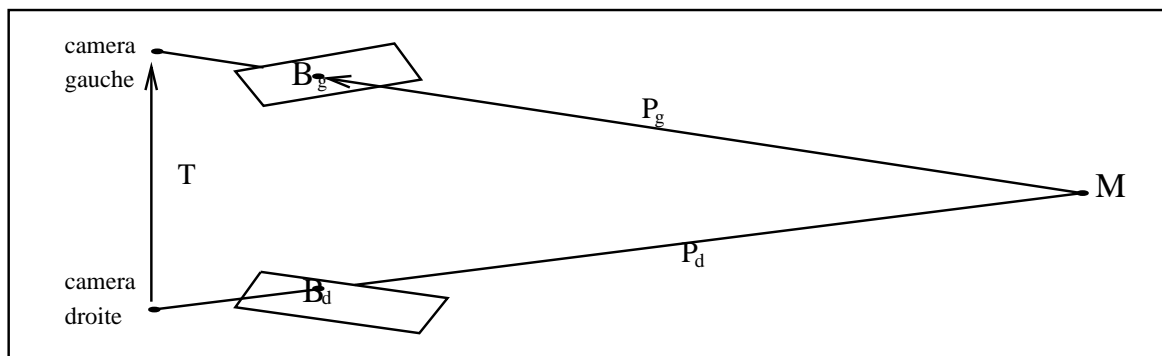


FIG. B.8 – triangulation stéréoscopique.

La localisation avec la vision stéréoscopique consiste en une triangulation (figure B.8) impliquant les caméras du système stéréoscopique rigide (décrit dans la section 3.1.2). et la relation géométrique existant entre elles.

La *configuration de la paire stéréoscopique* est constituée de plusieurs paramètres parmi lesquels : la relation géométrique entre les caméras et les caractéristiques internes de chaque caméra (les matrices de projection).

Nous décrivons dans dans cette annexe la calibration de la paire stéréoscopique.

### B.2.1 Calibration de la paire stéréoscopique

Le processus de calibration de la paire stéréoscopique rigide consiste en :

1. la calibration de chacune des caméras en utilisant le processus précédemment décrit,

2. la définition d'un repère unique pour le système de caméras.

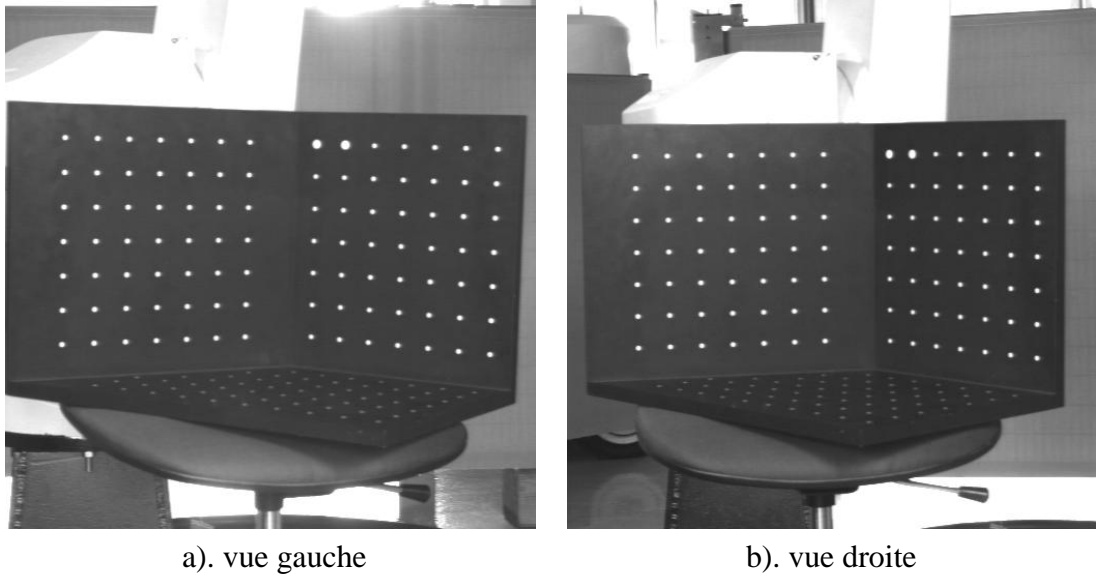


FIG. B.9 – Grille de calibration vue par la paire stéréoscopique

### B.2.1.1 Calibration des caméras de la paire stéréoscopique

La calibration stéréoscopique commence par la calibration de chacune des caméras en utilisant la même grille de calibration. Ces opérations se font sur une scène fixe : la grille de calibration et la paire stéréoscopique occupent la même position durant les deux calibrations (figure B.9). La calibration de chaque caméra est faite avec la méthode décrite dans la section B.1 en utilisant l'image de la grille correspondante.

### B.2.1.2 Changement du repère de référence de la paire stéréoscopique

Le processus de calibration d'une caméra donne la matrice de projection dans le repère attaché à la grille de calibration. Pour des raisons de commodité nous souhaitons maintenant exprimer les matrices de projection dans le repère de l'une des caméras (la caméra gauche par exemple).

La paire stéréoscopique a deux matrices de projection (figure B.10) :

1. la matrice de projection  $P_g$  de la caméra gauche,
2. la matrice de projection  $P_d$  de la caméra droite.

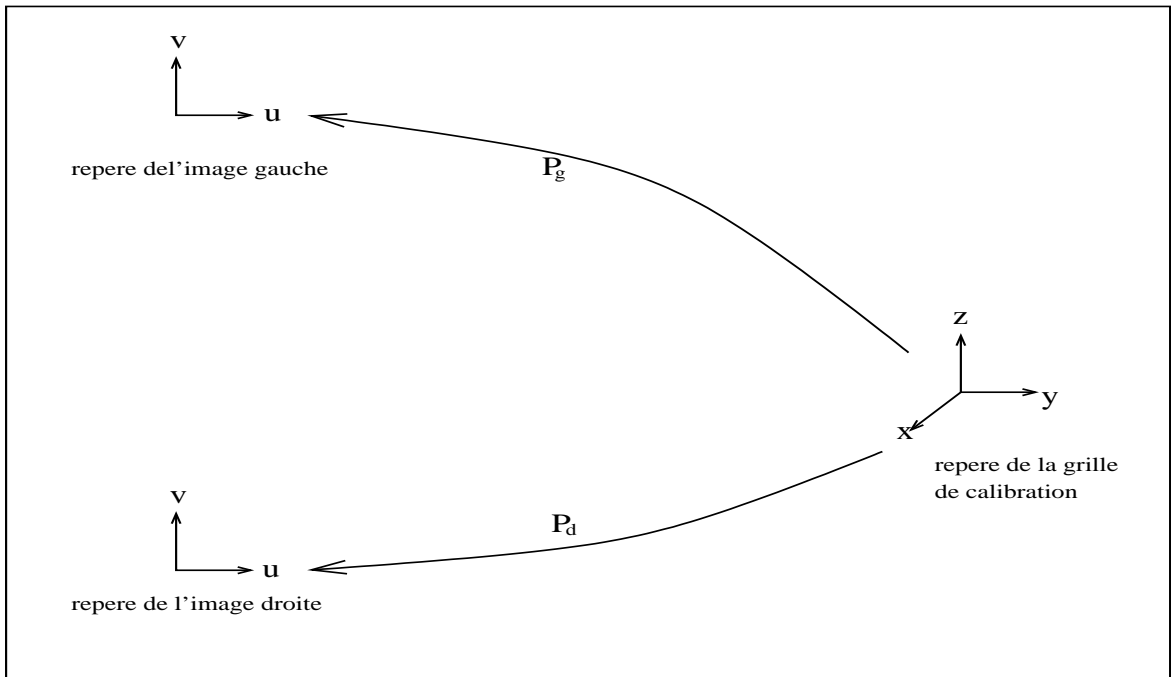


FIG. B.10 – Transformations dans la paire stéréoscopique

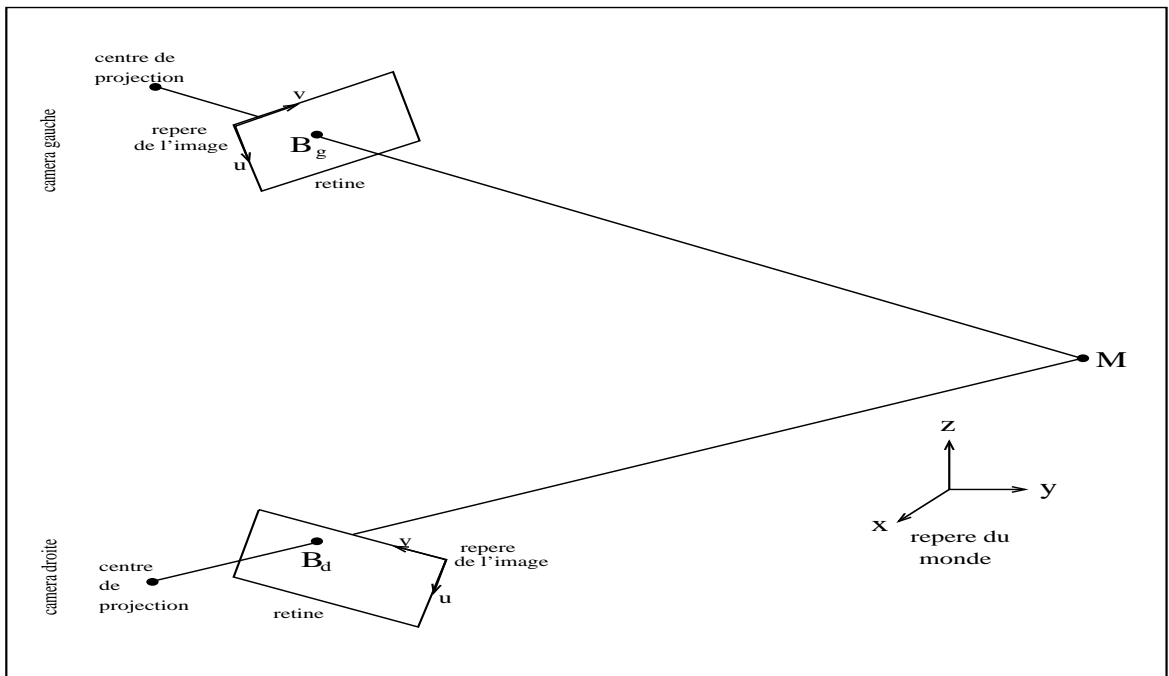


FIG. B.11 – Modèle de la paire stéréoscopique.

La figure B.11 représente le modèle de la paire stéréoscopique où  $M$  est un point dans l'espace avec ses deux projections  $B_g$  et  $B_d$  sur les images des caméras gauche et droite.

Du processus de calibration d'une caméra, on sait que :

$$\begin{aligned} B_g &\propto P_g M \\ B_d &\propto P_d M \end{aligned} \quad (\text{B.24})$$

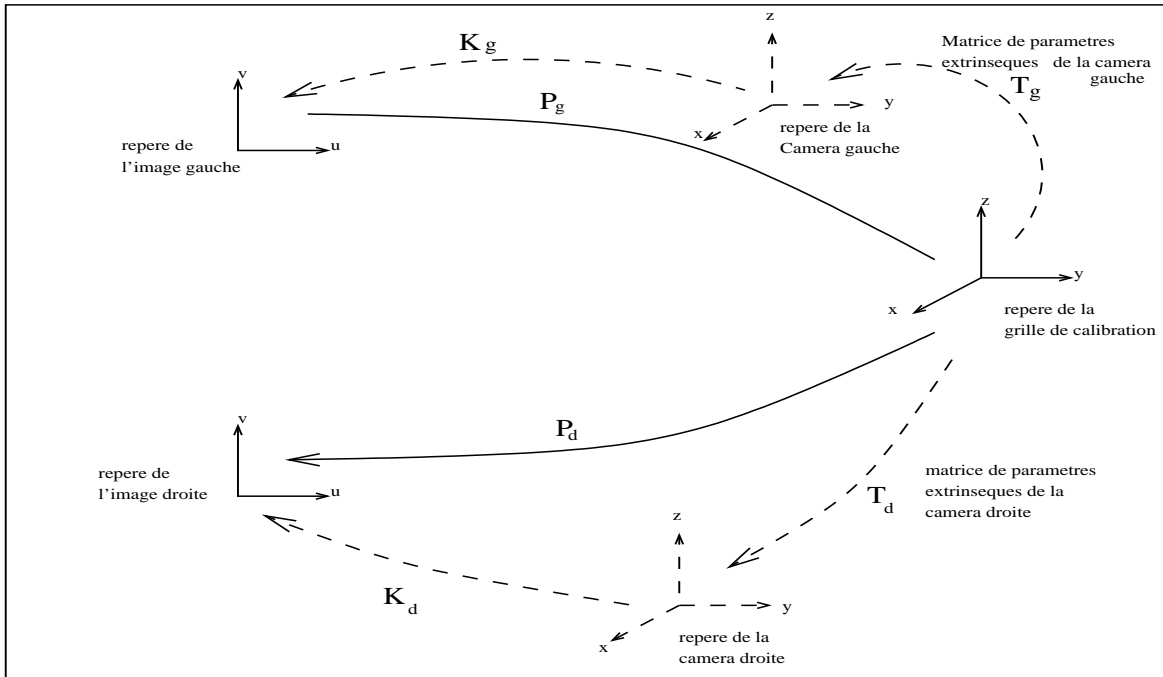


FIG. B.12 – Les matrices de projection juste après de la calibration.

