



HAL
open science

Planification de trajectoire pour la manipulation d'objets et l'interaction Homme-robot

Xavier Broquère

► **To cite this version:**

Xavier Broquère. Planification de trajectoire pour la manipulation d'objets et l'interaction Homme-robot. Robotique [cs.RO]. Université Paul Sabatier - Toulouse III, 2011. Français. NNT: . tel-00644776

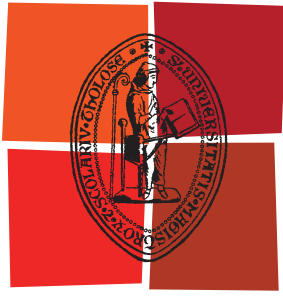
HAL Id: tel-00644776

<https://theses.hal.science/tel-00644776>

Submitted on 25 Nov 2011

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.



Université
de Toulouse

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse III Paul Sabatier (UT3 Paul Sabatier)

Discipline ou spécialité :

Informatique et Robotique

Présentée et soutenue par :

Xavier BROQUERE

le : mardi 5 juillet 2011

Titre :

Planification de trajectoire pour la manipulation d'objets
et l'interaction Homme-robot

Ecole doctorale :

Systèmes (EDSYS)

Unité de recherche :

LAAS-CNRS

Directeur(s) de Thèse :

M. Daniel SIDOBRE

Rapporteurs :

M. Philippe BIDAUD, Professeur des Universités

M. Philippe FRAISSE, Professeur des Universités

Membre(s) du jury :

M. Rachid ALAMI, Président, Directeur de Recherche CNRS

M. Philippe BIDAUD, Professeur des Universités

M. Philippe FRAISSE, Professeur des Universités

M. Michel COURDESSES, Professeur des Universités

M. Adolfo SUAREZ ROOS, EADS France, Innovation Works

M. Daniel SIDOBRE, Maître de Conférence

UNIVERSITÉ TOULOUSE III - PAUL SABATIER
ÉCOLE DOCTORALE SYSTÈMES

THÈSE

en vue de l'obtention du

Doctorat de l'Université de Toulouse
délivré par l'Université Toulouse III Paul Sabatier

Spécialité: Informatique et Robotique

présentée et soutenue publiquement le 5 juillet 2011

Planification de trajectoire pour la manipulation d'objets et l'interaction Homme-robot

Xavier BROQUÈRE

Préparée au Laboratoire d'Analyse et d'Architecture des Systèmes
sous la direction de M. Daniel SIDOBRE

Jury

M. Rachid ALAMI	Président
M. Philippe BIDAUD	Rapporteur
M. Philippe FRAISSE	Rapporteur
M. Michel COURDESSES	Examineur
M. Adolfo SUAREZ ROOS	Examineur
M. Daniel SIDOBRE	Directeur de Thèse

A mes parents

Remerciements

Les travaux présentés dans ce manuscrit ont été effectués au Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS de Toulouse. Je tiens d'abord à remercier M. Raja Chatila, directeur du LAAS-CNRS au début de mon doctorat, pour m'avoir accueilli au LAAS-CNRS. Je remercie ensuite M. Rachid Alami, responsable du groupe RIS (Robotique et InteractionS), de m'avoir permis d'effectuer cette thèse dans ce groupe de recherche et d'avoir présidé mon jury de thèse.

Je tiens à exprimer ma sincère reconnaissance à M. Daniel Sidobre, mon directeur de thèse, pour m'avoir encadré tout au long de mes travaux, pour sa patience, pour ses conseils et pour toutes les heures, parfois tardives, qu'il a pu me consacrer. Mes remerciements vont également à mes rapporteurs M. Philippe Bidaud et M. Philippe Fraisse pour le soin avec lequel ils ont lu ce manuscrit, ainsi que pour la qualité de leurs critiques. Merci aussi à M. Michel Courdresses et à M. Adolfo Suarez-Roos pour avoir accepté d'être membre de mon jury de thèse.

Je remercie également tous les permanents du pôle de Robotique et Intelligence Artificielle, spécialement Thierry Siméon (pour avoir été mon directeur de thèse pendant un an) ; Matthieu Herrb et Anthony Mallet pour leurs réponses à beaucoup de mes questions d'informatique. Je tiens également à remercier tous mes collègues du LAAS que j'ai eu la chance de rencontrer et particulièrement le docteur Mokhtar Gharbi que j'ai dû supporter pendant 3 ans avec ses blagues quotidiennes. J'en oublie sûrement mais merci à tous les doctorants et post-doctorants Panos, Matthieu P., Julien B., Benoît, Jean-Philippe, Akin, Jim, Romain, Sébastien, Samir, Manish, Raquel, Luis, Thierry, Bach Van, Anh, Pancho, Diego, Guido ...

Je remercie Norbert, pour toutes les sorties qui m'ont permis de m'aérer l'esprit pendant ces années... Merci à Emma pour son grand soutien et sa patience pendant la rédaction...

Merci à ma famille pour leur encouragement pendant les périodes difficiles. Je remercie sincèrement ma mère et mon beau-père Jean-Louis pour m'avoir accompagné toutes ces années.

Une dernière pensée à mon père, là haut, je souhaite qu'il soit fier de son fils...

Table des matières

1	Introduction	1
1.1	Contexte de la thèse	3
1.2	Contributions de la thèse	4
1.3	Organisation du manuscrit	4
1.4	Publications associées à cette thèse	4
1.5	Rapports de projet associés à cette thèse	5
2	État de l’art	7
2.1	Le robot manipulateur mobile autonome	8
2.1.1	Quelques robots	8
2.1.2	L’architecture du robot autonome	9
2.1.3	La sécurité	12
2.1.4	L’interaction Homme-robot	13
2.1.5	Des robots sociaux	14
2.2	Les mouvements pour l’interaction Homme-robot	14
2.2.1	Le mouvement des corps	14
2.2.2	Le mouvement des robots	19
2.2.3	La planification de chemin	21
2.2.4	La génération de trajectoire	24
2.3	Conclusion	29
3	La génération de trajectoire sous contraintes cinématiques	31
3.1	Modèle de trajectoire pour l’interaction Homme-robot	32
3.1.1	Notations	32
3.1.2	Des suites de segments de courbes cubiques pour représenter les trajectoires	32
3.1.3	Les contraintes cinématiques	34
3.1.4	Notion de trajectoire en temps minimal	34
3.1.5	Le mouvement dans le plan accélération-vitesse	34
3.1.6	Symétrie du plan accélération-vitesse	36
3.2	Les trajectoires mono-dimensionnelles point à point en temps minimal	37
3.3	Les trajectoires mono-dimensionnelles de type V en temps minimal	40
3.3.1	Les types de mouvements	40
3.3.2	Les types de mouvements composés de moins de sept segments	42

3.3.3	La distance critique : définition et calculs	42
3.3.4	L'algorithme général de planification en temps minimal	48
3.3.5	Exemples d'applications	53
3.3.6	Implémentation et résultats	56
3.4	Les trajectoires mono-dimensionnelles en temps imposé	56
3.4.1	Cas du mouvement point à point	56
3.4.2	Cas du mouvement de type V	57
3.5	Les trajectoires multidimensionnelles en temps minimal	58
3.5.1	Cas du mouvement point à point	59
3.5.2	Cas du mouvement de type V	59
3.6	Conclusion	60
4	Approximation de trajectoire	63
4.1	Définition du problème	63
4.2	Mouvement en temps imposé sans contrainte cinématique	64
4.3	Algorithmes d'approximation proposés	66
4.4	Calculs des erreurs d'approximation	66
4.5	Approximation de trajectoire multidimensionnelle avec un nombre imposé de triplets de segments	67
4.5.1	Application au cas simple du segment de droite	67
4.5.2	Application au cas du cercle	68
4.6	Approximation de trajectoire en fonction d'une erreur maximale imposée	70
4.6.1	Application au cas du segment de droite	71
4.6.2	Application au cas du cercle	72
4.7	Cas des trajectoires non C^2	73
4.7.1	Approximation par un nombre donné d'intervalles	73
4.7.2	Approximation par la méthode limitant l'erreur	76
4.8	Conclusion	76
5	Planification et exécution de mouvement pour l'interaction Homme-robot	77
5.1	Planification du chemin	78
5.1.1	Prise en compte des contraintes géométriques	78
5.1.2	Planification du chemin	80
5.2	Du chemin à la trajectoire	80
5.3	Lissage des mouvements d'arrêt	81
5.3.1	Méthodes de Lloyd et al. et de Macfarlane	81
5.3.2	Méthode de lissage proposée	83
5.4	Application aux robots manipulateurs	85
5.5	Planification dans l'espace cartésien	86
5.5.1	Génération de la trajectoire TR lissée dans l'espace cartésien	86
5.5.2	Conversion de la trajectoire dans l'espace articulaire du robot	87
5.6	Adaptation de la loi de mouvement pendant l'exécution	90

5.7	Conclusion	90
6	Expérimentations	91
6.1	Intégration de la génération de trajectoire sur la plateforme Jido	91
6.1.1	L'architecture logicielle de Jido	91
6.2	Les trajectoires dans la boucle de commande	95
6.3	Approximation de trajectoires : "Écrire et dessiner"	96
6.4	Génération de trajectoire : "Tâche de saisie d'un objet"	98
6.5	Modification de la loi d'évolution de la trajectoire en présence de l'Homme	100
6.6	Conclusion	102
7	Conclusion et perspectives	103
A	Calcul de l'état d'une trajectoire à un instant donné	109
B	Calcul de la distance critique	113
C	Algorithmes de génération de trajectoires mono-dimensionnelles (suite)	121
D	Calcul de la durée d'un mouvement point à point	127
	Références bibliographiques	129

1

Introduction

De l'appareil le plus simple aux machines humanoïdes, en passant par l'indexation de pages web, le terme robot regroupe beaucoup de choses. Il a été employé pour la première fois par le célèbre dramaturge tchèque *Karel Čapek* (1890-1938) dans la pièce de théâtre *R. U. R. (Rossum's Universal Robots)* de 1921. Ce mot dérive du mot tchèque "robota" qui signifie la corvée. Bien que le premier robot industriel soit apparu en 1954 avec le dépôt d'un brevet par George Devol, la première application, développée par *Unimation Inc.*, n'a vu le jour qu'en 1961 dans l'usine de Général Motors pour extraire des pièces d'une cellule de fonderie sous pression. Ce bras manipulateur exécutait des commandes stockées sur un tambour magnétique.