Par ailleurs, on sait aussi que la matrice de projection  $P_g$  inclut deux transformations :

1. la première transformation implique une translation et une rotation du repère “monde” vers le repère de la caméra. Elle est représentée par la matrice de paramètres extrinsèques de la caméra gauche  $T_g$ ,
2. la deuxième est la transformation du repère “caméra” vers le repère “image”. Cette transformation est représentée par la matrice de paramètres intrinsèques  $K_g$ .

Nous pouvons écrire :

$$\begin{aligned} B_g &\propto P_g T_g^{-1} T_g M \\ B_d &\propto P_d T_g^{-1} T_g M \end{aligned} \quad (\text{B.25})$$

où le produit  $T_g M$  représente le point  $M$  sur le repère de la caméra gauche.

Les matrices de projection pour le nouveau repère sont obtenues ainsi :

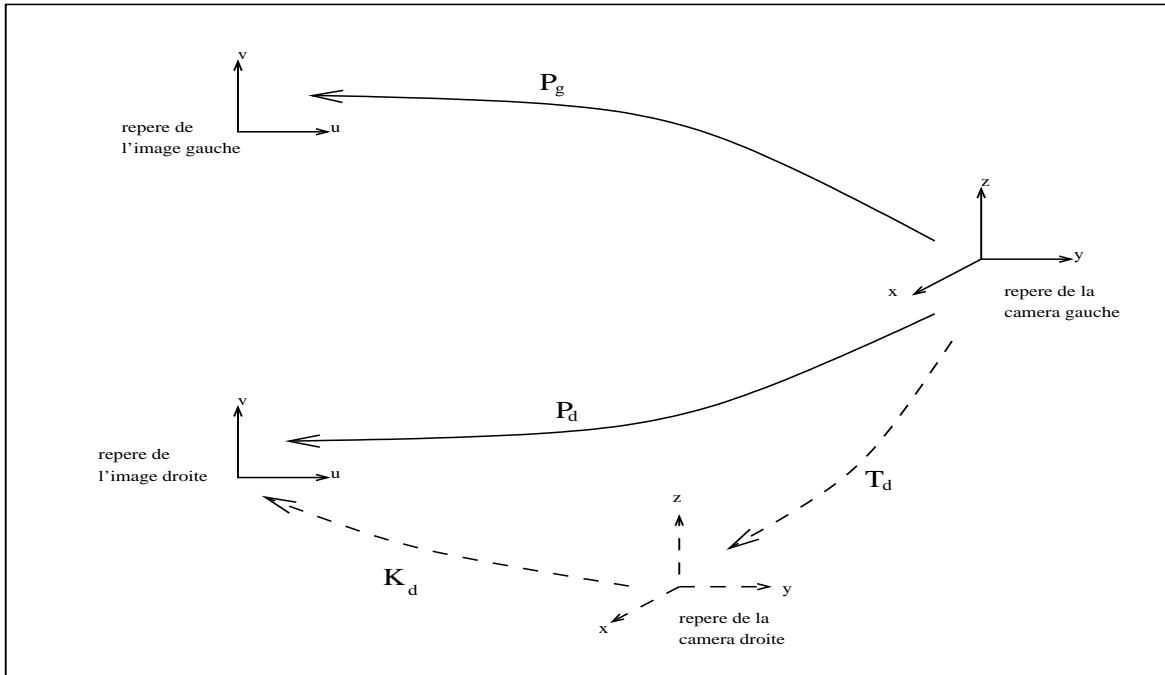


FIG. B.13 – Repère stéréoscopique sur la caméra gauche

- $P_g T_g^{-1}$  pour la caméra gauche et,
- $P_d T_g^{-1}$  pour la caméra droite.

**Attention**, nous allons utiliser les nouvelles matrices de projection, cependant nous n’allons pas changer leurs noms, cela veut dire que la nouvelle matrice de projection de la caméra gauche s’appelle  $P_g$  et  $P_d$  celle de la caméra droite.

La figure B.13 représente la nouvelle référence avec les transformations impliquées dans les matrices de projection.

Pour la nouvelle matrice de projection de la caméra gauche, on a :

- la transformation correspondant aux paramètres intrinsèques  $K_g$ , elle reste identique,
- la transformation correspondant aux paramètres extrinsèques devient la matrice identité.

Pour la matrice de projection de la caméra droite, on a :

- la transformation correspondant aux paramètres intrinsèques  $K_d$ , elle reste identique,
- la transformation correspondant aux paramètres extrinsèques change ainsi :  $T_d^* = T_d * T_g$ .

Les nouvelles matrices de projection  $P_g$  et  $P_d$  représentent la calibration de la paire stéréoscopique, le problème restant à résoudre est de calculer la position d’un point dans l’espace à l’aide de ce système. Nous allons de nouveau aborder ce problème de façon probabiliste.



## B.2.2 Localisation d'un point avec la paire stéréoscopique

Le problème est de trouver la position d'un point dans l'espace en utilisant la vision stéréoscopique. Pour cela il nous faut trouver les coordonnées du point dans l'espace à partir de ses projections sur les images des caméras gauche et droite (le processus de triangulation montré dans la figure B.8). En supposant que la paire stéréoscopique ait été calibrée nous pouvons appliquer notre méthode pour résoudre ce problème de façon probabiliste.

### B.2.2.1 Variables

Nous définissons les variables suivantes :

- $P_g$  - représente la matrice de projection de la caméra gauche,
- $P_d$  - est la matrice de projection de la caméra droite,
- $M$  - est le point dans l'espace avec  $M = (x, y, z, 1)^T$  dans le repère de la paire stéréoscopique,
- $B_g$  - est la projection  $(u_g, v_g, 1)^T$  du point  $M$  sur l'image de la caméra gauche,
- $B_d$  - la projection  $(u_d, v_d, 1)^T$  du point  $M$  sur l'image de la caméra droite.

La densité de probabilité conjointe sur ces variables s'écrit :

$$p(P_g P_d M B_g B_d)$$

### B.2.2.2 Décomposition

Il s'agit de choisir une décomposition de la distribution conjointe. Pour cela nous faisons une hypothèse d'indépendance conditionnelle entre les deux projections d'un point connaissant la position de ce point dans l'espace.

Ce raisonnement est classique dans tous les problèmes de fusion de capteurs. Clairement, la projection d'un point sur la caméra gauche est liée à la position de la projection sur la caméra droite. Cependant ces deux projections ne sont liées que par "le phénomène" : i.e. la présence d'un point dans l'espace. La connaissance de ce point suffit à déterminer les projections respectives et donc la connaissance d'une projection n'influe pas sur la connaissance de l'autre projection dès lors que l'on connaît la cause de ces projections.

À cette hypothèse, nous rajoutons une hypothèse facilement interprétable sur l'indépendance conditionnelle des matrices de projection des caméras.

Le raisonnement précédent nous conduit à choisir la décomposition suivante :

$$p(P_g P_d M B_g B_d) = p(M) p(P_g) p(P_d) p(B_g|P_g M) p(B_d|P_d M). \quad (\text{B.26})$$

### B.2.2.3 Formes paramétriques

En donnant les formes paramétriques nous complétons la formalisation de nos connaissances préalables sur le problème de la localisation d'un point ;

- $p(M) = \text{Uniforme}$  - qui manifeste notre manque d'*a priori* sur la position du point  $M$ .
- $p(P_g) = \delta_c(P_g)$  - Sur la matrice de projection de la caméra gauche nous considérons qu'il n'y a pas d'incertitude car, on prend la matrice de projection  $P_g$  estimée par le processus de calibration de la caméra.
- $p(P_d) = \delta_c(P_d)$  - idem.
- $p(B_g | P_g M) = \text{Normale}$  - Cette forme paramétrique est définie,

$$\begin{aligned} p(B_g | P_g M) &= p(u_g | P_g M) p(v_g | P_g M) \\ &= N_{u_g}(\mu_{u_g}[P_g, M], \sigma_{u_g}) N_{v_g}(\mu_{v_g}[P_g, M], \sigma_{v_g}) \end{aligned} \quad (\text{B.27})$$

avec,

$$(\mu_{u_g}, \mu_{v_g}, 1)^T \propto P_g M.$$

- $p(B_d | P_d M) = \text{Normale}$  - cette distribution est définie de façon similaire à la distribution précédente :

$$\begin{aligned} p(B_d | P_d M) &= p(u_d | P_d M) p(v_d | P_d M) \\ &= N_{u_d}(\mu_{u_d}[P_d, M], \sigma_{u_d}) N_{v_d}(\mu_{v_d}[P_d, M], \sigma_{v_d}) \end{aligned} \quad (\text{B.28})$$

Ayant défini l'ensemble des formes paramétriques, nous avons une description complète de notre système notée :  $c_{VS}$ , il devient maintenant possible d'utiliser cette description pour résoudre des problèmes portant sur les variables impliquées.

### B.2.2.4 Inférence

Nous posons la question conduisant à trouver la position d'un point dans l'espace en fonction de ses projections sur les caméras de la paire stéréoscopique.

*Quelle est la probabilité de la position d'un point dans l'espace sachant ses projections sur les caméras de la paire stéréoscopique et la calibration des caméras ?*

De façon symbolique :

$$p(M | P_g P_d B_g B_d c_{VS}). \quad (\text{B.29})$$

En utilisant le théorème de Bayes,

$$p(M | P_g P_d B_g B_d c_{VS}) \propto p(M P_g P_d B_g B_d) \quad (\text{B.30})$$

En utilisant l'expression (B.26) de la décomposition,

$$p(M | P_g P_d B_g B_d c_{VS}) \propto p(M) p(P_g) p(P_d) p(B_g | P_g M) p(B_d | P_d M), \quad (\text{B.31})$$

représente l'espace des solutions pour la position d'un point à partir de ses projections sur les caméras du système stéréoscopique.

### B.2.2.5 Solution

Nous allons nous intéresser au point  $M$  maximisant cette probabilité,

$$\max_M p(M | P_g P_d B_g B_d c_{VS})$$

représente la solution pour la position d'un point avec la vision stéréoscopique.

Pour résoudre ce problème, on réécrit l'expression en utilisant la décomposition :

$$\begin{aligned} \max_M p(M | P_g P_d B_g B_d c_{VS}) &= \max_M p(M) p(P_g) p(P_d) p(B_g | P_g M) p(B_d | P_d M) \\ &= \max_M p(B_g | P_g M) p(B_d | P_d M) \\ &= \max_M p(u_g | P_g M) p(v_g | P_g M) p(u_d | P_d M) p(v_d | P_d M) \end{aligned} \quad (\text{B.32})$$

nous avons éliminé les termes  $p(M)$ ,  $p(P_g)$  et  $p(P_d)$  car les distributions uniformes et les diracs ne contribuent pas à la maximisation.

En utilisant les formes paramétriques des facteurs de la décomposition,

$$\begin{aligned} \max_M p(M | P_g P_d B_g B_d c_{VS}) &= \max_M \begin{bmatrix} N_{u_g}(\mu_{u_g}[P_g, M], \sigma_{u_g}) N_{v_g}(\mu_{v_g}[P_g, M], \sigma_{v_g}) \\ N_{u_d}(\mu_{u_d}[P_d, M], \sigma_{u_d}) N_{v_d}(\mu_{v_d}[P_d, M], \sigma_{v_d}) \end{bmatrix} \\ &= \max_M \begin{bmatrix} \frac{1}{\sqrt{2\pi}\sigma_{u_g}} e^{-\frac{1}{2}\left(\frac{\mu_{u_g}[P_g, M] - u_g}{\sigma_{u_g}}\right)^2} \\ \frac{1}{\sqrt{2\pi}\sigma_{v_g}} e^{-\frac{1}{2}\left(\frac{\mu_{v_g}[P_g, M] - v_g}{\sigma_{v_g}}\right)^2} \\ \frac{1}{\sqrt{2\pi}\sigma_{u_d}} e^{-\frac{1}{2}\left(\frac{\mu_{u_d}[P_d, M] - u_d}{\sigma_{u_d}}\right)^2} \\ \frac{1}{\sqrt{2\pi}\sigma_{v_d}} e^{-\frac{1}{2}\left(\frac{\mu_{v_d}[P_d, M] - v_d}{\sigma_{v_d}}\right)^2} \end{bmatrix} \end{aligned} \quad (\text{B.33})$$

Du modèle de la caméra :

$$\begin{aligned} w\mu_g &= P_g M \\ \begin{pmatrix} w\mu_{u_g} \\ w\mu_{v_g} \\ w \end{pmatrix} &= \begin{pmatrix} p_{g11} & p_{g12} & p_{g13} & p_{g14} \\ p_{g21} & p_{g22} & p_{g23} & p_{g24} \\ p_{g31} & p_{g32} & p_{g33} & p_{g34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \end{aligned}$$

alors,

$$\mu_{u_g} = \frac{w\mu_{u_g}}{w} = \frac{pg_{11}x + pg_{12}y + pg_{13}z + pg_{14}}{pg_{31}x + pg_{32}y + pg_{33}z + pg_{34}} \quad (\text{B.34})$$

$$\mu_{v_g} = \frac{w\mu_{v_g}}{w} = \frac{pg_{21}x + pg_{22}y + pg_{23}z + pg_{24}}{pg_{31}x + pg_{32}y + pg_{33}z + pg_{34}} \quad (\text{B.35})$$

pour la caméra gauche.

De la même façon pour la caméra droite :

$$\mu_{u_d} = \frac{w\mu_{u_d}}{w} = \frac{pd_{11}x + pd_{12}y + pd_{13}z + pd_{14}}{pd_{31}x + pd_{32}y + pd_{33}z + pd_{34}} \quad (\text{B.36})$$

$$\mu_{v_d} = \frac{w\mu_{v_d}}{w} = \frac{pd_{21}x + pd_{22}y + pd_{23}z + pd_{24}}{pd_{31}x + pd_{32}y + pd_{33}z + pd_{34}} \quad (\text{B.37})$$

La forme paramétrique de  $p(u_g|P_gM)$  s'écrit :

$$p(u_g|P_gM) = \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{1}{2}\left(\frac{\mu_{u_g} - u_g}{\sigma_{u_g}}\right)^2} \quad (\text{B.38})$$

En utilisant l'expression (B.34), le terme  $\mu_{u_g} - u_g$  est calculé,

$$\begin{aligned} & \mu_{u_g} - u_g \\ &= \\ & \frac{pg_{11}x + pg_{12}y + pg_{13}z + pg_{14}}{pg_{31}x + pg_{32}y + pg_{33}z + pg_{34}} - u_g \\ &= \\ & \frac{pg_{11}x + pg_{12}y + pg_{13}z + pg_{14} - pg_{31}xu_g - pg_{32}yu_g - pg_{33}zu_g - pg_{34}u_g}{pg_{31}x + pg_{32}y + pg_{33}z + pg_{34}} \end{aligned} \quad (\text{B.39})$$

Du modèle de la caméra on sait que  $wB = PM$ . Puisque le facteur de proportionnalité  $w$  n'est pas connu, il est possible d'assumer sans conséquence que  $w = 1$ , de cette façon l'expression (B.39) devient :

$$\mu_{u_g} - u_g = F_1(B_g, P_g, M) = \frac{pg_{11}x + pg_{12}y + pg_{13}z + pg_{14} - pg_{31}xu_g - pg_{32}yu_g - pg_{33}zu_g - pg_{34}u_g}{pg_{31}x + pg_{32}y + pg_{33}z + pg_{34}} \quad (\text{B.40})$$

de façon similaire pour la coordonnée  $v_g$  de l'image sur la caméra gauche :

$$\mu_{v_g} - v_g = F_2(B_g, P_g, M) = \frac{pg_{21}x + pg_{22}y + pg_{23}z + pg_{24} - pg_{31}xv_g - pg_{32}yv_g - pg_{33}zv_g - pg_{34}v_g}{pg_{31}x + pg_{32}y + pg_{33}z + pg_{34}} \quad (\text{B.41})$$

Pour le cas des coordonnées de la projection sur l'image de la caméra droite, les expressions s'écrivent :

$$\mu_{u_d} - u_d = F_3(B_d, P_d, M) = \frac{pd_{11}x + pd_{12}y + pd_{13}z + pd_{14} - pd_{31}xu_d - pd_{32}yu_d - pd_{33}zu_d - pd_{34}u_d}{pd_{31}x + pd_{32}y + pd_{33}z + pd_{34}} \quad (\text{B.42})$$

$$\mu_{v_d} - v_d = F_4(B_d, P_d, M) = \frac{pd_{21}x + pd_{22}y + pd_{23}z + pd_{24} - pd_{31}xv_d - pd_{32}yv_d - pd_{33}zv_d - pd_{34}v_d}{pd_{31}x + pd_{32}y + pd_{33}z + pd_{34}} \quad (\text{B.43})$$

où,  $pd_{ij}$  représente un terme de la matrice de projection  $P_d$  de la caméra droite.

En substituant les expression (B.40), (B.41), (B.42) et (B.43) dans l'expression (B.33),

$$\max_M p(M|P_g P_d B_g B_d c_{VS}) = \max_M \prod_{i=1}^4 \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2} \left( \frac{F_i(B_i, P_i, M)}{\sigma_i} \right)^2} \quad (\text{B.44})$$

avec :

- $\{\sigma_1, \sigma_2, \sigma_3, \sigma_4\} = \{\sigma_{u_g}, \sigma_{v_g}, \sigma_{u_d}, \sigma_{v_d}\}$ ,
- $P_1 = P_2 = P_g$ ,
- $P_3 = P_4 = P_d$ ,
- $B_1 = B_2 = B_g$  et,
- $B_3 = B_4 = B_d$

Étant donné que le logarithme de la fonction est strictement croissant, nous pouvons le maximiser :

$$\begin{aligned} \max_M p(M|P_g P_d B_g B_d c_{VS}) &= \max_M \ln \left( \prod_{i=1}^4 \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2} \left( \frac{F_i(B_i, P_i, M)}{\sigma_i} \right)^2} \right) \\ &= \max_M \sum_{i=1}^4 \left( \ln \left( \frac{1}{\sqrt{2\pi}\sigma_i} \right) - \frac{1}{2} \left( \frac{F_i(B_i, P_i, M)}{\sigma_i} \right)^2 \right) \\ &= \min_M \sum_{i=1}^4 \left( \frac{F_i(B_i, P_i, M)}{\sigma_i} \right)^2 \end{aligned} \quad (\text{B.45})$$

Cette fonction nous montre que chercher le maximum revient à : minimiser la somme du carré de l'erreur entre la projection  $B$  et la projection estimée par le produit  $PM$  du point  $M$  sur les deux caméras de la paire stéréoscopique. Cette erreur est décrite par les fonctions  $F_i$ .

Une méthode pour minimiser l'erreur et ainsi estimer  $M$  est la méthode de moindre carré que l'on décrit ensuite.

Une fonction a ses extrêmes (maximum ou minimum) où sa dérivée est zéro. Dans le cas multidimensionnel, un minimum d'une fonction se trouve où ses dérivées partielles sont égales à zéro. Alors, pour trouver le minimum, on va dériver l'expression (B.45). Mais, d'abord nous allons réarranger les termes pour les expressions des  $F_i$  :

$$F_1(B_g, P_g, M) = (pg_{11} - pg_{31}u_g)x + (pg_{12} - pg_{32}u_g)y + (pg_{13} - pg_{33}u_g)z + pg_{14} - pg_{34}u_g \quad (\text{B.46})$$

$$F_2(B_g, P_g, M) = (pg_{21} - pg_{31}v_g)x + (pg_{22} - pg_{32}v_g)y + (pg_{23} - pg_{33}v_g)z + pg_{24} - pg_{34}v_g \quad (\text{B.47})$$

$$F_3(B_d, P_d, M) = (pd_{11} - pd_{31}u_d)x + (pd_{12} - pd_{32}u_d)y + (pd_{13} - pd_{33}u_d)z + pd_{14} - pd_{34}u_d \quad (\text{B.48})$$

$$F_4(B_d, P_d, M) = (pd_{21} - pd_{31}v_d)x + (pd_{22} - pd_{32}v_d)y + (pd_{23} - pd_{33}v_d)z + pd_{24} - pd_{34}v_d \quad (\text{B.49})$$

pour alléger la notation les fonctions  $F_i$  seront notées :

$$F_i = f_{i1}x + f_{i2}y + f_{i3}z + f_{i4}$$

où, par exemple, les termes de  $F_1$  sont définis :

- $f_{11} = p_{g11} - p_{g31}u_g$ ,
- $f_{12} = p_{g12} - p_{g32}u_g$ ,
- $f_{13} = p_{g13} - p_{g33}u_g$  et,
- $f_{14} = p_{g14} - p_{g34}u_g$

de la même façon pour les fonctions  $F_2$ ,  $F_3$  et  $F_4$ .

La dérivée partielle de (B.45) par rapport à  $x$  s'obtient ainsi :

$$\begin{aligned} \frac{\delta}{\delta x} p(M|P_g P_d B_g B_d c_{VS}) &= 2 \sum_{i=1}^4 \left( \frac{F_i f_{i1}}{\sigma_i \sigma_i} \right) = 0 \\ &= \sum_{i=1}^4 \left( \frac{F_i}{\sigma_i^2} f_{i1} \right) = 0 \end{aligned} \quad (\text{B.50})$$

elle est égale à zéro car on cherche ses extrêmes.

De façon similaire pour les dérivées partielles par rapport à  $y$  et  $z$  :

$$\frac{\delta}{\delta y} p(M|P_g P_d B_g B_d c_{VS}) = \sum_{i=1}^4 \left( \frac{F_i}{\sigma_i^2} f_{i2} \right) = 0 \quad (\text{B.51})$$

$$\frac{\delta}{\delta z} p(M|P_g P_d B_g B_d c_{VS}) = \sum_{i=1}^4 \left( \frac{F_i}{\sigma_i^2} f_{i3} \right) = 0 \quad (\text{B.52})$$

De l'expression (B.50) en développant  $F_i$ , on a l'équation pour la dérivée par rapport à  $x$  :

$$\begin{aligned} \sum_{i=1}^4 \left( \frac{F_i}{\sigma_i^2} f_{i1} \right) &= 0 \\ \sum_{i=1}^4 \left( f_{i1} \frac{f_{i1}}{\sigma_i^2} x + f_{i1} \frac{f_{i2}}{\sigma_i^2} y + f_{i1} \frac{f_{i3}}{\sigma_i^2} z + f_{i1} \frac{f_{i4}}{\sigma_i^2} \right) &= 0 \\ x \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i1}^2 \right) + y \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i2} f_{i1} \right) + z \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i3} f_{i1} \right) &= - \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i4} f_{i1} \right) \end{aligned} \quad (\text{B.53})$$

L'équation pour la dérivée par rapport à  $y$  ;

$$x \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i1} f_{i2} \right) + y \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i2}^2 \right) + z \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i3} f_{i2} \right) = - \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i4} f_{i2} \right) \quad (\text{B.54})$$

L'équation pour la dérivée par rapport à  $z$  ;

$$x \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i1} f_{i3} \right) + y \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i2} f_{i3} \right) + z \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i3}^2 \right) = - \sum_{i=1}^4 \left( \frac{1}{\sigma_i^2} f_{i4} f_{i3} \right) \quad (\text{B.55})$$

La solution pour  $M$ , par la méthode de moindre carré, est donnée par le système de trois équations (B.53), (B.54) et (B.55) et par trois inconnues qui sont les coordonnées  $x$ ,  $y$  et  $z$  du point  $M$ .

Nous pouvons représenter le système de façon matricielle ainsi :

$$\begin{pmatrix} \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i1}^2 & \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i2} f_{i1} & \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i3} f_{i1} \\ \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i1} f_{i2} & \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i2}^2 & \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i3} f_{i2} \\ \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i1} f_{i3} & \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i2} f_{i3} & \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i3}^2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} - \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i4} f_{i1} \\ - \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i4} f_{i2} \\ - \sum_{i=1}^4 \frac{1}{\sigma_i^2} f_{i4} f_{i3} \end{pmatrix} \quad (\text{B.56})$$

Ce système d'équations ne dépend que des matrices de projection  $P_g$  et  $P_d$  des caméras et des projections  $B_g$  et  $B_d$  du point  $M$  sur les images gauche et droite de chaque caméra.

La solution du système d'équations représente la position de  $M$ . Le système peut être résolu par n'importe quelle méthode de résolution de systèmes d'équations linéaires.

Une remarque importante porte sur  $\sigma_i$ . Il représente la précision de la caméra : l'erreur que l'on minimise est toujours pondérée par la précision de la caméra, c'est-à-dire que, l'algorithme de minimisation assigne plus de confiance à l'information fournie par la caméra la plus précise.

### B.2.3 Conclusion

Nous avons présenté le processus de calibration d'un système de vision stéréoscopique rigide présentant l'avantage de se calibrer qu'une seule fois au début de chaque séance d'expérimentation.

Nous avons montré aussi la façon de changer le repère du système de vision. Cela devient important au moment de repérer des points dans l'espace car, ils sont ainsi repérés dans des repères plus adéquats pour nos problèmes. Dans le cas particulier de notre système de vision, nous avons choisi de le placer au même endroit que le repère de la caméra gauche. Cela nous permet d'avoir la position des points dans l'espace sur un repère connu. Plus tard l'information pourra être utilisée pour le bras.

Dans un deuxième temps, nous avons développé un procédé probabiliste de localisation d'un point dans l'espace, il prend en compte la précision des caméras pour inférer cette position.

Dans l'annexe suivante nous allons utiliser le processus de calcul de la position d'un point pour inférer la transformation géométrique entre les robots. Pour cela, nous allons utiliser la description géométrique de la pince.

### **B.3 Calibration du système multirobots**

Dans les sections précédentes, nous avons décrit comment utiliser un système de vision stéréoscopique et nous avons construit un dispositif fournissant la position d'un objet. Nous voulons maintenant commander un bras manipulateur à partir de cette information.

Le capteur stéréoscopique étant placé sur le robot cartésien, nous devons trouver la relation géométrique existante entre le bras manipulateur et le robot cartésien. De cette façon, nous pourrions interpréter l'information fournie par le capteur stéréoscopique pour référencer les objets sur le repère du bras. L'idée est de prendre des objets avec le bras robot, lesquels sont repérés avec la vision stéréoscopique. Étant donné que la vision stéréoscopique repère les objets dans son système de coordonnées, nous devons avoir un moyen de les référencer dans le repère du bras.