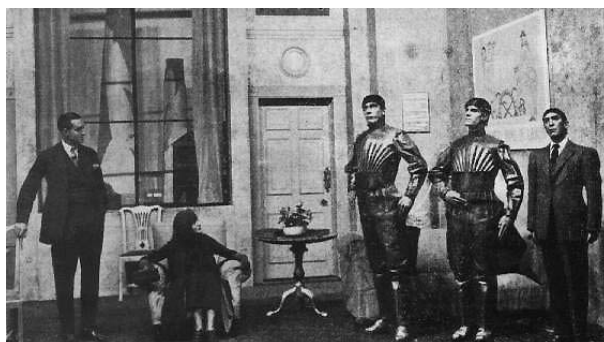


FIGURE 1.1 – Une scène de *R. U. R.* montrant les trois robots ¹

L'organisation internationale de normalisation (ISO) a officiellement défini un robot industriel comme « *un manipulateur automatiquement commandé, reprogrammable et possédant trois axes ou plus* » ².

1. source internet : <http://www.umich.edu/engb415/literature/pontee/RUR/RURsmry.html> accédé le 28/04/2011

2. Standard 8373 :1994, Manipulating Industrial Robots – Vocabulary



FIGURE 1.2 – Unimate, le premier robot industriel

Pour chaque tâche spécifique comme par exemple la soudure, l'encollage, la peinture ou l'assemblage, des solutions ont été proposées. Les bras manipulateurs qui évoluent dans des environnements structurés et entièrement connus exécutent des séries de mouvements à grande vitesse préalablement appris. La présence des robots manipulateurs dans l'industrie est aujourd'hui banalisée notamment dans le domaine de l'automobile (Figure 1.3).



FIGURE 1.3 – Une ligne de production robotisée

Ces robots sont conçus pour fonctionner dans des environnements dont tous les paramètres sont précisément contrôlés. Ces systèmes sont incapables de faire face à des changements de la structure de l'environnement sans reconfiguration ou reprogrammation. A l'inverse, les robots qui évoluent dans le même espace que les humains doivent interagir avec un environnement dynamique et changeant. Ces robots sont appelés *robot personnel* ou *robot de service*. Ces systèmes ne peuvent plus exécuter des mouvements pré-appris et ils doivent s'adapter aux changements importants de l'environnement et à la présence des humains. Dans ce contexte, nous avons cherché à améliorer le comportement du robot pour satisfaire les contraintes introduites par la présence de l'Homme.

1.1 Contexte de la thèse

Cette thèse a été effectuée au sein du LAAS-CNRS (Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique) dans le groupe de recherche RIS (Robotique et InteractionS). Les axes principaux de recherche du groupe portent sur les capacités de perception, de raisonnement, d'apprentissage, d'action et de réaction d'une machine autonome.

Les travaux présentés dans ce manuscrit ont été réalisés dans le cadre du projet européen *DEXMART*³ et du projet ANR⁴ *ASSIST*⁵.

Le projet *DEXMART*, financé par le septième programme cadre (FP7) de la communauté européenne, a pour objectif de combler l'écart des avancées technologiques entre la robotique industrielle et la robotique de service. De part ses huit partenaires, *UNINA*⁶ (coordinateur du projet), *LAAS-CNRS*, *DLR*⁷, *FZI*⁸, *USAAR*⁹, *SUN*¹⁰, *UNIBO*¹¹ et *OMG*¹², le projet a également pour but de renforcer la compétitivité de l'Europe dans les domaines de la robotique personnelle et de service où des capacités de manipulation des robots à deux bras/main sont nécessaires.

Le projet *ASSIST* a pour objectif le développement d'un robot mobile autonome (Figure 1.4), muni de deux bras manipulateurs et d'un système de vision. Il est destiné à porter assistance à des personnes handicapées tétraplégiques dans leurs tâches quotidiennes de manipulation d'objets. Ce projet regroupe quatre laboratoires de recherche, le *LIRMM*¹³ (coordinateur du projet), le *CEA-LIST*¹⁴, le *LAAS-CNRS*, et l'*ISIR*¹⁵. Un dernier acteur important du projet est le Centre de Rééducation Neurologique Propara de Montpellier.



FIGURE 1.4 – CAO du robot développé par le CEA-LIST dans le cadre du projet ASSIST

3. *DEXMART* : "DEXterous and autonomous dual-arm/hand robotic manipulation with sMART sensory-motor skills : A bridge from natural to artificial cognition"

4. ANR : Agence Nationale de la Recherche

5. *ASSIST* : "Étude et développement d'un manipulateur mobile à deux bras pour l'assistance aux handicapés"

6. *UNINA* : PRISMA Lab/DIS & DSF, Università di Napoli Federico II, Italy

7. *DLR* : Institut für Robotik und Mechatronik, Germany

8. *FZI* : Forschungszentrum Informatik an der Universität Karlsruhe, Germany

9. *USAAR* : LPA & EPO, Universität des Saarlandes, Germany

10. *SUN* : DII, Seconda Università degli Studi di Napoli, Italy

11. *UNIBO* : DEIS & DIEM, Alma Mater Studiorum Università di Bologna, Italy

12. *OMG* : OMG plc, England

13. *LIRMM* : Laboratoire d'informatique de Robotique et de Microélectronique de Montpellier

14. *CEA-LIST* : Laboratoire d'Intégration des Systèmes et des Technologies, Fontenay-aux-Roses-aux-Roses

15. *ISIR* : Institut des Systèmes Intelligents et de Robotique, Paris

1.2 Contributions de la thèse

Le travail présenté dans ce manuscrit porte sur le problème de la génération de trajectoire pour les robots manipulateurs dans un contexte d'interaction Homme-robot. Ce contexte introduit des contraintes de sécurité et de confort par rapport à l'Homme que nous traduisons par la limitation de la vitesse, de l'accélération et du jerk. Ces limitations permettent de s'affranchir, du point de vue de la génération de trajectoire, des problèmes liés à la dynamique du système.

La première contribution est le développement du premier, à notre connaissance, générateur de trajectoires en ligne pour des mouvements avec des conditions initiales et finales en accélération et vitesse non nulles et des contraintes en jerk, accélération et vitesse.

La seconde contribution est une méthode d'approximation de trajectoire qui permet d'approximer des trajectoires par des suites de segments de cubiques. Ces deux premières contributions permettent de réaliser un lien entre le planificateur de chemin et le système de commande du robot. Une application originale de l'approximation est l'approximation d'une trajectoire définie dans l'espace cartésien par une trajectoire définie dans l'espace articulaire. Cette approche simplifie la structure du contrôleur du robot et permet d'adapter la trajectoire pour satisfaire les limites cinématiques du robot.

Nous proposons enfin une méthode pour prendre en compte les contraintes de l'interaction Homme-robot pendant l'exécution des mouvements en adaptant la loi de mouvement des trajectoires.

1.3 Organisation du manuscrit

Après ce chapitre d'introduction, nous présentons dans le chapitre 2 un état de l'art général sur les robots de service et sur les bases de la planification de mouvement. Dans le chapitre suivant, nous détaillerons un générateur de trajectoires multidimensionnelles sous contraintes cinématiques basé sur un modèle de trajectoire défini par des suites de segments de courbes polynomiales cubiques (Chapitre 3). Nous présenterons ensuite une méthode d'approximation de trajectoires par des trajectoires définies par le même modèle que celui du générateur (Chapitre 4). Dans le chapitre 5, nous présenterons une architecture planification/contrôle de mouvement pour les robots manipulateurs prenant explicitement en compte l'Homme. Les travaux seront illustrés dans le chapitre 6 par quatre expérimentations réalisées avec la plateforme Jido du LAAS-CNRS. Enfin, nous effectuerons une conclusion générale dans le chapitre 7 et proposerons des perspectives de recherches.

1.4 Publications associées à cette thèse

1. XAVIER BROQUÈRE, DANIEL SIDOBRE ET KHOI NGUYEN, From Motion Planning to Trajectory Control with Bounded Jerk for service Manipulator Robots , *IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska (USA), 2010*. [Broquère 10]
2. EMRAH AKIN SISBOT, LUIS F. MARIN-URIAS, XAVIER BROQUÈRE, DANIEL SIDOBRE ET RACHID ALAMI, Synthesizing Robot Motions Adapted to Human Presence, *International Journal of Social Robotics, Volume 2, Number 3, 2010*. [Sisbot 10]

3. DANIEL SIDOBRE, JEAN-PHILIPPE SAUT, XAVIER BROQUÈRE ET JIM MAINPRICE, Reactive Grasping for Human-Robot Interaction, *Robotics Science and Systems (RSS), Workshop Representations for object grasping and manipulation in single and dual arm tasks, Zaragoza, Spain, 2010.* [Sidobre 10]
4. DANIEL SIDOBRE ET XAVIER BROQUÈRE, Natural and intuitive exchanges of objects between humans and robots, *Workshop on Actuating & Sensing in Robotics, Saarbrücken, Germany, October 6th, 2010.*
5. XAVIER BROQUÈRE, DANIEL SIDOBRE, Planification de mouvements souples, de la planification au déplacement, *Congrès des doctorants EDSYS, Toulouse, France, 2009.*
6. XAVIER BROQUÈRE, DANIEL SIDOBRE ET IGNACIO HERRERA-AGUILAR, Soft motion trajectory planner for service manipulator robot, *IEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, 2008.* [Broquère 08]

1.5 Rapports de projet associés à cette thèse

1. DANIEL SIDOBRE, JEAN-PHILIPPE SAUT, XAVIER BROQUÈRE, JIM MAINPRICE, JUAN CORTES, THIERRY SIMÉON, MOKHTAR GHARBI, FÉLIX INGRAND, MATHIEU HERRB, JÉRÔME MANHES, RACHID ALAMI, RAJA CHATILA, Model of human interaction, *Rapport LAAS N°10473, Rapports de Contrat, ICT-FP7216239-DEXMART, Août 2010, 47p.*
2. DANIEL SIDOBRE, MATHIEU HERRB, JÉRÔME MANHES, XAVIER BROQUÈRE, RACHID ALAMI, RAJA CHATILA, JUAN CORTES, FÉLIX INGRAND, JEAN-PHILIPPE SAUT, THIERRY SIMÉON, Models of information exchanged by two humans exchanging an object for several strategies of handling, *Rapport LAAS N°10360, Rapports de Contrat, ICT-FP7216239-DEXMART, Juin 2010, 74p.*

2

État de l'art

Dans l'industrie, la plupart des robots ne sont constitués que d'un simple bras manipulateur. Ils sont confinés dans des espaces structurés, isolés des Hommes, sans obstacle inconnu. De par leur faible volume de travail, ils réalisent un nombre limité de tâches. Ce volume peut être augmenté par l'ajout d'axes externes mais reste délimité par la zone atteignable par l'organe effecteur du robot. La dimension de cette zone varie en fonction des dimensions du bras, de ses butées et de son nombre de degrés de liberté. Les mouvements exécutés sont décrits par des programmes embarqués sur le robot. Des méthodes manuelles ou automatiques permettent de développer ces programmes. Dans le cas des méthodes manuelles, l'opérateur doit écrire directement le programme. Les programmes générés automatiquement résultent d'une interaction entre l'opérateur et le robot. Les différents fabricants comme par exemple KUKA ¹, ABB ² ou Stäubli ³ proposent des environnements et des langages de programmation différents, ce qui rend l'utilisation de ces robots difficile [Biggs 03].