La *calibration du système multirobots* consiste à trouver la relation géométrique entre les deux robots. Il s'agit d'un processus d'étalonnage similaire au processus de la localisation d'un objet à partir de la vision stéréoscopique présenté dans la section 4.2.

#### **B.3.1 Description du problème de la calibration multirobots**

Nous abordons le problème de la calibration multirobots comme le problème de localisation d'un objet avec le système de vision stéréoscopique car, nous avons pratiquement la même information ; à savoir :

1. le repère de la pince.
2. la description géométrique de la pince.
3. le repère du système de vision stéréoscopique.
4. l'information sur les points de référence sur la pince fournie par le système stéréoscopique.

Nous ajoutons aux informations précédentes l'information concernant les configurations des robots. C'est-à-dire que pour le cas des points de référence de la pince, il doivent être référencés dans le repère du bras robot. Cela implique la connaissance de la géométrie du bras robot (section 3.2.1). Pour le cas du robot cartésien, l'information sur le système de vision doit être amenée au repère du robot cartésien impliquant sa géométrie (section 3.1.1).

En se basant sur le raisonnement précédent, nous pouvons établir la démarche nécessaire pour résoudre le problème. Elle se compose des étapes suivantes :

1. fixer la scène, c'est-à-dire que les configurations des robots ne doivent pas changer au moment de la prise d'informations,



2. prendre une paire d'images de la pince avec le système de vision stéréoscopique,
3. établir à la main la correspondance entre les deux images pour les points de référence de la pince,
4. calculer avec le système de vision les positions des points de référence,
5. construire la partie du graphe correspondant aux points de référence et impliquant la configuration du robot cartésien,
6. à partir de la description de la pince, construire la partie du graphe correspondant, impliquant la configuration du bras,
7. compléter le graphe cinématique avec l'arc représentant la position entre les robots (il représente la solution au problème),
8. faire l'inférence.

Dans le paragraphe suivant, nous décrivons la construction du graphe cinématique.

### B.3.2 Graphe cinématique pour la calibration multirobots

La spécification du problème est liée à la géométrie des dispositifs impliqués. Elle se fait au moyen de graphes cinématiques représentant leurs relations géométriques.

La construction du graphe cinématique du problème prend en compte :

- le graphe du bras (voir section 3.2.1),
- le graphe du robot cartésien (voir section 3.1.1),
- le graphe de la description de la pince (voir section 4.2.2),
- le graphe construit grâce à l'information sur la pince fournie par la paire stéréoscopique (voir section 4.2.2).

La figure B.14 représente le graphe du problème de calibration multirobots avec :

- l'ensemble de sommets,

$$\{RX_0, \dots, RX_5, Pince, P_0, \dots, P_{n-1}, RC_0, \dots, RC_2, VS\}$$

on en compte : 7 pour le bras, 4 pour le robot cartésien et  $n$  pour les points de référence sur la pince. Cela fait  $n + 11$  sommets,

- l'ensemble d'arcs,

$$\{Q_0, \dots, Q_{9+2n}\}$$

on en compte : 6 pour le bras, 3 pour le robot cartésien,  $2n$  pour les points de référence sur la pince et 1 pour l'arc entre les robots. Cela fait  $2n + 10$  arcs.

Le tableau (B.1) contient l'information sur les arcs du problème, où les arcs :

- $Q_1..Q_6$  représentent les transformations géométriques du bras,
- $Q_7..Q_9$  représentent les transformations géométriques du robot cartésien,
- $Q_{10}, Q_{12}, \dots, Q_{8+2n}$  représentent les points de référence sur la pince,

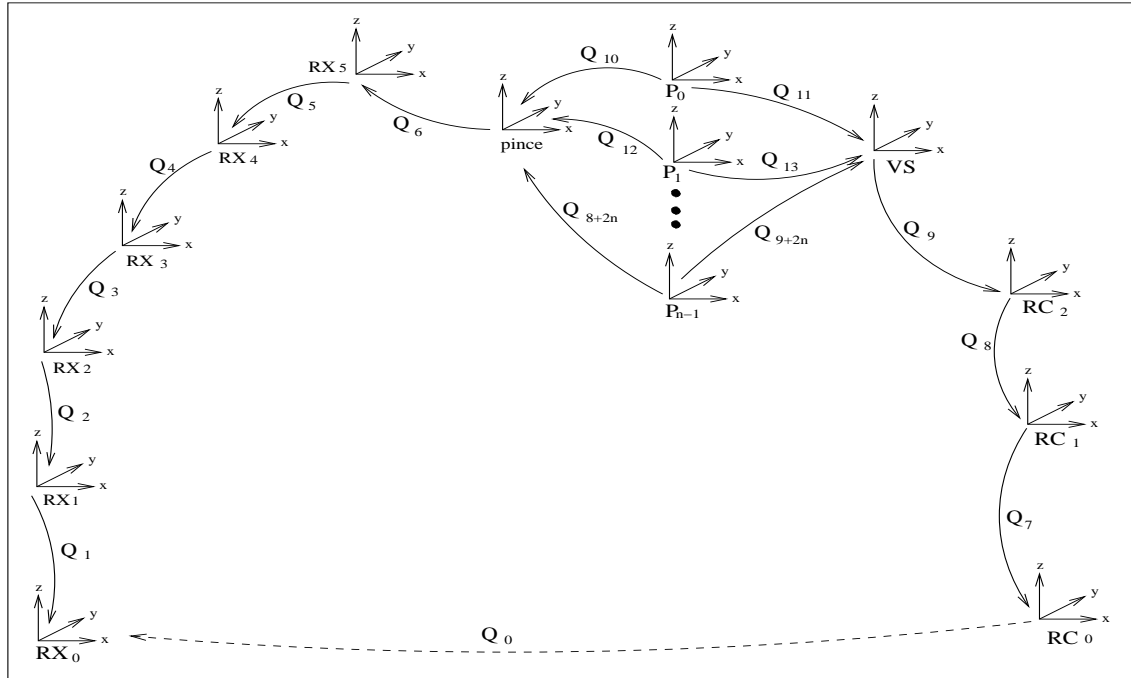


FIG. B.14 – Graphe pour la calibration du système multirobots

- $Q_{11}, Q_{13}, \dots, Q_{9+2n}$  représentent les positions de points de référence sur la vision stéréoscopique et,
- $Q_0$  représentant la transformation géométrique entre le robot cartésien et le bras. Cette transformation représente la solution au problème de calibration multirobots.

Comme pour le problème de la localisation avec la vision stéréoscopique, les paramètres représentant des variables sont classés avec :

- $L$ , si leurs valeurs sont représentées avec des distributions de probabilité,
- $X$ , si nous voulons connaître leurs valeurs.

L'existence de plusieurs cycles dans le graphe du problème (figure B.14) établit des contraintes globales qu'il faut respecter. Le problème est géométriquement bien posé si les cycles sont fermés.

Dans la section 4.2.2 sur la localisation stéréoscopique, nous avons appris à calculer le nombre minimal de cycles :

$$\begin{aligned} \text{Nombre\_minimal\_de\_cycles} &= 2n + 10 - (n + 11) + 1 \\ &= n \end{aligned}$$

Étant donné que chaque cycle doit contenir le plus grand nombre d'arcs avec des inconnues

<b>Arcs</b>	<b>Distribution de (<math>x, y, z, r_x, r_y, r_z</math>)</b>	<b>Type de paramètre</b>
$Q_0$	<i>Uniforme</i> [ <i>Inf, Sup</i> ]	( $X, X, X, X, X, X$ )
$Q_1$	( $0, 0, 0, 0, 0, q_0$ ) = <i>Normale</i> ( $q_0, \sigma_{q_0}$ )	( $F, F, F, F, F, L$ )
$Q_2$	( $0, 0, 0, q_1, 0, 0$ ) = <i>Normale</i> ( $q_1, \sigma_{q_1}$ )	( $F, F, F, L, F, F$ )
$Q_3$	( $0, 0, 450, q_2, 0, 0$ ) = <i>Normale</i> ( $q_2, \sigma_{q_2}$ )	( $F, F, F, L, F, F$ )
$Q_4$	( $0, 0, 0, 0, 0, q_3$ ) = <i>Normale</i> ( $q_3, \sigma_{q_3}$ )	( $F, F, F, F, F, L$ )
$Q_5$	( $0, 0, 450, q_4, 0, 0$ ) = <i>Normale</i> ( $q_4, \sigma_{q_4}$ )	( $F, F, F, L, F, F$ )
$Q_6$	( $0, 0, 85, 0, 0, q_5$ ) = <i>Normale</i> ( $q_5, \sigma_{q_5}$ )	( $F, F, F, F, F, L$ )
$Q_7$	( $x, y, z, 0, 0, 0$ ) = <i>Normale</i> ( $x, \sigma_{Q_7.x}$ ) <i>Normale</i> ( $y, \sigma_{Q_7.y}$ ) <i>Normale</i> ( $z, \sigma_{Q_7.z}$ )	( $L, L, L, F, F, F$ )
$Q_8$	( $0, 0, 0, 0, 0, \theta$ ) = <i>Normale</i> ( $\theta, \sigma_{Q_7.\theta}$ )	( $F, F, F, F, F, L$ )
$Q_9$	( $0, 0, 0, \alpha, 0, 0$ ) = <i>Normale</i> ( $\alpha, \sigma_\alpha$ )	( $F, F, F, L, F, F$ )
$Q_{10}$	( $x_0, y_0, z_0, -, -, -$ )	( $F, F, F, -, -, -$ )
$Q_{11}$	( $VS.x_0, VS.y_0, VS.z_0, -, -, -$ ) = <i>Normale</i> ( $VS.x_0, \sigma_{VS.x}$ ) <i>Normale</i> ( $VS.y_0, \sigma_{VS.y}$ ) <i>Normale</i> ( $VS.z_0, \sigma_{VS.z}$ )	( $L, L, L, -, -, -$ )
$\vdots$	$\vdots$	$\vdots$
$Q_{8+2n}$	( $x_{n-1}, y_{n-1}, z_{n-1}, -, -, -$ )	( $F, F, F, -, -, -$ )
$Q_{9+2n}$	( $VS.x_{n-1}, VS.y_{n-1}, VS.z_{n-1}, -, -, -$ ) = <i>Normale</i> ( $VS.x_{n-1}, \sigma_{VS.x}$ ) <i>Normale</i> ( $VS.y_{n-1}, \sigma_{VS.y}$ ) <i>Normale</i> ( $VS.z_{n-1}, \sigma_{VS.z}$ )	( $L, L, L, -, -, -$ )

TAB. B.1 – Les arcs du graphe pour la calibration du système multirobots.

(variables  $X$ ). Les cycles respectant cette contrainte ont la structure,

$$\{Q_0, \dots, Q_9, \dots, Q_{10+2i}, Q_{11+2i}\}$$

avec  $i = 0 \dots n - 1$  où  $n$  représente le nombre de points de référence sur la pince.

### B.3.3 Variables pour la calibration multirobots

Le graphe cinématique contient l'information nécessaire pour définir les variables impliquées :

1. les variables des vecteurs de pose sont tirées du tableau (B.1). Le type de variable,
  - $X$  implique que c'est une variable que l'on recherche,
  - $L$  implique qu'elle est définie avec une distribution de probabilité,
  - $F$  veut dire qu'il s'agit d'une valeur constante.

Dans tous les cas, les variables sont définies dans l'espace  $\mathbb{R}$ ,

2. les  $n$  variables booléennes de fermeture de cycle,

$$\{K_i\}_{i=0}^{n-1}$$

En prenant en compte tous les variables impliquées, nous définissons la distribution de probabilité conjointe,

$$p(\mathcal{X} \ \mathcal{L} \ K_0 \dots K_{n-1})$$

avec,

- $\mathcal{X}$  représentant l'ensemble de variables inconnues et,
- $\mathcal{L}$  représentant l'ensemble de variables libres.

Nous devons maintenant définir la décomposition permettant de donner un sens précis à cette distribution de probabilité conjointe.

### B.3.4 Décomposition pour le système multirobots

La décomposition choisie représente la connaissance que l'on a du problème, elle prend en compte les contraintes qu'il faut respecter.

Pour le choix de la décomposition, on fait les hypothèses suivantes :

1. étant donné que les arcs ne sont liés que par les contraintes de fermeture des cycles, fait qui garantit la cohérence du modèle. Nous pouvons dire que les arcs sont mutuellement indépendants,
2. les éléments des arcs sont mutuellement indépendants en sachant qu'ils ne sont liés que par leur appartenance au même arc,
3. des deux hypothèses antérieures, nous pouvons établir que les éléments des arcs en général sont mutuellement indépendants, en conséquence, les variables libres sont mutuellement indépendantes entre elles et les variables inconnues sont aussi mutuellement indépendantes entre elles,
4. les variables pour les conditions de fermeture de cycles sont mutuellement indépendantes en fonction de la connaissance que l'on a sur les variables libres et inconnues du même cycle.

La décomposition choisie en prenant en compte ces hypothèses est :

$$p(\mathcal{L} \mathcal{X} K_0 \dots K_{n-1}) = p(\mathcal{L}) p(\mathcal{X}) \prod_{i=0}^{n-1} p(K_i | \mathcal{X} \mathcal{L}) \quad (\text{B.57})$$

avec,

$$p(\mathcal{L}) = \prod_{j=0}^{|\mathcal{L}|-1} p(l^{(j)})$$

où  $|\mathcal{L}|$  représente le cardinal de l'ensemble  $\mathcal{L}$  et  $l^{(j)}$  représente la variable libre  $j$ .

$$p(\mathcal{X}) = \prod_{i=0}^{|\mathcal{X}|-1} p(x^{(i)})$$

où  $|\mathcal{X}|$  représente le cardinal de l'ensemble  $\mathcal{X}$  et  $x^{(i)}$  est la variable inconnue  $i$ .

L'expression (B.57) est similaire à l'expression pour la décomposition du problème de la localisation avec la vision stéréoscopique.

### B.3.5 Formes paramétriques pour la calibration multirobots

Le cadre de connaissance sur le problème sera complet dès lors que les facteurs de la décomposition auront une forme paramétrique associée.

#### Éléments des arcs

Le choix de formes paramétriques est lié au problème et aux dispositifs impliqués. En conséquence il y a des formes paramétriques imposées par les dispositifs. C'est le cas pour les formes paramétriques pour les éléments des arcs représentant l'architecture des robots (des articulations ou des longueurs des pièces). Ces formes paramétriques sont décrites dans le tableau (B.1) interprété ainsi :

- les variables libres sont indiquées par un  $L$ ,
- les variables inconnues sont indiquées par un  $X$ ,
- les articulations du bras sont indiquées par  $q_i$  avec  $i = 0 \dots 5$ , ces articulations appartiennent aux arcs  $Q_1, \dots, Q_6$ ,
- les articulations du robot cartésien sont représentées par les arcs  $Q_7, Q_8$  et  $Q_9$ . Les noms des variables pour les articulations de mouvement cartésien sont  $x, y$ , et  $z$ . Le tangage est  $\theta$  et  $\alpha$  est le roulis,
- toutes les variables sont représentées avec des distributions de probabilité.

Exemple des formes paramétriques pour les paramètres des arcs :

1. la première articulation du bras robot est indiquée par la variable  $q_0$ , elle est décrite au moyen d'une distribution de probabilité normale avec une moyenne correspondant à la valeur actuelle  $q_0$  de l'articulation et un écart-type  $\sigma_{q_0}$  représentant la précision sur le positionnement de cette articulation,

2. les variables cherchées sont toutes dans l'arc  $Q_0$ . Chacune a une distribution de probabilité uniforme indiquant notre manque d'*a priori* sur sa valeur. Par ailleurs, la distribution de probabilité uniforme indique les intervalles pour les valeurs de la variable,
3. les paramètres labélisés  $F$  sont considérés constants.

### Contraintes de fermeture

Elles sont des variables booléennes représentant les contraintes de fermeture des cycles du graphe cinématique.

Nous pouvons construire les cycles de façon automatique en considérant les restrictions suivantes :

1. les cycles doivent contenir le plus grand nombre d'arcs avec des paramètres inconnus. Dans ce cas, un seul arc contient tous les paramètres inconnus, alors cet arc doit impérativement être présent dans tous les cycles,
2. un arc ne peut être présent dans un cycle qu'une seule fois,
3. les arcs peuvent être parcourus dans n'importe quel sens,
4. tant que possible, il faut minimiser le nombre d'arcs dans un cycle.

En prenant en compte les restrictions antérieures, la définition de chaque variable de fermeture a la forme :

$$p([K_i = \text{vrai}] | \mathcal{X} \mathcal{L}) = \begin{cases} 1 & \text{si } I = T_{Q_{11+2i}}^{-1} * \dots * T_{Q_7}^{-1} * T_{Q_0}^{-1} * T_{Q_1} * \dots * T_{Q_6} * T_{Q_{9+2i}} \end{cases} \text{(B.58)}$$

avec,

- $i = 0 \dots n - 1$  où,  $n$  est le nombre de points de référence,
- $T$  est une matrice de transformation homogène correspondant à l'arc  $Q$ ,
- $T^{-1}$  est l'inverse de la matrice de transformation homogène  $T$ , cela veut dire que le parcours de l'arc est à l'inverse du sens indiqué dans la figure B.14.

Ce cadre de connaissance est nommé  $c_{mr}$ . Puisque le cadre de connaissance est complet, nous sommes prêt pour faire l'inférence probabiliste.

### B.3.6 Inférence probabiliste pour la calibration multirobots

Le but est de trouver la position du robot cartésien par rapport au bras robot. Cette transformation permet au bras robot d'interpréter l'information fournie par la vision stéréoscopique. Pour formuler le problème, nous posons dans notre cadre de connaissance la question,

*Quelle est la probabilité de la position du robot cartésien dans le repère du bras sachant les configurations des robots, la description de la pince et la position des points de référence sur la*

vision stéréoscopique ?

Cette question s'écrit,

$$p(\mathcal{X} \mid [K_0 = \text{vrai}] \dots [K_{n-1} = \text{vrai}] c_{mr}), \quad (\text{B.59})$$

car les cycles cinématiques impliquent : la connaissance des configurations des robots, la description de la pince et la connaissance de la position de points de référence sur le repère de la vision stéréoscopique.

En utilisant la distribution de probabilité conjointe pour la calibration multirobots et en appliquant le théorème de Bayes et la règle de la marginalisation, on obtient :

$$\begin{aligned} p(\mathcal{X} \mid [K_0 = \text{vrai}] \dots [K_{n-1} = \text{vrai}] c_{mr}) &\propto p(\mathcal{X} \mid [K_0 = \text{vrai}] \dots [K_{n-1} = \text{vrai}]) \\ &\propto \int_{\mathcal{L}} p(\mathcal{L} \mid \mathcal{X} \mid [K_0 = \text{vrai}] \dots [K_{n-1} = \text{vrai}]) d\mathcal{L} \end{aligned} \quad (\text{B.60})$$

en utilisant l'expression (B.57) de la décomposition,

$$p(\mathcal{X} \mid [K_0 = \text{vrai}] \dots [K_{n-1} = \text{vrai}] c_{mr}) \propto p(\mathcal{X}) \int_{\mathcal{L}} p(\mathcal{L}) \prod_{i=0}^{n-1} p([K_i = \text{vrai}] \mid \mathcal{X} \mathcal{L}) d\mathcal{L} \quad (\text{B.61})$$

Étant donnée la taille de l'espace d'intégration, on le réduit en exploitant les contraintes de fermeture (B.58). Ainsi, l'un des arcs de chaque cycle cinématique sera calculé en fonction des autres arcs du même cycle.

$$T_{Q_{11+2i}} = T_{Q_9}^{-1} * \dots * T_{Q_7}^{-1} * T_{Q_0}^{-1} * T_{Q_1} * \dots * T_{Q_6} * T_{Q_{10+2i}},$$

où l'arc  $Q_{11+2i}$  représentant la position du point de référence  $i$  sur le repère de la vision stéréoscopique est calculé en fonction des autres arcs du même cycle.

Pour indiquer le fait que l'arc  $Q_{11+2i}$  est calculé en utilisant les autres arcs du cycle auquel il appartient, nous le représentons avec  $O_i$ ,

$$T_{O_i} = T_{Q_9}^{-1} * \dots * T_{Q_7}^{-1} * T_{Q_0}^{-1} * T_{Q_1} * \dots * T_{Q_6} * T_{Q_{10+2i}}, \quad (\text{B.62})$$

indiquant le fait que  $O_i$  est fonction des inconnues (éléments de l'arc  $Q_0$ ) et des variables libres (éléments des arcs du même cycle cinématique étiquetés  $L$ ). La forme paramétrique de l'arc  $O_i$ ,

$$p(O_i \mid \mathcal{X} \mathcal{L}), \quad (\text{B.63})$$

a les mêmes paramètres que celle correspondante à l'arc  $Q_{11+2i}$  cependant,  $O_i$  est fonction des paramètres libres et inconnues se trouvant dans le cycle cinématique auquel il appartient.

En exploitant les contraintes de fermeture, c'est-à-dire, en utilisant (B.62) on peut réécrire la question (B.61),

$$p(\mathcal{X} | [K_0 = vrai] \dots [K_{n-1} = vrai] c_{mr}) \propto p(\mathcal{X}) \int_{\mathcal{L}} p(\mathcal{L}) \prod_{i=0}^{n-1} p(O_i | \mathcal{X} \mathcal{L}) d\mathcal{L}. \quad (\text{B.64})$$

cette expression est moins lourde à calculer de point de vue de la méthode d'intégration de Monte-Carlo. Elle représente l'espace solution pour la relation géométrique entre les robots.

### B.3.7 Solution pour la calibration multirobots

Pour la calibration multirobots, nous choisissons la solution maximisant la distribution de probabilité représentant l'espace solution,

$$\max_{\mathcal{X}} p(\mathcal{X} | [K_0 = vrai] \dots [K_{n-1} = vrai] c_{mr}) = \max_{\mathcal{X}} p(\mathcal{X}) \int_{\mathcal{L}} p(\mathcal{L}) \prod_{i=0}^{n-1} p(O_i | \mathcal{X} \mathcal{L}) d\mathcal{L}.$$

cette expression a deux opérations imbriquées qui ne sont pas faciles à résoudre : l'intégration et la maximisation. Nous allons les aborder de façon indépendante.

#### B.3.7.1 Intégrale

Dans le problème de la calibration multirobots, comme les problèmes de CAO-Robotique, on trouve des nombreuses transformations (vecteurs de pose) donc l'intégration doit se faire dans des espaces de grande dimension car, dans ces vecteurs de pose se trouvent les variables libres définissant l'espace d'intégration.

Nous voulons résoudre,

$$\int_{\mathcal{L}} p(\mathcal{L}) \prod_{i=0}^{n-1} p(O_i | \mathcal{X} \mathcal{L}) d\mathcal{L}, \quad (\text{B.65})$$

définie dans l'espace décrit par  $p(\mathcal{L})$ . Espace de grand dimension car, le nombre de variables libres est grand. De plus, il faut respecter les contraintes générées par les distributions sur les variables libres. En conséquence, on a une intégrale difficile à résoudre et qui n'a pas en général de solution analytique. Cependant il y a plusieurs méthodes numériques pouvant fournir une bonne approximation.

En prenant en compte toutes les contraintes et caractéristiques du problème, Nous utilisons les *méthodes de Monte-Carlo* pour faire l'intégrale, car elles sont bien adaptées aux problème de CAO-Robotique comme le nôtre.



Les méthodes de Monte-Carlo sont utilisées dans plusieurs applications de localisation (voir par exemple [DFBT99, TFB00]) et de robotique [RBF<sup>+</sup>00]. Elles sont décrites dans plusieurs documents [Mek99, DGdF00].

La méthode d'intégration de Monte-Carlo que l'on a utilisée pour approcher cette intégrale est complètement décrite dans l'annexe E. Elle nous amène à estimer l'intégrale avec l'expression :

$$\int_{\mathcal{L}} p(\mathcal{L}) \prod_{i=0}^{n-1} p(O_i | \mathcal{X} \mathcal{L}) d_{\mathcal{L}} \propto \frac{1}{N} \sum_{r=0}^{N-1} \left( \prod_{i=0}^{n-1} p(O_i | \mathcal{X} \mathcal{L}^{\{r\}}) \right).$$

Pour la calculer nous devons générer  $N$  ensembles de valeurs de  $\mathcal{L}$  par des tirages aléatoires en suivant la distribution de probabilité de chacun des paramètres libres. Nous pouvons le faire car nous avons posé une l'hypothèse d'indépendance entre eux.

L'expression  $\mathcal{L}^{\{r\}}$  représente le  $r^{\text{ème}}$  ensemble de valeurs de  $\mathcal{L}$ .

Le tirage aléatoire est en général difficile à faire, cependant dans le cas de la calibration multirobots, les formes paramétriques sont des distributions de probabilité normales et uniformes. Nous savons faire des tirages aléatoires sur ces fonctions (voir annexe D).

### B.3.7.2 Maximisation

Le processus de maximisation utilisé a été décrit dans la section 4.2.7.1. Cependant cette fois-ci, il devient plus lourd à cause de l'intégrale imbriquée : l'évaluation de chaque individu implique d'approximer cette intégrale (eqn. B.65). Pour évaluer chaque individu de la population, il faut approcher une intégrale et, ceci pour toute la population et pour chaque itération de l'algorithme génétique. En conséquence, le temps de calcul augmente cependant, la méthode est bien adaptée pour ce type de problème. En plus la calibration multirobots n'est faite qu'une seule fois au début des expériences et le temps de réponse n'est pas critique dans ce cas.

## B.3.8 Conclusion

Nous avons abordé le problème de la calibration multirobots comme un problème de localisation d'un objet par la vision stéréoscopique impliquant une chaîne cinématique plus grande. Nous avons établi les mêmes *a priori* et nous avons considéré des incertitudes sur les articulations des robots impliqués. Par conséquent, le nombre de variables augmente et il y a des incertitudes que les variables ont associées. Alors le problème est plus difficile.