Dans le contexte de la robotique de service, le robot doit effectuer des tâches dans des environnements équivalents à l'espace atteignable des Hommes. Pour augmenter l'espace de travail, les bras sont montés sur des bases mobiles. Les tâches à réaliser sont plus diverses que dans le contexte industriel (e.g. transporter un objet, débarrasser la table, ramasser un objet au sol, amener un objet à l'Homme...). Chaque tâche peut être décomposée en plusieurs sous tâches élémentaires comme par exemple pour le transport d'un objet : localiser l'objet si sa position n'est pas connue, calculer une prise accessible en prenant en compte l'environnement, calculer un mouvement pour aller saisir l'objet, calculer un mouvement pour amener l'objet à la position désirée... Ainsi pour réussir une tâche de transport d'un objet, le robot doit être capable de planifier et de contrôler une suite de tâches élémentaires. C'est à ce prix qu'il peut agir sur son environnement. Un problème élémentaire de robotique, ici déplacer un objet, souligne

1. KUKA : <http://www.kuka-robotics.com>

2. ABB : <http://www.abb.com/>

3. Stäubli : <http://www.staubli.com>

l'intérêt de la phase de planification.

Dans ce chapitre, nous allons d'abord présenter le concept de robot mobile autonome et les contraintes liées à la présence de l'Homme (Section 2.1). Le chapitre proposera ensuite une étude des mouvements pour l'interaction Homme-robot (Section 2.2). Dans cette deuxième section, les modèles du robot manipulateur et de ses mouvements seront d'abord introduits. Ensuite nous présenterons un état de l'art sur les méthodes de génération de chemin. Enfin nous introduirons le temps en détaillant les méthodes de génération de trajectoire.

2.1 Le robot manipulateur mobile autonome

2.1.1 Quelques robots

Au cours des dernières années, de nombreux robots utilisés comme plateforme d'intégration pour étudier la manipulation ou l'interaction ont vu le jour. Dans cette section, nous allons nous attarder particulièrement sur quatre d'entre eux.

Le premier, Justin [Ott 06] (figure 2.1), est un robot de l'*Institute of Robotics and Mechatronics (DLR)*. Il est composé de deux bras DLR-Lightweight-Robot-III [Hirzinger 02] à sept degrés de liberté chacun, de deux mains à quatre doigts et d'une tête pour la perception de l'environnement. Ses corps sont fixés sur un torse à trois degrés de liberté. Le tout est monté sur une base mobile équipée de quatre roues indépendantes et rétractables [Borst 09, Albu-Schäffer 07, Fuchs 09].

L'Université Technique de Munich (*Technische Universität München, TUM*), développe le robot *Rosie* (figure 2.2) qui est composé de deux bras KUKA-lightweight LWR-IV et d'une base mobile omnidirectionnelle KUKA [Maldonado 10].



FIGURE 2.1 – Justin, robot mobile du DLR

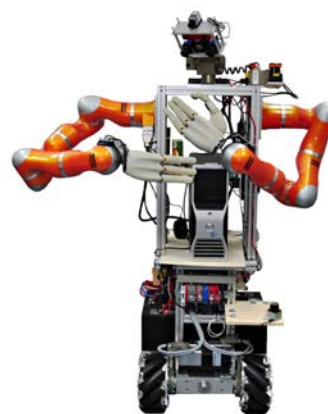


FIGURE 2.2 – Rosie, développé par TUM

Willow Garage commercialise la plateforme *PR2 (Personal Robot 2)*⁴ (figure 2.3). Entièrement développée avec des logiciels libres, ses concepteurs ont la volonté de favoriser le développement collaboratif. Au dessus du système d'exploitation Linux, un des atouts du PR2 est le système libre *ROS (Robot Operating System)*⁵. Les différentes équipes travaillant avec ROS et utilisant la plateforme PR2 peuvent

4. <http://www.willowgarage.com/pages/pr2/overview>

5. <http://www.ros.org>

échanger et comparer leurs résultats. Ce robot possède deux bras à sept degrés de liberté montés sur une base mobile. Il est également équipé d'un large panel de capteurs (camera, télémètre laser, capteur tactile sur les pinces, centrale inertielle...).



FIGURE 2.3 – PR2, développé par Willow Garage

Enfin, le robot sur lequel nous avons mené les travaux de cette thèse est Jido. Il est développé au LAAS-CNRS pour effectuer des recherches sur l'interaction Homme-robot. Jido est un robot de service évolutif composé à ce jour d'un bras manipulateur KUKA-lightweight LWR-IV augmenté d'une main Schunk SAH (figure 2.4, c). Ce système est monté sur une plateforme non holonome Néobotix. Chaque articulation du bras Kuka est dotée d'un capteur de couple, ce qui permet de détecter, par exemple, les collisions avec l'environnement. Le robot est également doté d'un banc de stéréovision et récemment d'un Kinect. La puissance de calcul est fournie par quatre ordinateurs. Le bras et la main ont remplacé un bras Mitsubishi PA10-6C et une pince à mors parallèles (figure 2.4, a,b) depuis juillet 2010. Durant cette thèse, nous avons participé aux évolutions mécaniques de Jido et à la conception du futur robot à deux bras que le LAAS va acquérir courant 2011 (figure 2.4, d). Comme le PR2, Jido est développé avec des logiciels libres dont nos travaux font partie. Comme pour l'ensemble des robots du LAAS⁶, l'architecture logicielle est composée de modules GenoM [Fleury 97, Mallet 10] accessibles par le gestionnaire de paquet *Robotpkg*⁷ également développé au LAAS. Une description plus détaillée de l'architecture logicielle de Jido est faite dans le chapitre présentant les expérimentations (Section 6.1.1).

2.1.2 L'architecture du robot autonome

Un système robotique peut être défini comme un système complexe composé de sous systèmes. Par exemple, pour un robot mobile, deux sous systèmes peuvent être facilement identifiés : un système pour effectuer des tâches de manipulation (système complexe composé d'un ou de plusieurs bras, d'organes effecteurs comme les pinces ou les mains) et un système de locomotion (base mobile à roue, jambes...). L'actionneur généralement utilisé pour mouvoir ces types de robot est le moteur électrique. Ces actionneurs nécessitent un système de contrôle pouvant être un simple asservissement sur une consigne d'un

6. <http://www.laas.fr/robots>

7. <http://homepages.laas.fr/mallet/robotpkg>

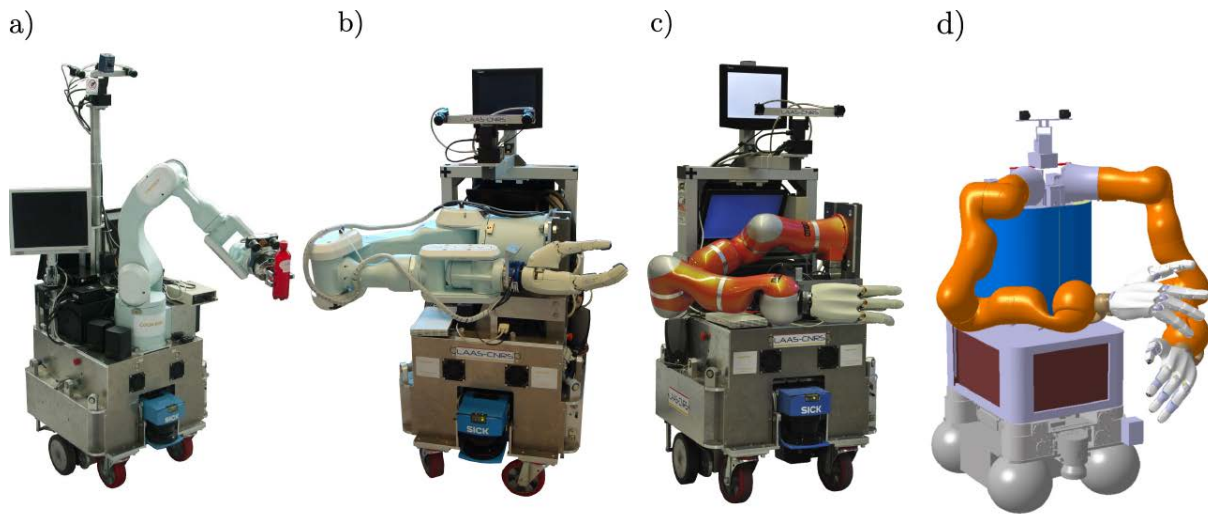


FIGURE 2.4 – Jido, un robot évolutif du LAAS-CNRS

téléopérateur ou un système autonome piloté par un superviseur (composant logiciel). La planification et l'exécution de la tâche se basent sur des informations sur l'état du robot et de l'environnement acquises par des capteurs proprioceptifs (capteurs de positions, force...) et extéroceptifs (caméra, capture de mouvement, ..). La figure 2.5 présente l'architecture développée au LAAS (a) [Alami 98] et ses principaux sous systèmes (b).

Dans l'exemple de l'architecture LAAS, trois niveaux sont identifiés :

1. *Le niveau décisionnel ou délibératif* : ce niveau contient les systèmes décisionnels de haut niveau. Généralement, c'est un couple composé d'un planificateur de tâche (planificateur symbolique) et d'un superviseur [Clodic 09]. Le superviseur est en charge d'émettre les requêtes vers le niveau fonctionnel afin de permettre la réalisation des actions courantes. Il est également en charge de contrôler le bon déroulement de l'exécution des tâches. Le superviseur développé au LAAS est nommé SHARY [Clodic 07].
2. *Le contrôle d'exécution* : ce niveau est chargé de vérifier à tout moment la validité des requêtes destinées au niveau fonctionnel. Dans l'architecture LAAS, ce niveau est pris en charge par le superviseur.
3. *Le niveau fonctionnel* : ce niveau est composé d'un ensemble de modules logiciels permettant de contrôler les actionneurs du robot et de transcrire les données fournies par les capteurs en informations exploitables par le niveau délibératif ou bien exécuter les commandes de ce dernier. Deux composants principaux font partie de ce niveau :
 - *Le planificateur du mouvement* : cette étape effectue le choix du mouvement ;
 - *Le contrôle* : réalise l'interface avec les systèmes mécaniques, commande l'exécution de chaque mouvement élémentaire prévu et les adapte localement si nécessaire. Il permet également de récupérer les données capteurs.

L'ensemble des données des modèles (modèle 3D du robot, de l'environnement, de l'Homme) est accessible par l'ensemble des composants de l'architecture. L'ouvrage de Siciliano et Khatib, *Springer*

Handbook of robotics présente une introduction des domaines relatifs à chaque niveau [Siciliano 08].

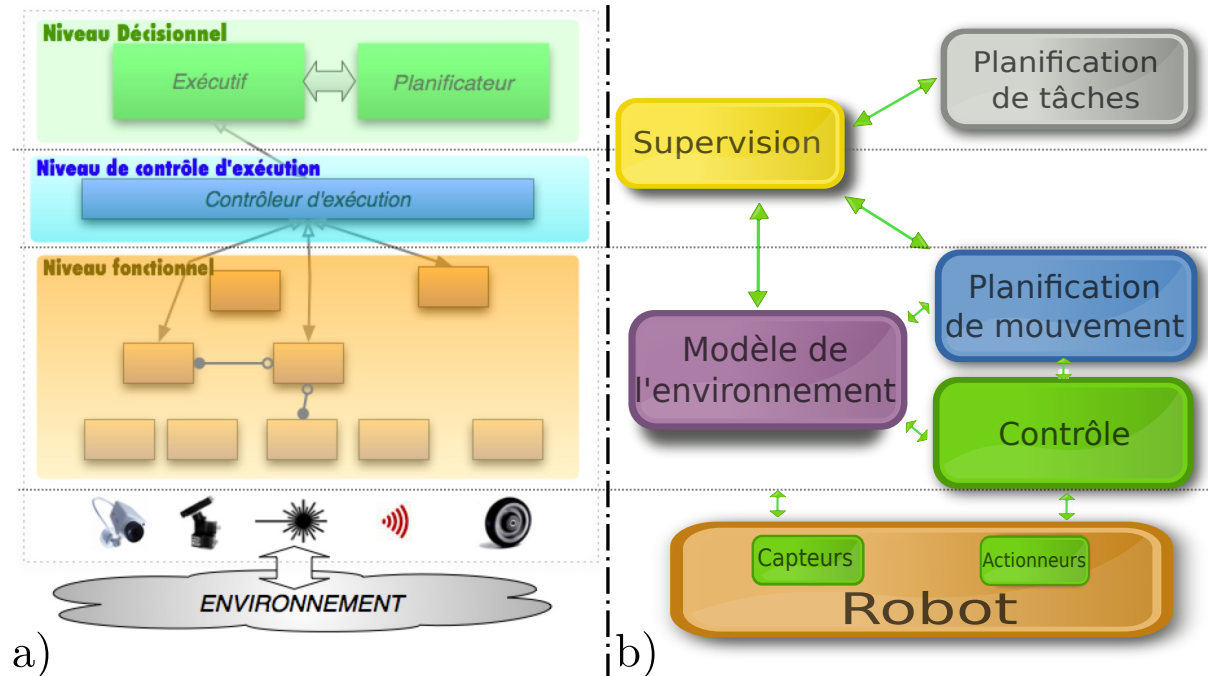


FIGURE 2.5 – a) Structure d'une architecture de contrôle à trois niveaux ; b) Les principaux composants de l'architecture

Prenons l'exemple d'un robot mobile avec un bras manipulateur effectuant une tâche de saisie d'un objet. Pour planifier un mouvement, le robot doit raisonner sur un modèle de l'environnement qui est soit fourni par l'utilisateur, soit construit à l'aide des capteurs extéroceptifs. Le superviseur envoie une requête au planificateur de tâche pour lui demander de générer un plan. Ce plan est une suite de tâches élémentaires comme le calcul de prise sur l'objet ou le calcul du mouvement d'approche du robot au point de saisie de l'objet. Chaque sous tâche est ensuite planifiée par le planificateur de mouvement. Si la planification de la tâche globale est correcte, le superviseur va ordonner l'exécution des différents mouvements à travers le système de contrôle.

Pour garantir le succès de la tâche, il est nécessaire de séquencer correctement les sous tâches et donc de maîtriser le temps pendant l'exécution. Le système de supervision communique avec le système de contrôle. Par exemple, le superviseur doit donner l'ordre de saisir l'objet seulement lorsque l'organe préhenseur du robot est correctement positionné par rapport à l'objet. Considérons un robot mobile constitué de deux bras manipulateurs ; pour exécuter une tâche de saisie d'objet à deux bras / deux mains, les bras doivent exécuter les mouvements planifiés de manière synchronisée.

En présence de l'Homme, les mouvements du robot devront être adaptés pendant l'exécution pour prendre en compte la modification de l'environnement et les mouvements de l'Homme. Nous présenterons des méthodes de planification en présence de l'Homme dans la section 2.2. Dans ce contexte, la sécurité de l'Homme devient la contrainte principale. Le paragraphe suivant présente les normes actuelles et les méthodes permettant de quantifier et de réduire les risques de blessures.

2.1.3 La sécurité

La norme développée par l'Organisation Internationale de Standardisation (ISO) ISO-10218 spécifie les exigences de sécurité pour l'intégration des robots industriels dans un contexte d'opération collaborative [ISO-10218 06]. Selon cette norme, dans l'ensemble de l'espace de travail du robot, la vitesse de l'organe terminal (*TCP*) ne doit pas dépasser 0.25 m/s, la force statique exercée doit être inférieure à 150 N et la puissance du système doit être inférieure à 80 watts. Moon et al. présentent une étude sur les différentes normes existantes (figure 2.6) [Moon 09]. Les robots y sont classés en deux catégories : les robots industriels et les robots de service. Cependant, dans le contexte de la robotique de service, ces normes ne sont pas suffisantes. Les vitesses et les forces maximales peuvent encore être dangereuses dans certaines circonstances (eg. manipulation d'un couteau...) et sont insuffisantes pour exécuter certaines tâches qui ne sont pas dangereuses. Ces dernières années des efforts ont été réalisés pour la mise au point de nouvelles normes concernant les robots en interaction avec l'Homme (*IROS 2010 Workshop on Standardization for Service Robots*). Des projets européens ont participé à la définition de robots intrinsèquement sûrs et à la conception de systèmes de contrôle assurant la sécurité de l'Homme [Phriends 09, SMErobot™ 09].

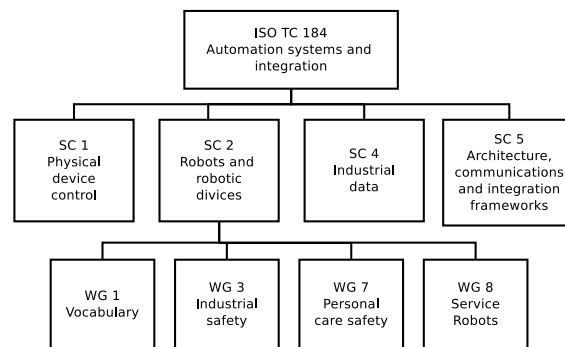


FIGURE 2.6 – Organisation de l'ISO TC 184 [Moon 09]

Pour quantifier les blessures dues à une collision, plusieurs critères existent. Un des plus connus est le critère HIC “*Head Injury Criteria*” défini par la NHTSA⁸ en 1972. Ce critère est principalement utilisé pour évaluer les risques dans le domaine de l'automobile. Cependant, Haddadin et al. ont montré par des études expérimentales que ce critère n'est pas suffisant pour l'interaction physique Homme-robot (*pHRI*). En effet, à faible vitesse, les robots peuvent provoquer des blessures de type différents. Ainsi, ils introduisent la fracture du crâne comme critère d'évaluation des blessures [Haddadin 08b, Haddadin 08a, Haddadin 09]. Park et al. proposent un modèle de collision plus précis qui inclut la raideur du cou [Park 09]. A partir de ce modèle et d'un critère basé non plus sur la fracture du crâne mais sur une blessure au niveau du cou (fissure du cartilage de la thyroïde et de la cricoïde, angle d'extension de la tête), ils montrent que le critère de blessure du cou est plus approprié que celui du crâne. Comme nous pouvons le voir dans le tableau 2.1, le cou représente la partie anatomique la plus vulnérable du corps humain. Ikuta et al. classifient les stratégies de sécurité en deux catégories (tableau 2.2) : la sécurité avant la collision du robot avec l'humain et celle après le contact [Ikuta 03]. Lors du contact, la sécurité est prise en compte à deux niveaux : elle doit être assurée d'une part par les propriétés mécaniques du

8. NHTSA : *National Highway Traffic Safety Administration*

système [Bicchi 04, Zinn 04] et d'autre part par l'utilisation de lois de commande permettant d'absorber l'énergie du système due à sa masse et à sa vitesse [De Santis 08, Zollo 05].

L'ensemble de ces études montre que la vitesse du robot est un paramètre important à maîtriser pour respecter la contrainte de sécurité. Cependant, comme nous l'avons vu, limiter la vitesse selon les normes actuelles n'est pas suffisant. Ces limites doivent varier en fonction de la présence ou non de l'Homme dans le volume de travail du robot. Dans la section suivante nous allons voir que la limitation de la vitesse n'est pas suffisante dans l'interaction Homme-robot.

TABLE 2.1 – Limites en effort acceptables avant blessure grave [Park 09]

		Limites acceptables
Crane [Melvin 80]	Os frontal	4.0 kN
	Os temporal	3.12 kN
	Os occipital	6.41 kN
Cou	Extension [Mertz 71]	47.4 Nm
	Max. Incl. de la tête [Buck 59]	73°
	Thyroïde et Cricoïde [Melvin 73]	0.34 kN
Autres parties	Bas ventre [Miller 89]	3.76 kN
	Tibia [Devore 99]	5.4 kN
	Cubitus [Begeman 99]	1.37 kN

2.1.4 L'interaction Homme-robot

L'interaction Homme-robot *HRI* est un axe d'étude relativement récent ; Goodrich en a proposé une synthèse [Goodrich 07]. Il distingue deux situations d'interaction :

- *l'interaction distante* : l'Homme et le robot ne partagent pas le même espace physique comme par exemple dans le cas de la téléopération chirurgicale, l'interaction entre le chirurgien et le robot est distante (contrairement à celle entre le patient et le robot) ;
- *l'interaction de proximité* : le robot et l'humain partagent le même espace de travail physique. Il distingue aussi l'interaction cognitive de l'interaction physique :
 - l'interaction cognitive *cPHRI* correspond à l'interaction verbale ou gestuelle ;
 - l'interaction physique *pHRI* est lorsque l'Homme et le robot rentrent en contact. Ce contact est soit prévu (échange d'objet, coopération pour réaliser une tâche) soit imprévu (collision).