Comme pour le problème de localisation d'un objet décrit dans la section 4.2. La calibration multirobots a une unique solution que nous calculons en maximisant la distribution décrite par la question probabiliste posant le problème. Au contraire du problème précédent, cette fois

ils existent des intégrales imbriquées dans le processus de maximisation. Pour approximer ces intégrales, on a utilisé la méthode de Monte-Carlo.

Toute information fournie par le système de vision est calculée sur le repère propre au bras en utilisant la transformation représentant la calibration multirobots.

## B.4 Expérimentation

La mise au point du système avec le but de réaliser une tâche de “pick and place implique plusieurs processus de calibration, nous les citons et présentons quelques expériences.

### Calibration du système stéréoscopique

Cette calibration est faite au début des expériences et reste invariant pendant les expériences. Elle permet d’inférer la position d’un point dans l’espace à partir de ses projections sur les caméras du système de vision stéréoscopique.

### Calibration multirobots

La relation géométrique existante entre les deux robots (robot cartésien et bras manipulateur) est inférée avec le processus décrit dans la section B.3.

### Localisation d’un objet

Ce processus correspond à un processus de calibration, il est utilisé pour repérer les objets qui seront pris par le bras robot. il a été décrit dans la section 4.2. Nous montrons quelques résultats.

point	x	y	z
1	28.0	44.0	15.0
2	1.0	44.0	15.0
3	-30.0	44.0	15.0
4	37.0	44.0	35.0
5	0.0	44.0	30.5
6	1.0	44.0	66.0

TAB. B.2 – Description de l’objet.

La figure B.15 montre l’objet de nos expériences, le tableau (B.2) contient sa description (les coordonnées des points de référence) tandis que le tableau (B.3) présente le résultat de l’inférence sur sa position. Cette position est référencée dans le repère du bras (voir tableau B.4) pour qu’il soit capable d’aller le saisir.

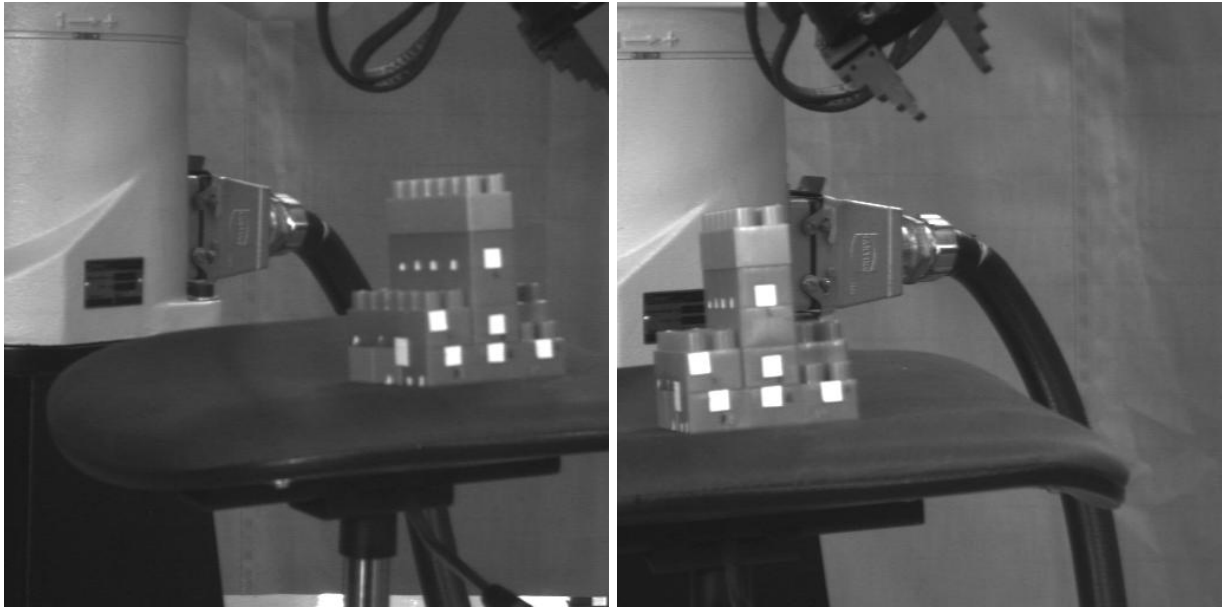


FIG. B.15 – Objet vu par la paire stéréoscopique

<b>x</b>	<b>y</b>	<b>z</b>	<b>r<sub>x</sub></b>	<b>r<sub>y</sub></b>	<b>r<sub>z</sub></b>
53	119	1216	-0.611	-2.41	2.35

TAB. B.3 – Position de l'objet fournie par la V.S.

<b>x</b>	<b>y</b>	<b>z</b>	<b>r<sub>x</sub></b>	<b>r<sub>y</sub></b>	<b>r<sub>z</sub></b>
713	-339	-316	0.074	-0.091	-1.956

TAB. B.4 – Position de l'objet sur le repère du bras.

### B.4.1 Conclusion

Le calcul de la calibration multirobots est un processus difficile prenant en moyenne environ 60 secondes. Nous avons constaté l'importance des *a priori* sur les inconnues car en réduisant leurs plages de valeurs, le temps de calcul de ces processus est réduit.

Les résultats du processus de calibration multirobots sont très satisfaisants car la calibration est faite en n'utilisant que le système de vision et la description de la pince. Cela a montré des avantages en particulier au moment où le bras robot a été déplacé car, on a refait la calibration tout de suite.

Le temps de calcul de la position d'un objet est un processus similaire au processus précédent.

Il prend en moyenne environ 20 secondes posant des problèmes pour le calcul de la position d'un objet en mouvement. Cependant, le temps reste dans des limites raisonnables pour réaliser la prise de l'objet. Étant donnée la rapide évolution des ordinateurs, ce temps pourrait être réduit par un facteur de 50 à moyen terme.

## B.5 Conclusion

Dans cette annexe, nous avons proposé une méthode probabiliste pour calculer la position du robot cartésien par rapport au bras (la calibration multirobots), à partir d'un ensemble de points caractérisant la pince du bras dans le repère des caméras. Pour cela, nous avons défini trois programmes probabilistes :

- un premier programme probabiliste permet de calibrer chaque caméra. Il infère la valeur de la matrice de projection à partir d'un ensemble de points et de leur projection sur l'image ;
- un deuxième programme probabiliste fournit la position d'un point dans l'espace à partir de sa projection sur l'image de chaque caméra et de la matrice de projection de chaque caméra ;
- un troisième programme probabiliste calcule la transformation permettant de passer du repère du système stéréoscopique au repère du bras manipulateur. Grâce à ce changement de repère, le bras robot peut utiliser l'information du système stéréoscopique pour déterminer la position des objets à saisir. Ce programme probabiliste résout le problème d'inversion géométrique consistant à calculer la position du robot cartésien par rapport au bras en utilisant le système de vision stéréoscopique à partir d'un ensemble de points caractérisant la pince du bras. L'intérêt majeur de l'approche est la prise en compte des incertitudes liées à la précision des caméras du système. D'autre part, le problème de calibration multirobots a une solution unique qui se trouve généralement dans une petite zone d'un espace de grande dimension. Par conséquent, la recherche de cette solution est réalisée en utilisant un algorithme génétique couplé avec des concepts de température.

Ces programmes probabilistes ont apporté une solution originale au problème de calibration multirobot.

De plus, nous constatons que ces programmes probabilistes sont en fait des processus de calibration. En effet, dans chaque programme, le but est de trouver une transformation permettant de passer d'un espace (ou repère) à un autre. Nous avons vu que dans tous ces cas, le problème est représenté en utilisant un graphe. C'est toujours le cas même pour la calibration d'une caméra où, les points dans l'espace, leurs projections et la matrice de projections forment un graphe. Grâce à cette représentation standard du problème de calibration, nous suivons toujours la même méthodologie de description du problème. C'est-à-dire que les nœuds du graphe représentent des repères géométriques tandis que les arcs représentent des transformations entre ces repères. La présence de cycles, dans le graphe représentant le problème, ajoutent des contraintes globales qui nous amènent à de meilleures solutions. Ainsi, en fonction de la taille du problème, il est nécessaire d'augmenter le nombre de cycles (par exemple, il faut avoir plus des points

caractéristiques sur la pince). Malheureusement, le temps de calcul augmente en fonction du nombre de variables et de plus dans tous les cas, l'amélioration est limitée.

Enfin, pour des problèmes de calibration, la solution la plus adéquate est celle maximisant la distribution de probabilité générée par la question. Pour effectuer cette recherche de maximum, nous avons adopté une approche adaptée à chaque problème de calibration : méthode des moindres carrés ou algorithmes génétiques.

D'autre part, ces programmes ont permis de mettre en évidence trois propriétés importantes de la programmation probabiliste :

1. la capacité d'exprimer sous forme probabiliste des problèmes classiques, et de montrer qu'ils sont une généralisation des techniques classiques. C'est le cas, par exemple, de la calibration d'une caméra pour lequel nous avons montré que le programme probabiliste défini, était équivalent à la méthode classique de calibration dans le cas particulier où les formes paramétriques sont des gaussiennes et l'inférence est faite par la méthode des moindres carrés ;
2. la capacité de réutiliser le résultat d'une question dans un autre programme probabiliste. C'est le cas, tout au long de cette annexe, où chaque programme réutilise les résultats du programme précédent. Par exemple, la calibration multirobots qui utilise les points sur la pince fournis par la vision stéréoscopique.
3. la capacité de fusionner des formes paramétriques et de les inverser lors de l'inférence. Prenons par exemple, le cas de la calibration d'une caméra. La décomposition est la suivante :

$$p(P M_1 \dots M_n B_1 \dots B_n) = p(P) \prod_{i=1}^n p(M_i) p(B_i | P M_i). \quad (\text{B.66})$$

Cette décomposition est intéressante dans la mesure où elle permet d'exprimer d'une part nos *a priori* sur la matrice de projection et sur la localisation des points  $M_i$  dans l'espace, et d'autre part, le produit des distributions décrivant la distribution du point  $B_i$  (qui est la projection du point  $M_i$  dans l'image) connaissant la matrice de projection et de la localisation du point  $M_i$  dans l'espace. Ce produit peut être vu comme une manière de fusionner les différentes informations provenant des points  $B_i$ .

# Annexe C

## Traitement de la couleur

### C.1 Algorithme de trAVIS pour aller de RGB vers HSV

On décrit l'algorithme de trAVIS utilisé pour passer de l'espace de couleur *RGB* vers l'espace de couleur *HSV*.

Des valeurs *RGB*, trouver le maximum (*max*) et le minimum (*min*).

```
V = max/255.;
if( max != 0 ) S = (max-min)/max;

if( S==0 ) H = INDEFINI;
else
{
    r1 = (max-R) / (max-min);
    g1 = (max-G) / (max-min);
    b1 = (max-B) / (max-min);

    if( R==max && G==min )      H = 5+b1;
    else if ( R==max && G!=min ) H = 1-g1;
    else if ( G==max && B==min ) H = 1+r1;
    else if ( G==max && B!=min ) H = 3-b1;
    else if ( R==max          ) H = 3+g1;
    else                       H = 5-r1;

    H = H*60;
}
```

L'expression  $H = H * 60$  convertit à degrés. En conséquence, le domaine de  $H$  est  $[0..359]$  tandis que le domaine de  $S$  et  $V$  est  $[0..1]$ .



# Annexe D

## Méthodes de tirage

### D.1 Tirage direct à partir d'une distribution normale univariée

Il existe plusieurs méthodes pour faire des tirages aléatoires selon une distribution de probabilité normale à partir de nombres aléatoires générés avec une distribution uniforme. Pour avoir plus d'information on peut voir par exemple [Rub81].

Étant donné une équation décrivant la fonction de distribution, il est possible d'obtenir une fonction de transformation, laquelle prend des nombres aléatoires en suivant une distribution uniforme et génère des nombres aléatoires en suivant la fonction de distribution désirée. Une des plus importantes de ces fonctions est connue comme *la transformation de Box-Muller* [BM58]. Elle transforme un ensemble de nombres suivant une loi uniforme en un ensemble de nombres aléatoires suivant une loi normale.

La forme la plus basique de cette transformation s'écrit :

$$\begin{aligned}y_1 &= \cos(2\pi u_2) \sqrt{-2 \ln(u_1)} \\y_2 &= \sin(2\pi u_2) \sqrt{-2 \ln(u_1)}\end{aligned}$$

où :

- $u_1$  et  $u_2$  sont deux nombres aléatoires mutuellement indépendants tirés d'une distribution uniforme avec une plage de valeurs dans l'intervalle  $[0, 1]$ ,
- $y_1$  et  $y_2$  sont les nombres aléatoires générés par la fonction de transformation. Ils suivent une distribution normale avec moyenne  $\mu = 0$  et écart type  $\sigma = 1$ .

Cette transformation a deux problèmes,

1. Lenteur de calcul à cause des appels à la librairie mathématique.
2. Possibilité de problèmes de stabilité quand la valeur de  $x_1$  est proche du zéro.



La forme polaire de la transformation de Box-Muller est plus rapide et numériquement plus robuste. Nous présentons l'algorithme la décrivant,

```
float u1, u2, w, y1, y2;

do {
    u1 = 2.0 * rand() - 1.0;
    u2 = 2.0 * rand() - 1.0;
    w = u1 * u1 + u2 * u2;
} while ( w >= 1.0 );

w = sqrt( -2.0 * ln( w ) / w );

y1 = u1 * w;
y2 = u2 * w;
```

où, *rand()* est la fonction pour obtenir un nombre aléatoire suivant une distribution uniforme dans l'intervalle  $[0, 1]$ .

La forme polaire est plus rapide car elle fait l'équivalent géométrique de sinus et de cosinus sans faire appel à la librairie mathématique.

Pour générer des nombres aléatoires suivant n'importe quelle distribution normale, c'est-à-dire pour n'importe quelle valeur de la moyenne et de l'écart type, il faut prendre deux nombres aléatoires  $y1$  et  $y2$  générés par l'algorithme et faire les transformations suivantes :

$$\begin{aligned} p1 &= \mu + y1 * \sigma \\ p2 &= \mu + y2 * \sigma \end{aligned}$$

en conséquence,  $p1$  et  $p2$  sont deux nombres aléatoires suivant une distribution *Normale*( $\mu, \sigma$ ).

# Annexe E

## Méthode d'intégration de Monte Carlo

Les méthodes de Monte Carlo fournissent des solutions approchées pour divers problèmes mathématiques. Elles peuvent être utilisées avec ou sans contexte probabiliste.

Il y a plusieurs techniques d'intégration avec les méthodes de Monte Carlo. Nous allons décrire celle que l'on a utilisée dans nos diverses expériences.

Nous considérons l'intégrale

$$I = \int_a^b g(x) dx$$

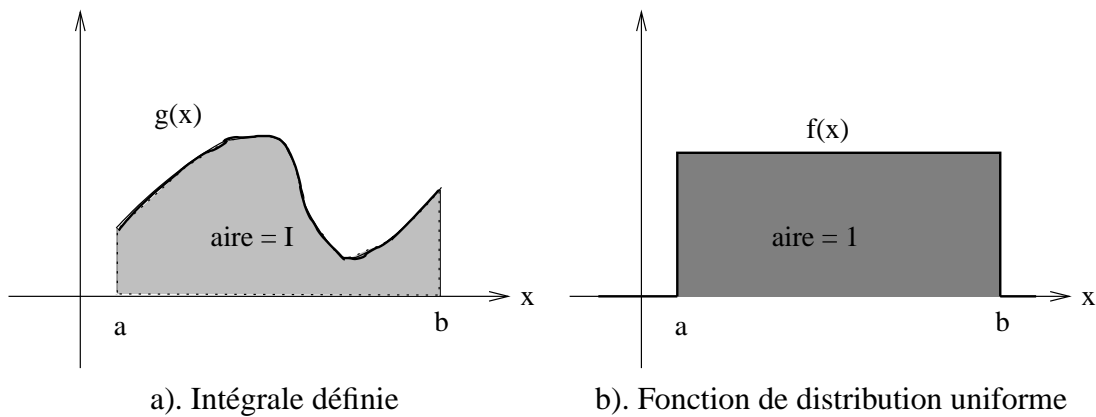


FIG. E.1 – Intégration par Monte Carlo.

où,  $g(x)$  est une fonction avec des valeurs réelles sur le domaine  $(-\infty, +\infty)$ .

La figure E.1a montre une intégrale définie. Nous allons la manipuler de façon qu'il soit possible de la résoudre avec une méthode de Monte Carlo.

Nous définissons la fonction  $f(x)$  sur l'intervalle  $[a, b]$ ,

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b, \\ 0 & \text{sinon.} \end{cases}$$

Elle peut se voir comme une distribution de probabilité uniforme dans l'intervalle  $[a, b]$  (figure E.1b). Nous l'insérons dans l'expression de  $I$ ,

$$I = (b-a) \int_a^b g(x)f(x)dx$$

On sait que l'espérance d'une fonction est définie :

$$E(h(x)) = \int h(x)r(x)dx \quad (\text{E.1})$$

où :

- $h(x)$  est une fonction continue quelconque et,
- $r(x)$  est une distribution de probabilité.

Étant donné que  $f(x)$  est une distribution de probabilité uniforme sur l'espace  $[a, b]$ . Il est possible d'écrire,

$$\begin{aligned} I &= (b-a) \int_a^b g(x)f(x)dx \\ &= (b-a)E(g(x)) \end{aligned}$$

Nous savons aussi que l'espérance d'une fonction peut être approchée par la valeur moyenne de  $N$  exemples de la fonction ;

$$E(h(x)) \approx \frac{1}{N} \sum_{i=0}^{N-1} h(x_i) \quad (\text{E.2})$$

Pour approcher  $E(g(x))$ , il faut faire :

1. Des exemples avec des tirages aléatoires de  $x_i$  en suivant la distribution de probabilité  $f(x)$ .
2. Pour chaque exemple  $x_i$ , on évalue  $g(x_i)$  pour calculer la moyenne ;

$$\hat{g} = \frac{1}{N} \sum_{i=0}^{N-1} g(x_i)$$

La valeur d'espérance de  $g(x)$  dénotée  $E(g(x))$  s'approche ainsi :

$$E(g(x)) \approx \frac{1}{N} \sum_{i=0}^{N-1} g(x_i)$$

Finalement, l'intégrale  $I$  peut être estimée par :

$$\begin{aligned} I &= (b-a)E(g(x)) \\ &\approx (b-a) \left( \frac{1}{N} \sum_{i=0}^{N-1} g(x_i) \right) \end{aligned} \quad (\text{E.3})$$

L'erreur de l'approximation peut être estimée par :

$$\sigma^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (g(x_i) - \bar{g})^2$$

où,

$$\bar{g} = \frac{1}{N} \sum_{i=0}^{N-1} g(x_i)$$

L'erreur de l'approximation de l'intégrale par la méthode de Monte Carlo est  $O(N^{-\frac{1}{2}})$ . Cela veut dire qu'elle ne dépend pas de la dimension de l'espace d'intégration.

Cette méthode d'intégration peut être généralisée pour des espaces multidimensionnels.



# Annexe F

## Algorithme génétique continu

Les algorithmes génétiques se sont inspirés de l'évolution des êtres vivants : il y en a plusieurs types. Une classification dépend du type de données qu'il manipule ; quand les données sont représentées avec des chaînes de 0's et 1's, on parle d'algorithmes génétiques discrets et, si les données sont représentées par des nombres réels, nous les appelons *algorithmes génétiques continus*.

Les algorithmes génétiques sont utilisés pour résoudre des problèmes d'optimisation. Les solutions possibles sont représentées avec les individus d'une population. Un individu est un ensemble de chromosomes dont chacun représente une des multiples caractéristiques de l'individu.

Sur une population, on définit trois opérations :

1. Croisement.
2. Sélection.
3. Mutation.

Nous les décrivons par rapport à notre algorithme génétique de type continu.

**Croisement :** c'est l'opération consistant à combiner deux individus ; *les parents*, pour produire un nouvel individu avec l'idée d'améliorer la descendance, c'est-à-dire que le nouvel individu doit être meilleur que ses parents.

Nous utilisons deux techniques de croisement :

1. Croisement heuristique décrit par les expressions :

$$parent2 + (a) * (parent2 - parent1)$$

$$parent1 + (a) * (parent1 - parent2)$$

où  $a$  est un nombre aléatoire entre 0 et 1.

2. Croisement arithmétique décrit par :

$$(1 - a) * parent1 + (a) * parent2.$$

Pour le choix des parents, nous utilisons deux types de tirages aléatoires :

1. Tirage aléatoire sur une distribution de probabilité uniforme sur l'ensemble de la population.
2. Tirage aléatoire sur la distribution de probabilité de la population. Cette distribution est tirée de la capacité d'adaptation de chaque individu, c'est-à-dire que les individus les plus aptes ont plus de probabilité d'être choisis.

La capacité d'adaptation de l'individu dans notre cas est sa probabilité obtenue de la distribution de probabilité définie par la question du problème qu'on essaye de résoudre.

**Sélection :** elle consiste à choisir les individus de la nouvelle génération. Nous choisissons l'individu le plus performant résultant de chaque couple. Pour évaluer sa performance nous calculons sa capacité d'adaptation (dans notre cas l'adaptation est la valeur de probabilité de la solution correspondant à l'individu). Si le nouvel individu n'est pas meilleur que ses parents, on choisit le meilleur de ses parents, s'il n'avait pas encore été choisi, sinon on refait le croisement.

**Mutation :** c'est un opérateur génétique modifiant un ou plusieurs chromosomes d'un individu, de façon à ce qu'il puisse devenir un individu complètement différent. L'idée est de chercher dans d'autres régions des solutions possibles.

La mutation participe à l'évolution en fonction d'une probabilité définie par l'utilisateur. En général, cette probabilité est faible puisque, si elle est grande, l'algorithme revient à une recherche aléatoire.

**Algorithme génétique.** Étant données les opérations, l'algorithme génétique les applique sur la population de façon itérative dans l'ordre suivant :

1. Croisement pour générer de nouveaux individus.
2. Sélection pour choisir les individus de la nouvelle génération (cette opération est imbriquée dans le processus de croisement).
3. Mutation pour modifier certains individus de la nouvelle population.

À chaque itération la population s'améliore et nous retenons le meilleur individu, de façon qu'à la fin, nous avons un individu représentant une solution du problème.

# Annexe G

## Éléments de Géométrie

### G.1 Transformations géométriques

Nous allons définir deux structures de données qui sont utilisées dans notre travail pour mener les opérations et représentations géométriques :

1. la matrice homogène de transformation.
2. le vecteur de pose.

#### G.1.1 Matrice homogène de transformation

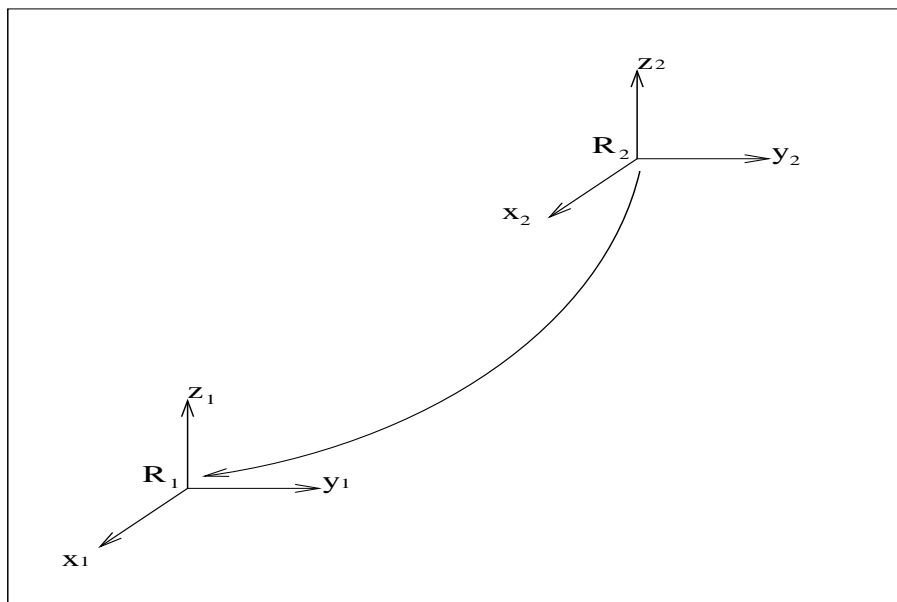


FIG. G.1 – Position relative entre repères



C'est une matrice 4X4 avec la structure suivante :

$$T_{12} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \quad (\text{G.1})$$

où :

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

c'est une matrice 3X3 représentant la rotation du repère  $R_2$  par rapport au repère  $R_1$  et,

$$t = (t_x, t_y, t_z)^{-1}$$

représente la translation entre ces repères (figure G.1).

### G.1.2 Vecteur de pose

Un vecteur de pose  $Q$  est défini dans  $\mathbb{R}^6$  :

$$Q_{12} = (t_x, t_y, t_z, r_x, r_y, r_z)^T \quad (\text{G.2})$$

Ce vecteur est équivalent à la matrice homogène de transformation. Il représente la position relative du repère  $R_2$  par rapport au repère  $R_1$  (figure G.1).

Le vecteur de pose est interprété ainsi :

- Les trois premières coordonnées ;

$$\rho = (t_x, t_y, t_z)^T$$

représentent la translation du repère  $R_2$  par rapport au repère  $R_1$ ,

- les trois dernières coordonnées ;

$$\omega = (r_x, r_y, r_z)^T$$

représentent la rotation existant entre le deuxième repère par rapport au premier.

La rotation est décrite ainsi : les paramètres  $r_x, r_y$  et  $r_z$  représentent le vecteur  $\omega$  de rotation tandis que, la grandeur de l'angle de rotation est donnée par  $\|\omega\|$ .

Il est utile de pouvoir passer de la représentation de matrice homogène à celle de vecteur de pose et réciproquement. En effet la composition de transformations se fait aisément avec les matrices homogènes alors que la spécification des incertitudes est plus naturelle avec les vecteurs de pose.

Ils existent des mécanismes pour passer de la représentation du *vecteur de pose* dans  $\mathbb{R}^6$  à la représentation de *matrice de transformation homogène* et vice-versa.