Les travaux de cette thèse entrent dans le cadre de l'interaction de proximité.

TABLE 2.2 – Classification des stratégies de sécurité [Ikuta 03]

		stratégie de contrôle	stratégie de conception	
Avant la collision	éviter la collision	distance	-	
	minimiser la force de l'impact	vitesse moment d'inertie	-	
Après la collision	atténuation diffusion	raideur	coque	compliance des articulations
			surface	forme

2.1.5 Des robots sociaux

Les robots qui évoluent en présence de l'Homme doivent respecter des règles sociales. Contrairement au milieu industriel où les robots sont isolés des humains, les robots sociaux doivent non seulement être sûrs (Section 2.1.3) mais aussi respecter des règles et des protocoles lorsqu'ils évoluent [Chatila 02, Fong 03, Sisbot 07] :

- *Avoir des mouvements sûrs* : le robot ne doit pas blesser l'humain (objet du paragraphe 2.1.3) ;
- *Avoir des mouvements fiables et efficaces* : le robot doit accomplir la tâche demandée au mieux par rapport à ses capacités ;
- *Avoir des mouvements socialement acceptables* : le robot doit prendre en compte le confort et les préférences de l'humain ; le mouvement pour accomplir la tâche doit être lisible, c'est à dire qu'il doit permettre à l'humain de comprendre ce que le robot réalise.

Les travaux de Flash et Hogan [Flash 84, Hogan 84] ont montré que les mouvements effectués par les humains sont en général limités en jerk et en accélération. Ainsi, un système limitant ces deux paramètres aura un mouvement souple qui ressemble à ceux des humains et sera plus facilement accepté par l'Homme. Ensuite, l'Homme doit explicitement être pris en compte, d'une part pendant la planification et d'autre part pendant l'exécution. Explicitement ne signifie pas seulement en tant qu'obstacle à éviter mais aussi en considérant l'Homme avec sa structure cinématique et un espace atteignable limité. Par exemple dans le cadre particulier du projet ASSIST où le robot devra porter assistance à des personnes tétraplégiques, les mouvements devront être réalisés en prenant en compte la mobilité réduite des patients.

Enfin, la planification pourra également prendre en compte les habitudes et les préférences de l'Homme (par exemple gaucher ou droitier).

Ce paragraphe met en avant la notion de confort de l'Homme pendant l'interaction avec le robot. Le respect de cette contrainte de confort influence les caractéristiques du mouvement qu'effectue le robot. Pour être confortable, le mouvement doit être lisible et compris par l'Homme. Une solution est de limiter l'accélération et le jerk. La section suivante présente la planification de mouvement pour satisfaire la contrainte de confort et la contrainte de sécurité liée à la vitesse du mouvement.

2.2 Les mouvements pour l'interaction Homme-robot

Avant d'introduire les modèles utilisés pour la planification de mouvement pour les robots manipulateurs mobiles, nous présentons dans un premier temps quelques notions de mécanique du mouvement et les notations pour décrire le mouvement des corps. Nous précisons ensuite ces notions dans le cas des robots manipulateurs mobiles.

2.2.1 Le mouvement des corps

L'étude de la cinématique du mouvement se différencie de celle de la dynamique. La cinématique est l'étude des mouvements sans se soucier des causes qui les produisent. Concrètement, la cinématique s'intéresse à l'évolution des corps au cours du temps et plus précisément elle étudie la position, la vitesse, l'accélération et ses dérivées. La dynamique étudie les relations entre les forces et les mou-

vements qu'elles produisent. Dans notre architecture présentée dans la section 2.1.2, la cinématique intervient au niveau de la planification de chemin tandis que la dynamique est traitée au niveau de l'exécution/contrôle. Dans nos travaux, nous nous sommes uniquement intéressé à la planification avec des contraintes cinématiques.

Avant d'étudier le mouvement d'un corps, nous introduisons les outils géométriques.

2.2.1.1 Situation d'un corps :

Considérons un référentiel \mathcal{R}_W de système d'axes $(O_W, \vec{x}_W, \vec{y}_W, \vec{z}_W)$. La situation d'un corps rigide B est la localisation spatiale de ce corps (*position* et *orientation*). Pour définir la situation X d'un corps B dans l'espace, nous lui associons un repère \mathcal{R}_B (figure 2.7).

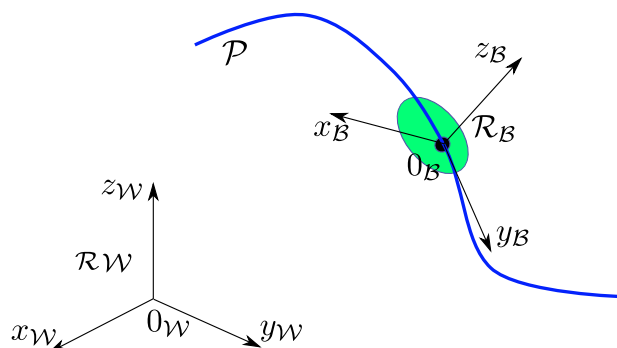


FIGURE 2.7 – Localisation d'un corps dans l'espace 3D

Trois systèmes de coordonnées sont couramment utilisés dans la littérature pour donner la position relative de deux repères :

- Coordonnées cartésiennes ;
- Coordonnées cylindriques ;
- Coordonnées sphériques.

L'orientation d'un repère peut être donnée par plusieurs représentations dont :

- 3 angles (angles de Bryant, angles d'Euler) ;
- Matrice de rotation ou cosinus directeurs ;
- Paramètres d'Euler-Rodrigues (quaternion unitaire) ;
- Angle plus vecteur.

La situation d'un corps B peut donc être modélisée par de nombreuses combinaisons des représentations de la position et de l'orientation et de leurs variantes. Six paramètres (3 translations, 3 rotations) définissent cette transformation.

La matrice de transformation homogène $T_{W/B}$ est couramment utilisée. Elle permet d'exprimer la transformation entre un repère de référence, par exemple \mathcal{R}_W , et un repère \mathcal{R}_B associé au corps B :

$$T_{W/B} = \begin{bmatrix} R & P \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

où R est une matrice de rotation 3×3 appartenant au groupe spécial orthogonal des rotations $SO(3)$ et P une matrice 3×1 représentant la position du point O_B . La matrice de transformation homogène 4×4

associée appartient alors au groupe des déplacements $SE(3) = \mathbb{R}^3 \times SO(3)$. Soit b un point appartenant au corps B et positionné par rapport au repère \mathcal{R}_B par le vecteur :

$${}^B P_b = [X_b \ Y_b \ Z_b \ 1]^T$$

La position de ce point b dans le repère \mathcal{R}_W est :

$${}^W P_b = T_{W B} {}^B P_b$$

Soit \mathcal{R}_D un repère appartenant à un corps D et défini par rapport au repère \mathcal{R}_B par la transformation $T_{B D}$. La situation $T_{W D}$ du repère \mathcal{R}_D par rapport à \mathcal{R}_W est :

$$T_{W D} = T_{W B} T_{B D}$$

Il est important de noter que la représentation par les matrices homogènes est insuffisante dans certains cas. Par exemple, prenons le cas d'un téléphone avec un combiné filaire. La matrice homogène exprimant la transformation entre la base et le combiné ne permet pas de connaître le nombre de tours que le combiné a effectué pour torsader le fil. La représentation de l'orientation par un quaternion unitaire (4 paramètres) ne permet également pas de représenter ce nombre de tours (si supérieur à 2). Dans ce cas, la seule modélisation possible pour représenter le nombre de tours est l'angle plus vecteur (4 paramètres avec le vecteur normé). Trois paramètres suffisent lorsque le module du vecteur est égal au module de l'angle. La situation sera alors représentée par un déplacement Dep :

$$Dep_{W B} = \begin{bmatrix} \vec{P}_B & a \cdot \vec{V}_B \end{bmatrix} \quad (2.2)$$

avec P_B le vecteur de position et V_B le vecteur unitaire autour duquel la rotation de module a est effectuée.

Dans nos travaux, nous avons utilisé la représentation par matrice homogène, quaternion unitaire et déplacement.

Le lecteur pourra se référer au chapitre 1 du livre de Siciliano [Siciliano 08] pour plus de détails sur les représentations de la situation et les calculs de passage entre les types de représentations.

2.2.1.2 Interpolation entre deux situations :

L'interpolation consiste à calculer l'évolution d'un repère entre deux situations. Pour la position, l'interpolation linéaire dans l'espace cartésien est généralement utilisée. Soit u un paramètre $\in \mathbb{R}$ tel que $u \in [0, 1]$. L'interpolation linéaire entre la position P_B et P_C est donnée par la relation :

$$P(u) = P_B + u \cdot (P_C - P_B) \quad (2.3)$$

L'interpolation de l'orientation peut être effectuée, par exemple, en utilisant les quaternions unitaires ou la représentation vecteur/angle (déplacement).

La forme de l'espace des quaternions unitaires est une surface d'hyper-sphère de dimension 4 et de rayon égal à 1. L'interpolation linéaire entre deux quaternions est la plus simple et correspond à une ligne droite située à l'intérieur de cette hyper-sphère. Le quaternion est normalisé à chaque pas

de l'interpolation pour le projeter sur la sphère de dimension 4. Cette méthode est nommée *NLERP*⁹. Le problème de l'interpolation linéaire des quaternions vient du fait que les échantillons ne sont pas uniformément répartis sur la sphère à cause de la normalisation. Cette non uniformité introduit une non linéarité de la vitesse angulaire.

Shoemake [Shoemake 85] propose le *SLERP*¹⁰ pour corriger les problèmes de l'interpolation linéaire. Le slerp produit un arc sur la surface de l'hyper-sphère où ses échantillons sont uniformément distribués. Soit q_B le quaternion représentant l'orientation du repère \mathcal{R}_B par rapport à \mathcal{R}_W et q_C le quaternion de l'orientation du repère \mathcal{R}_C par rapport à \mathcal{R}_W . Le slerp est défini par la relation :

$$\forall u \in [0, 1] \quad \text{Slerp}(q_B, q_C, u) = q(u) = q_B(q_B^{-1}q_C)^u \quad (2.4)$$

Dans nos travaux, l'interpolation entre deux situations a été effectuée en utilisant la représentation de l'orientation par angle plus vecteur avec la norme du vecteur égale au module de l'angle. Pour cela nous avons calculé le déplacement relatif en passant par les quaternions :

$$\text{Dep}_{\mathcal{R}_C|\mathcal{R}_B} = \begin{bmatrix} \vec{P}_{BC} & a_{BC}\vec{V}_{BC} \end{bmatrix} \quad (2.5)$$

L'interpolation est calculé par l'équation :

$$\forall u \in [0, 1] \quad q(u) = q_B \cdot f_{va \rightarrow quat}(u \cdot a_{BC}\vec{V}_{BC}) \quad (2.6)$$

où $f_{va \rightarrow quat}$ est la fonction de conversion d'un vecteur-angle en quaternion. En utilisant les matrices homogènes, l'interpolation de la situation est donnée par :

$$\forall u \in [0, 1] \quad T(u) = T_B \cdot f_{dep \rightarrow th}(u \cdot a_{BC}\vec{V}_{BC}, u \cdot \vec{P}_{BC}) \quad (2.7)$$

où $f_{dep \rightarrow th}$ est la fonction de conversion du déplacement en matrice homogène.

Une explication détaillée des différents types d'interpolation est traitée dans le rapport technique de Dam [Dam 98].

2.2.1.3 Chemin :

Un chemin entre une situation initiale X_I et une situation finale X_F est une représentation géométrique du déplacement du corps qui ne considère pas le temps. Soit u un paramètre tel que ($u \in [u_{min}, u_{max}]$), le chemin est défini par une fonction continue $\mathcal{P}(u)$ de $[u_{min}, u_{max}]$ dans \mathbb{R}^n telle que $\mathcal{P}(u_{min}) = X_I$, $\mathcal{P}(u_{max}) = X_F$ et n la dimension de l'espace considéré.

L'abscisse curviligne d'un chemin de classe C^1 et régulier (vecteur dérivé non nul en chaque point) permet de paramétrer un chemin par sa longueur. L'abscisse curviligne s d'un chemin \mathcal{P} est la fonction strictement croissante de $[u_{min}, u_{max}]$ dans \mathbb{R} telle que si u_0 et u_1 sont deux éléments de $[u_{min}, u_{max}]$

9. NLERP : **n**ormalized **l**inear **i**nter**p**olation

10. SLERP : **s**pherical **l**inear **i**nter**p**olation

vérifiant $u_0 < u_1$, alors $s(u_1) - s(u_0)$ est la longueur du chemin \mathcal{P} de u_0 à u_1 (fig 2.8) :

$$s(u_1 - u_0) = \int_{u_0}^{u_1} \left\| \frac{d\mathcal{P}(\mu)}{d\mu} \right\| d\mu \quad (2.8)$$

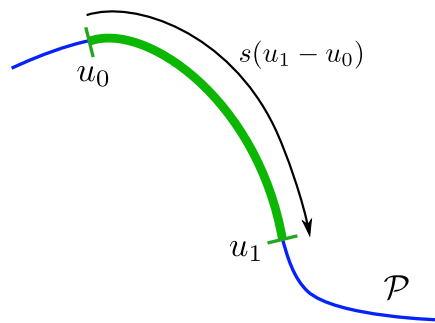


FIGURE 2.8 – L'abscisse curviligne orientée

Cependant cette intégrale est généralement très difficile à calculer.

Un chemin de X_I à X_F peut être également représenté par une suite de sous chemins définis entre des points du chemin. Par exemple si X_1 et X_2 sont des points du chemin (X_I, X_F) , ils permettent de définir trois sous chemins : (X_I, X_1) , (X_1, X_2) et (X_2, X_F) .

Deux classes de chemins définis par une série de points sont très souvent rencontrés : l'interpolation et l'approximation (figure 2.9). L'interpolation consiste à générer un chemin passant par l'ensemble des points intermédiaires tandis que l'approximation consiste à calculer un chemin passant proche des points intermédiaires.

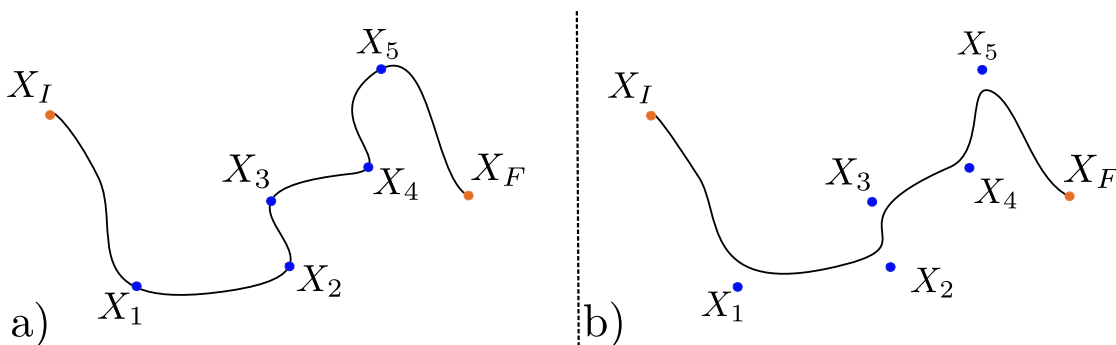


FIGURE 2.9 – Un chemin interpolé (a) et un chemin approximé (b)

Pour interpoler les points du chemin, les splines cubiques (suite de courbes polynomiales cubiques) ont été la première méthode mise au point pour le domaine de la conception mécanique. Les courbes de Béziérs sont très utilisées pour l'approximation. De nombreux travaux traitent le problème de l'interpolation et de l'approximation avec différents types de courbes telles que les courbes de Béziérs, généralisées par les B-Splines ou encore les NURBS (Non Uniform Rational B-Spline) [Piegl 87, Shikin 95, Piegl 97].

2.2.1.4 Trajectoire :

Un mouvement (ou trajectoire) caractérise le déplacement d'un repère au cours du temps et est défini par un chemin \mathcal{P} et une loi d'évolution continue $u(t)$. Une trajectoire $\text{TR}(t)$ représentant un déplacement de X_I à X_F sur le chemin \mathcal{P} est une fonction continue du temps t tel que $t \in [t_I, t_F]$:

$$\text{TR}(t) = \mathcal{P}(u(t)) \quad (2.9)$$

avec $u(t)$ fonction continue de $\mathbb{R} \rightarrow \mathbb{R}$, $\text{TR}(t_I) = X_I$ et $\text{TR}(t_F) = X_F$.

Dans cette partie, nous avons introduit quelques outils pour représenter la situation et le mouvement d'un corps dans \mathbb{R}^3 . Dans le paragraphe suivant, nous allons étendre ces concepts pour décrire le mouvement des robots manipulateurs mobiles.

2.2.2 Le mouvement des robots

La modélisation géométrique et cinématique des robots et la planification de mouvement sont largement décrites dans la littérature [Canny 88, Latombe 91, Choset 05, LaValle 06].

L'environnement dans lequel évolue le robot est l'*espace de travail* \mathcal{W} (figure 2.10) avec $\mathcal{W} \subset \mathbb{R}^3$. Cet espace comprend les objets sur lesquels le robot doit agir, les obstacles et les humains. Un robot est un ensemble de corps rigides reliés par des liaisons cinématiques (e.g. liaison glissière, pivot, rotule...). Chaque liaison constitue un ou plusieurs degrés de liberté du robot. L'ensemble de ces liaisons définissent la *chaîne cinématique*. Une extrémité de la chaîne cinématique est appelée *base*. Les extrémités du ou des bras manipulateurs sont appelés *organe(s) effecteur(s)*. Lozano-Pérez propose de représenter la posture du robot par un vecteur dont la dimension correspond au nombre de degrés de liberté du robot [Lozano-Pérez 83]. L'espace de la représentation paramétrique des postures du robot est nommé *espace des configurations* \mathcal{C} . Par exemple, un objet en déplacement libre dans l'espace, possède six degrés de liberté (trois translations et trois rotations), son espace des configurations est donc de dimension six.

Nous noterons $\mathcal{R}_{\mathcal{W}}$, le repère définissant l'origine de l'espace de travail \mathcal{W} . Le repère de la base du robot sera noté \mathcal{R}_{Rob} .

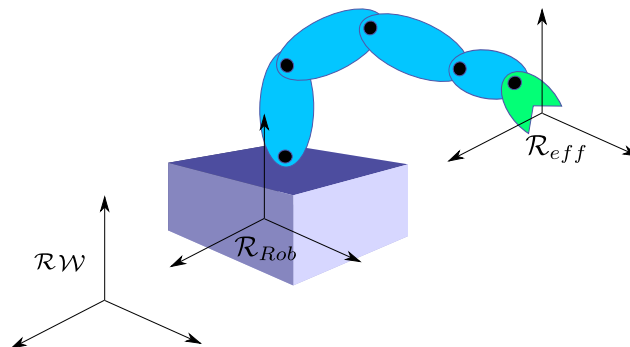


FIGURE 2.10 – Repères associés au robot

Le modèle géométrique direct (MGD) d'un robot, défini par la fonction \mathbf{f}_{MGD} (eq. 2.10), permet d'exprimer la situation de l'organe terminal X (appelée également coordonnées opérationnelles) du robot

manipulateur par rapport à \mathcal{R}_{rob} en fonction de sa configuration q :

$$X = \mathbf{f}_{\text{MGD}}(q) \quad (2.10)$$

Le modèle géométrique inverse (MGI) permet de déterminer une configuration (pas forcément unique) du robot q en fonction de la situation de l'organe terminal :

$$q = \mathbf{f}_{\text{MGI}}(X) \quad (2.11)$$

Le modèle cinématique direct permet de calculer les vitesses des coordonnées opérationnelles à partir des vitesses articulaires \dot{q} en utilisant la matrice jacobienne $\mathbf{J}(q)$:

$$\dot{X} = \mathbf{J}(q)\dot{q} \quad (2.12)$$

Le problème de cinématique inverse est le problème inverse de la cinématique directe. Sa résolution doit permettre de déterminer les vitesses articulaires \dot{q} en fonction des vitesses opérationnelles pour une configuration q donnée. Dans le cas des robots manipulateurs de type série, la résolution du problème de cinématique inverse est plus compliquée que celle du problème cinématique direct. Nous proposons au lecteur de se reporter aux ouvrages de Dombre et Khalil [Khalil 99, Dombre 07] qui présentent les méthodes de résolution.