## Conversion de la représentation de matrice homogène à la représentation de vecteur de pose et vice-versa.

On considère la matrice de transformation,

$$T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

et le vecteur de pose ;

$$Q = \begin{pmatrix} \rho \\ \omega \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \\ t_z \\ r_x \\ r_y \\ r_z \end{pmatrix}$$

### Conversion du 6-vecteur de pose à la matrice homogène de transformation 4X4.

On définit :

$$\mathbf{u} = (u_1 \ u_2 \ u_3)^T = \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$$

$$\{\mathbf{u}\} = \begin{pmatrix} 0 & u_3 & -u_2 \\ -u_3 & 0 & u_1 \\ u_2 & -u_1 & 0 \end{pmatrix}$$

alors, la matrice de transformation est obtenue par :

$$t = \rho$$

$$R = \cos(\|\boldsymbol{\omega}\|)I_3 + (1 - \cos(\|\boldsymbol{\omega}\|))\mathbf{u}\mathbf{u}^T - \sin(\|\boldsymbol{\omega}\|)\{\mathbf{u}\}$$

### Conversion de la matrice homogène de transformation 4X4 au 6-vecteur de pose.

On définit :

$$v = \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}$$

alors :

$$\rho = t$$

$$\boldsymbol{\omega} = \frac{v}{\|v\|} \cos^{-1}(r_{11} + r_{22} + r_{33}).$$



# Bibliographie

- [Ala94] Jarmo T. Alander. An indexed bibliography of genetic algorithms and simulated annealing : Hybrids and comparisons. Technical Report 94-1-SA, University of Vaasa, Vaasa, Finland, 1994. [39](#), [72](#)
- [Bay63] T. Bayes. An essay towards solving a problem in the doctrine of chance. *Philosophical Transactions of the Royal Society of London*, 53 :370–418, 1763. Reprinted in *Biometrika* (1958) 45, pp. 293-315. [14](#)
- [BDL<sup>+</sup>99a] P. Bessière, E. Dedieu, O. Lebeltel, E. Mazer, and K. Mekhnacha. Interprétation ou description (i) : Proposition pour une théorie probabiliste des systèmes cognitifs sensori-moteurs. *Intellectica*, 1999. [8](#), [15](#)
- [BDL<sup>+</sup>99b] P. Bessière, E. Dedieu, O. Lebeltel, E. Mazer, and K. Mekhnacha. Interprétation ou description (ii) : Fondements mathématiques de l’approche f+d. *Intellectica*, 1999. [8](#), [15](#)
- [BEF96] J. Borenstein, B. Everett, and L. Feng. Navigating mobile robots : Systems and techniques. In *A. K. Peters, ltd., Wellesley, MA.*, 1996. [26](#)
- [Ber85] J. Berger. Statistical decision theory and bayesian analysis, 1985. [16](#), [139](#), [140](#)
- [BFHS96] Wolfram Burgard, Dieter Fox, Daniel Hennig, and Timo Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *AAAI/IAAI, Vol. 2*, pages 896–901, 1996. [27](#)
- [BM58] G.P.E. Box and M.E. Muller. A note on the generation of random normal deviates. In *Annals Math. Stat*, volume 29, pages 610–611, 1958. [191](#)
- [Bre90] G. Larry Bretthorst. An introduction of parameter estimation using bayesian probability. in *Maximum Entropy and Bayesian Methods*, pages 53–79, 1990. Kluwer Academic, Fougere (ed.). [14](#)
- [BS94] Jose M. Bernardo and Adrian F. M. Smith. *Bayesian Theory*. John Wiley & Sons, Inc., 1994. [140](#)
- [CB02] C. Coué and P. Bessière. Chasing an elusive target with a mobile robot. In *the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Hawaii, HI (USA), 2002. [15](#)
- [CFBM02] C. Coué, Th. Fraichard, P. Bessière, and E. Mazer. Multi-sensor data fusion using bayesian programming : an automotive application. In *the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne (CH), 2002. [15](#)

- [Cox46] R. T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14 :1–13, 1946. [8](#)
- [Cox61] R. T. Cox. The algebra of probable inference. *Johns Hopkins Univ. Press*, 14 :1–13, 1961. [8](#)
- [Cox78] R. T. Cox. Of inference and inquiry : An essay in inductive logic. In *The Maximum Entropy Formalism*, pages 119–167, Cambridge, Mass, 1978. MIT Press, Levine, R. D. & Tribus, M. (Eds.). [8](#)
- [DBFT99] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, 1999. IEEE. [28](#), [39](#), [72](#)
- [DFBT99] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999. [28](#), [184](#)
- [DGdF00] A. Doucet, N. J. Gordon, and J. F. G. de Freitas. Sequential monte carlo methods in practice. In *forthcoming*, 2000. [39](#), [184](#)
- [Dij59] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Math*, 1 :269–271, 1959. [67](#)
- [DL99] J. Diard and O. Lebeltel. Bayesian learning experiments with a khepera robot. In *Experiments with the Mini-Robot Khepera : Proceedings of the 1st International Khepera Workshop*, Paderborn, HNI-Verlagsschriftenreihe, band 64 ; Germany, 1999. Löffler, Mondada, Rückert (Eds). [15](#)
- [DL00] J. Diard and O. Lebeltel. Bayesian programming and hierarchical learning in robotics. In *SAB2000 Proceedings Supplement Book*, Honolulu, 2000. Publication of the International Society for Adaptive Behavior. [15](#)
- [DLR77] A. Dempster, A. Laird, and D. Rubin. Maximun likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 :Series B, 1977. [79](#), [147](#)
- [DLRR96] M.F. De-La-Rosa-Rosero. *Planification de mouvements avec prise en compte explicite des incertitudes géométriques*. PhD thesis, Institut National Polytechnique de Grenoble, 1996. [x](#), [25](#), [138](#)
- [ECR92] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3) :313–326, June 1992. [149](#)
- [FB91] E. Falkenauer and S. Bouffouix. A genetic algorithm for job shop. In *Proceedings 1991 IEEE International Conference on Robotics and Automation*, number Cat. No. 91CH2969-4 in 1, pages 824–829, Sacramento, CA, 9-11 1991. IEEE Computer Society Press, Los Alamitos, CA. [39](#), [72](#)
- [FBT99] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11 :391–427, 1999. [27](#)

- [FM99] Alexandre R.J. François and Gérard G. Medioni. Adaptive color background modeling for real-time segmentation of video streams. In *the International on Imaging Science, Systems, and Technology*, 1999. 18
- [GM90] M. Gondran and E. Minoux. Graphes et algorithmes, 1990. 67
- [Gre88] John J. Grefenstette. Credit assignment in rule discovery systems based on genetic algorithms. *Machine Learning*, 3 :225, 1988. 72
- [Har94] I. Harvey. The artificial evolution of adaptive behaviour, 1994. 39
- [HM95] Radu Horaud and Olivier Monga. *Vision par ordinateur, Outils fondamentaux*. Editions Hermès, 1995. 149, 150, 152, 155, 156
- [Hol75] John H. Holland. *Adaptation in natural artificial systems*. University of Michigan Press, Ann Arbor, 1975. 39
- [Jay] E.T. Jaynes. Probability theory - the logic of science. inachevé; Disponible à <http://bayes.wustl.edu>. 8, 12
- [JRGB94] Frédéric Jurie, Patrick Rives, J. Gallice, and J.L. Brame. High speed vehicle guidance based on vision. *IFAC Journal in Control Engineering Practice*, 2(2) :289–299, April 1994. 149
- [Kal60] R. Kalman. A new approach to linear filtering and prediction problems, 1960. 27
- [KBM] D. Kortenkamp, R. Bonassi, and R. Murphy. AI-based mobile robots : Case studies of successful robot systems. 26
- [KCK96] L. P. Kaelbling, A. R. Cassandra, and J. A. Kurien. Acting under uncertainty : Discrete bayesian models for mobile-robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996. 27, 28
- [KD99] Wisama Khalil and Etienne Dombre. *Modélisation identification et commande des robots*. Hermes Sciences Publications., 1999. 23, 56
- [KLC98] L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains, 1998. 29
- [KS96] S. Koenig and R. Simmons. Passive distance learning for robot navigation. In *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996. 27
- [Lap95] P.-S. Laplace. Philosophical essay on probabilities, 1995. Springer-Verlag. Originally published 1825. 14
- [LC98] J. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93, 1998. 39
- [LDBM00] O. Lebeltel, J. Diard, P. Bessière, and E. Mazer. A bayesian framework for robotic programming. In *MaxEnt 2000, Twentieth International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, 2000. 15
- [LDH98] G. Lacey and K. M. Dawson-Howe. The application of robotics to a mobility aid for the elderly blind. *Robotics and Autonomous Systems*, 23 :245–252, 1998. 25

- [Leb99] O. Lebeltel. *Programmation Bayésienne des Robots*. PhD thesis, INPG, 1999. [ix](#), [15](#), [31](#), [32](#), [34](#), [37](#), [40](#), [41](#), [84](#)
- [LV00] S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, San Francisco, CA, 2000. IEEE. [28](#)
- [MCSP01] E. Marchand, F. Chaumette, F. Spindler, and M. Perrier. Controlling the manipulator of an underwater roV using a coarse calibrated pan tilt camera. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 2773–2778, Seoul, Korea, 2001. [50](#), [149](#)
- [Mek99] K. Mekhnacha. *Méthodes Probabilistes Bayésiennes pour la prise en compte des Incertitudes Géométriques : Application à la CAO-Robotique*. PhD thesis, INPG, 1999. [ix](#), [35](#), [36](#), [37](#), [39](#), [40](#), [41](#), [63](#), [65](#), [184](#)
- [MG76] Hans Moravec and D. B. Gennery. Cart project progress report. September 1976. NASA contract NASW 2916. [1](#)
- [MK97] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley., New York, 1997. [79](#), [139](#), [147](#)
- [Mon82] George E. Monahan. A survey of partially observable Markov decision processes : Theory, models and algorithms. *Management Science*, 28(1) :1–16, January 1982. [28](#)
- [Muh92] H. Muhlenbein. How genetic algorithms really work, 1992. [39](#), [72](#)
- [Nea93] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993. [39](#)
- [Pap84] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw Hill, 1984. [139](#)
- [PGRKB95] R. Pissard-Gibollet, P. Rives, K. Kapellos, and J.J. Borrelly. Real-time programming of a mobile robot actions using advanced control technics. In *4th Int. Symposium on Experimental Robotics*, Stanford, California, June 30 - July 2 1995. [149](#)
- [Pug89] P. Puget. *Vérification-Correction de programme pour la prise en compte des incertitudes en programmation automatique des robots*. PhD thesis, INPG, 1989. [25](#), [135](#)
- [RBF<sup>+</sup>00] Nicholas Roy, Gregory Baltus, Dieter Fox, Francine Gemperle, Jennifer Goetz, Tad Hirsch, Dimitris Margaritis, Michael Montemerlo, Joelle Pineau, Jamieson Schulte, and Sebastian Thrun. Towards personal service robots for the elderly. In *Workshop on Interactive Robots and Entertainment (WIRE 2000)*, 2000. [25](#), [184](#)
- [RdTG98] Alberto Ruiz, Pedro E. Lopez de Teruel, and M. Carmen Garrido. Probabilistic inference from arbitrary uncertainty using mixtures of factorized generalized gaussians. *Journal of Artificial Intelligence Research*, 9 :167–217, 1998. [25](#), [76](#), [79](#), [139](#)

- [Rub81] R.Y. Rubinstein. *Simulation and the Monte Carlo method*. John Wiley & Sons, 1981. 191
- [Rub88] D. B. Rubin. Using the SIR algorithm to simulate posterior distribution. In *Bayesian Statistics*, Oxford, UK, 1988. Oxford University Press, M. H. Bernardo, K. M. an DeGroot, D. V. Lindley and A.F.M. Smith, editors. 27
- [RW84] R. Redner and H. Walker. Mixture densities, maximum likelihood and the EM algorithm, 1984. *SIAM Rev.*, vol. 26, no. 2, pp. 195–239. 79, 147
- [SB98] R. S. Sutton and A. G. Barto. Reinforcement learning : An introduction. In *MIT Press*, Cambridge, MA, 1998. 28
- [SK95] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environment. In *Proceedings of IJCAI-95*, 1995. 27
- [Smi78] A. R. Smith. Color gamut transformation pairs. *Computer Graphics*, 3(12) :12–19, 1978. 18
- [Son78] E. J. Sondik. The optimal control of partially observable markov processes over the infinite horizon : Discounted costs. *Operations Research*, 26(2) :282–304, 1978. 28
- [SS73] R. W. Smallwood and E. J. Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, 21 :1071–1088, 1973. 28
- [SS00] R. D. Schraft and G. Schmierer. *Service Robots*. A. K. Peters, 2000. 25
- [Tan96] M. A. Tanner. *Tools for Statistical Inference*. Springer-Verlag, New York, 1996. 147
- [TBB<sup>+</sup>00] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Creemers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 2000. 25, 27, 29
- [TFB00] S. Thrun, D. Fox, and W. Burgard. Monte carlo localization with mixture proposal distribution. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, TX, 2000. 28, 39, 184
- [Thr98a] S. Thrun. A framework for programming embedded systems : Initial design and results, 1998. 30
- [Thr98b] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1) :21–71, 1998. 27
- [Thr00] S. Thrun. Probabilistic algorithms in robotics. Technical report, Computer Science Department, Carnegie Mellon University, 2000. 25, 26, 28, 31