Dans le cas d'un robot mobile manipulateur, les degrés de liberté sont redondants ; c'est à dire qu'une infinité de configurations du robot sont possibles pour une même situation de l'organe effecteur. Pour représenter la posture complète d'un robot mobile composé de deux bras et deux mains dans l'espace cartésien, nous utiliserons un vecteur X_I composé de :

- la situation de la base du robot ;
- des situations des organes effecteurs des deux bras ;
- de la configuration des axes indépendants des bras s'ils possèdent plus de six degrés de liberté ;
- des configurations des mains ;
- de la configuration de la tête.

La planification d'un chemin pour le robot peut être effectuée soit dans l'espace des configurations (espace articulaire), soit dans l'espace cartésien. La plupart des planificateurs calculent les chemins dans l'espace des configurations ; c'est la méthode la plus simple. Elle permet d'obtenir directement l'évolution des actionneurs du robot. La planification dans l'espace cartésien quant-à-elle permet de définir facilement les contraintes du déplacement (e.g. maintenir une carafe verticale se traduira par la limitation de certaines rotations de la situation de l'organe terminal). Lorsque le chemin et la trajectoire sont planifiés dans l'espace cartésien, le niveau de contrôle calcule la cinématique inverse pendant l'exécution. Dans le chapitre 5, nous proposons une méthode originale pour convertir une trajectoire définie dans l'espace cartésien en une trajectoire définie dans l'espace articulaire.

Dans cette partie, nous avons introduit les notions nécessaires pour la planification de mouvement. La section suivante présente les méthodes de planification du chemin.

2.2.3 La planification de chemin

La planification de chemin a suscité un très grand nombre de travaux. Nous proposons au lecteur de se reporter au chapitre 5 du livre *Springer Handbook of Robotics* rédigé par Kavraki et LaValle [Kavraki 08] qui présente une introduction du domaine ou encore à l'ouvrage de LaValle [LaValle 06] qui en détaille de nombreuses méthodes.

Dans cette section, nous traitons d'abord les méthodes de base de la planification de chemin puis nous nous intéressons aux méthodes prenant explicitement en compte l'Homme.

2.2.3.1 Les méthodes générales de planification de chemin

L'espace des configurations \mathcal{C} est décomposé en deux sous espaces :

- l'espace des configurations libres de collision, noté \mathcal{C}_{free} ;
- l'espace des configurations du robot en collision, noté \mathcal{C}_{obst} .

L'espace des configurations \mathcal{C} est l'union de ces deux espaces (figure 2.11) : $\mathcal{C} = \mathcal{C}_{free} \cup \mathcal{C}_{obst}$.

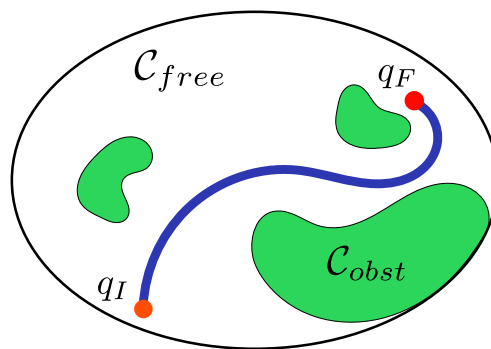


FIGURE 2.11 – L'espace des configurations \mathcal{C}

Le problème de planification de chemin est le calcul d'un chemin sans collision allant d'une configuration initiale $q_I \in \mathcal{C}_{free}$ à une configuration finale $q_F \in \mathcal{C}_{free}$. La solution, si elle existe, est un chemin \mathcal{P} sans collision.

Pour résoudre ce problème, trois grandes familles de techniques ont été développées : les méthodes de *décomposition cellulaire* [Schwartz 83], les *champs de potentiel* [Khatib 86] et la *construction de graphes* [Nilsson 69]. La notion de *complétude* différencie ces approches. On distingue des planificateurs complets, des planificateurs complets en résolution et des planificateurs de complétude probabiliste.

Les méthodes de planification complètes (ou méthodes exactes) garantissent de trouver une solution ou d'indiquer qu'il n'y en a pas. C'est la première famille de méthode développée pour résoudre des problèmes de planification de chemin (e.g. Déménageur de piano [Schwartz 83, Canny 88]). Les méthodes complètes en résolution sont basées sur la discrétisation de l'espace des configurations. Elles garantissent de trouver un chemin solution s'il existe, mais uniquement à une résolution donnée. Les méthodes probabilistiquement complètes garantissent quant à elles de trouver une solution en un temps fini si elle existe. Contrairement aux méthodes exactes et complètes en résolution, les méthodes probabilistes ne sont pas basées sur une construction préalable de l'espace des configurations. On distingue deux catégories de méthodes probabilistes : les techniques basées sur la construction de *réseaux probabilistes* et les techniques de diffusions (ou incrémentales) :

– les méthodes basées sur des réseaux probabilistes :

Ces méthodes ont été introduite parallèlement par Kavraki et Latombe sous le nom de *PRM* (*Probabilistic Roadmap*) [Kavraki 96] et par Svestka et Overmars sous le nom de *PPP* (*Probabilistic Path Planner*) [Švestka 97]. Une première phase d'initialisation capture la connexité de l'espace libre \mathcal{C}_{free} indépendamment des configurations initiale q_I et finale q_F . Les requêtes de planification sont ensuite résolues rapidement en connectant les configurations initiale et finale au réseau pré-calculé. Les réseaux (ou graphes) sont constitués de nœuds (configurations) $q \in \mathcal{C}_{free}$ connectés par des arrêtes (*chemin locaux*) (figure 2.12). Ces méthodes sont dites multi-requêtes. Plusieurs méthodes s'appuyant sur ce principe ont été proposées notamment pour traiter les passages étroits [Hsu 03], pour concentrer des noeuds à la surface des obstacles *OBPRM* [Amato 98], pour obtenir des réseaux plus compact *Visib-PRM* [Siméon 00], *PDR* [Jaillet 08] ...).

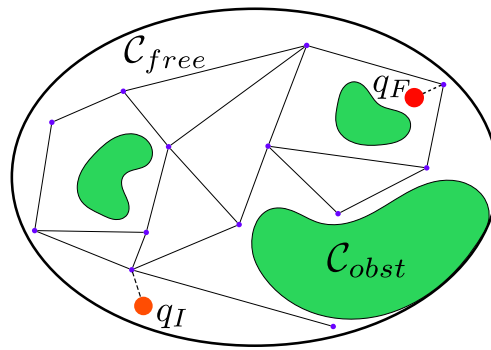


FIGURE 2.12 – Construction d'un réseau probabiliste PRM

– les méthodes de diffusion :

Ces méthodes ne nécessitent pas le pré-calcul d'un réseau, elles sont basées sur l'expansion d'un arbre dans l'espace des configurations à partir des configurations initiales et finales (figure 2.13). Contrairement aux méthodes *PRM*, ces techniques ne cherchent pas à calculer toute la connexité de \mathcal{C}_{free} mais explorent seulement certaines de ses régions afin de résoudre un problème donné. Ces méthodes sont dites mono-requête. La méthode *RRT* (*Rapidly-exploring Random Trees*) est aujourd'hui la plus populaire [LaValle 98, LaValle 01]. Cet algorithme permet de résoudre des requêtes pour des systèmes complexes dans des temps raisonnables. Avec la méthode *RRT* basique nommée "RRT-Extend", l'extension de l'arbre se fait avec un pas régulier. Ceci signifie que les configurations générées sont au plus à une distance d du noeud de l'arbre le plus proche. La variante "RRT-Connect" proposée dans [Kuffner 00] réitère la phase d'extension de *RRT-Extend* tant que les contraintes du système (*e.g.* absence de collisions) sont respectées. D'autres extensions au *RRT* ont été récemment développées. Parmi elles, Rodriguez et al. [Rodriguez 06] ont proposé une méthode appelée *Obstacle Based RRT* qui permet de prendre en compte la nécessité d'éloigner le chemin des obstacles pour éviter les collisions. Jaillet et al. ont proposé d'introduire un critère de visibilité des noeuds dans l'expansion de l'arbre [Jaillet 05] ou encore un critère de coût [Jaillet 08].

Dans un contexte où le robot possède un grand nombre de degrés de liberté, les méthodes exactes ne

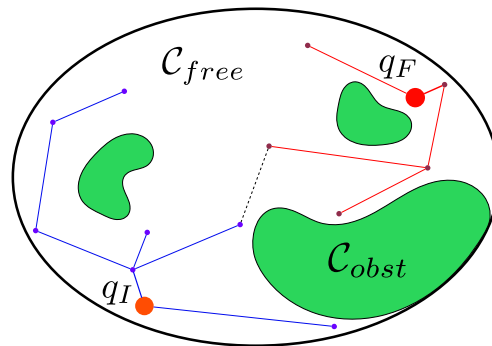


FIGURE 2.13 – Requête de planification résolue par un algorithme de diffusion

peuvent pas être utilisées car elles sont trop coûteuses en temps de calcul. Les planificateurs probabilistes sont les plus adaptés pour résoudre dans des temps raisonnables des problèmes de manipulation qui sont de grandes dimensions.

Ces méthodes ne sont pas suffisantes pour l'interaction Homme-robot. La présence de l'humain dans l'espace de travail \mathcal{W} doit être prise en compte dès l'étape de planification de chemin pour respecter les contraintes de sécurité et de confort.

2.2.3.2 Les planificateurs de chemin pour l'interaction Homme-robot

Sisbot a présenté récemment un état de l'art sur les planificateurs de chemin en présence de l'Homme [Sisbot 07]. Il distingue deux catégories principales :

- les planificateurs pour la navigation : le planificateur considère uniquement la base du robot ;
- les planificateurs pour la manipulation : le planificateur considère l'ensemble des degrés de liberté du robot ;

Cas de la navigation :

Les méthodes sont généralement développées pour des scénarios très spécifiques (e.g croisement du robot et de l'Homme dans un espace étroit, suivi de l'Homme, respect de règles sociales ou la navigation dans l'espace libre).

La plupart des systèmes prennent en compte l'Homme pendant l'exécution du mouvement. Pacchierotti et al. ont proposé une méthode pour éviter l'Homme s'il se trouve sur le chemin du robot en fonction de la distance Homme/robot et de la vitesse de l'Homme [Pacchierotti 06]. Yoshimi et al. ont présenté une méthode pour le suivi de personnes implémentée sur le robot *ApriAttenda*TM [Yoshimi 06]. Une extension de cette méthode qui permet de prendre en compte l'évitement d'obstacle a été proposée par Takemura et al. [Takemura 07]. Les travaux de Sisbot ont introduit, en plus de la notion de sécurité, des notions de visibilité du robot par l'Homme le long du chemin par l'intermédiaire de cartes de coût.

Cas de la manipulation :

Dans un contexte où le robot doit interagir avec l'Homme, par exemple lui apporter des objets, la présence de personnes dans l'environnement doit être prise en compte non seulement par la navigation mais aussi par la planification des chemins des autres corps du robot. Nokata et al. [Nokata 02], ont proposé de minimiser le critère de danger, précédemment introduit par Ikuta [Ikuta 03], pour générer des mouvements sûrs en prenant en compte uniquement l'organe effecteur du robot. Kulic a proposé

un planificateur de chemin minimisant ce même critère de danger le long du chemin en considérant le modèle complet du robot [Kulić 05]. Ce planificateur, intégré sur un bras manipulateur à six degrés de liberté, a permis de montrer que les mouvements planifiés engendrent moins d'anxiété et de surprise que les mouvements générés par des champs de potentiel [Khatib 86].

Sisbot et al. ont étendu leur précédente approche basée sur des cartes de coût pour la navigation à l'ensemble des degrés de liberté du robot. Le planificateur développé à partir de ces travaux est nommé *HAMP (Human Aware Motion Planner)* et est construit à partir du planificateur de chemin Move3D [Siméon 01].

Les travaux récents de Mainprice [Mainprice 11] reprennent les coûts définis par Sisbot. Ces coûts sont appliqués à la méthode de planification probabiliste par diffusion *T-RRT* [Jaillet 08].

Dans cette section nous avons survolé les méthodes de planification de chemin prenant en compte explicitement l'Homme. Le robot évoluant dans un environnement dynamique où les objets, les obstacles et les humains bougent, les mouvements doivent être adaptés en ligne pour augmenter la réactivité tout en satisfaisant les contraintes de sécurité et de confort. Une fois le chemin généré, il faut calculer l'évolution temporelle sur le chemin pour pouvoir l'exécuter ; c'est l'étape de la planification de la trajectoire.

2.2.4 La génération de trajectoire

Le problème de la génération de trajectoire est important dans de nombreux domaines comme les machines de production industrielle et les robots. L'ouvrage de Biagiotti et Melchiorri [Biagiotti 08] présente une synthèse des méthodes de génération de trajectoire pour le contexte industriel. Nous proposons également au lecteur de se reporter aux ouvrages de Khalil et de Dombre qui présentent des méthodes de génération de trajectoire pour les robots manipulateurs [Khalil 03, Dombre 07].

Les générateurs de trajectoires ont pour fonction de calculer l'évolution des positions des articulations ou de la situation du robot en fonction du temps. Cette trajectoire de référence définit l'entrée du système de contrôle (Section 2.1.2). Le générateur présente une grande importance car de la continuité de la trajectoire produite et de ses dérivées dépend la qualité du mouvement produit. Cette qualité impacte non seulement la précision de suivi du chemin mais aussi la durée de vie du système mécanique.

2.2.4.1 Les types de trajectoire

Les trajectoires peuvent être classées en plusieurs catégories (figure 2.14). La catégorie *Mono-dimensionnelle* correspond à la trajectoire d'un degré de liberté du robot. La trajectoire *Multi-dimensionnelle* est associée soit à un ensemble de degrés de liberté du robot (planification articulaire) soit à l'organe effecteur (planification cartésienne). Par opposition à la planification mono-dimensionnelle, la planification multi-dimensionnelle peut nécessiter de synchroniser temporellement l'ensemble des axes du robot. Les trajectoires qui sont définies par deux points (les conditions initiales et finales) sont des trajectoires point à point. Les trajectoires multi-points, quant-à-elles, passent par des points intermédiaires ou approximent un ensemble de points comme dans le cas des chemins (Section 2.2.1.3).

On distingue la génération de la trajectoire calculée dans l'espace articulaire du robot de celle dans l'espace cartésien. La planification dans l'espace articulaire présente plusieurs avantages :

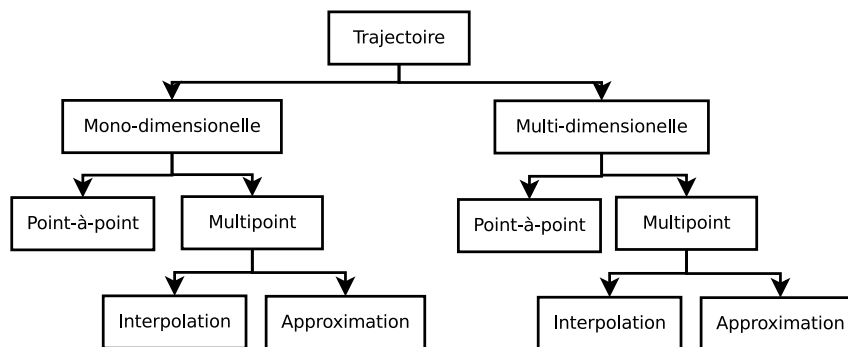


FIGURE 2.14 – Les catégories de trajectoires [Biagiotti 08]

- la planification dans l'espace articulaire n'utilise pas les fonctions d'inverse cinématique ou dynamique et est donc moins coûteuse en calcul ;
- la trajectoire articulaire permet de s'affranchir des problèmes liés aux configurations singulières et de la redondance des degrés de liberté.
- les contraintes cinématiques et dynamiques sont directement déduites des limites de chaque actionneur. Pour les trajectoires cartésiennes, ces contraintes nécessitent un calcul.

La planification de trajectoire peut se faire *hors-ligne* ou *en-ligne*. Par opposition aux méthodes hors lignes qui pré-calculent la trajectoire avant de l'exécuter, les méthodes en-ligne travaillent en temps réel. Ces dernières années, grâce à l'importante augmentation des capacités de calcul des ordinateurs, les recherches sur la génération de trajectoire se sont portées principalement sur les méthodes en-ligne.

2.2.4.2 La génération de trajectoire

Le problème de la génération de trajectoire pour des bras manipulateurs est largement traité dans la littérature [Brady 82, Khalil 99, Biagiotti 08]. Kröger, dans le chapitre 2 de son ouvrage [Kröger 10a], présente un état de l'art détaillé sur les méthodes de génération de trajectoire en-ligne et hors-ligne. Le chapitre 4 de l'ouvrage de Brady et al. [Brady 82] référence les travaux antérieurs aux années 80. Deux techniques de base de planification de trajectoire y sont présentées. La première est la planification sous contraintes cinématiques (limites en vitesse et accélération) entre deux positions. En fonction du nombre de ces contraintes imposés, un type de représentation de trajectoire est utilisé (courbe polynomiale de degré 2, cubique, de degré 4, quintique). La seconde famille de méthodes référencée dans cet ouvrage est la planification dans l'espace cartésien à partir d'un chemin donné [Paul 79, Paul 81] [Taylor 79]. Paul et Taylor proposent des lois d'accélération de type bang-bang. En 1984, Castain propose de changer la loi bang-bang par une loi trapèze et introduit un point intermédiaire entre les points initial et final [Castain 84]. Les trois courbes polynomiales cubiques qui définissent la loi trapèze d'accélération sont utilisées pour atteindre le point intermédiaire à vitesse constante depuis le point initial et trois autres courbes polynomiales sont utilisées pour relier le point intermédiaire et le point final. En utilisant cette méthode, sept courbes polynomiales cubiques sont utilisées pour relier les points du chemin. L'utilisation des lois d'accélération en trapèze introduit les mouvements bornés en jerk.

Le problème de la génération de trajectoire est généralement traité comme un problème de satisfaction de contraintes. Par exemple, une cubique peut être utilisée pour satisfaire des contraintes de position

et de vitesse ; une quintique peut être utilisée pour satisfaire des contraintes de position, de vitesse et d'accélération. Le critère le plus couramment utilisé est la minimisation du temps de parcours de la trajectoire, c'est la planification en *temps optimal*. Le calcul de la trajectoire optimale consiste à calculer la trajectoire pour satisfaire des contraintes cinématiques de la tâche [Lin 83, Chand 85] et/ou des contraintes dynamiques du robot [Hollerbach 83, Sahar 86, Geering 86]. Dans le milieu des années 80, d'autres travaux ont été menés sur la planification en temps optimal [Bobrow 85, Bobrow 88, Shin 85, Pfeiffer 87]. Ces travaux se basent sur un calcul du profil d'accélération en considérant la courbe maximale de vitesse admissible. Le profil d'accélération obtenu est composé de mouvements saturés en accélération. Slotine [Slotine 89] propose une méthode de calcul des instants de commutation du couple appliqué à chaque articulation, en considérant qu'au moins un actionneur a le couple saturé. Piazzi présente une méthode pour minimiser le jerk pendant le processus de génération de trajectoire dans l'espace articulaire [Piazzi 97].

D'autres techniques de génération de trajectoire sont basées sur des méthodes de filtrage non-linéaire [Libre 95, Gerelli 10]. Ces méthodes permettent de générer des mouvements à partir de consignes non admissibles et respectant des contraintes (par exemple cinématiques). Avec ces techniques, la trajectoire n'est connue qu'a posteriori.

La notion de *planification en temps optimal* varie en fonction des applications et de leurs contraintes. Pour les applications industrielles, où les robots évoluent à grande vitesse, cette technique de planification prendra en compte non seulement les contraintes cinématiques des actionneurs mais aussi les contraintes dynamiques nécessaires pour tenir compte des phénomènes inertiels.

Dans le contexte de nos travaux, la notion de *planification en temps optimal ou minimal* est associée aux contraintes cinématiques liées aux mouvements en présence de l'Homme. Ces contraintes sont de natures différentes et ne sont pas liées aux capacités du robot (actionneurs, inerties) mais à la tâche. En effet la présence de l'Homme au voisinage du robot entraîne, comme nous l'avons vu, des contraintes cinématiques.

En présence d'humain, les limites dynamiques ne sont pas atteintes, ce qui permet de traiter en temps réel les contraintes plus simples. Les méthodes de génération de trajectoires citées précédemment ne répondent pas aux critères imposés par l'interaction Homme-robot car elles sont hors ligne. Par la suite nous nous intéressons aux méthodes par satisfaction de contraintes cinématiques qui peuvent être utilisées en ligne.

2.2.4.3 La génération de trajectoire en-ligne par satisfaction de contraintes cinématiques

Kröger classe les différents types de générateur de trajectoire en ligne en fonction du nombre de contraintes cinématiques initiales, finales et sur le mouvement (tableau 2.3) [Kröger 10a]. Pour chaque type, les conditions initiales sont quelconques en vitesse et accélération. Les conditions finales en vitesse, accélération et jerk sont respectivement notées V_F , A_F et J_F . La première ligne (types I et II) correspond aux cas classiques de l'évolution en trapèze de la vitesse (contraintes maximales en vitesse et accélération). La deuxième ligne (types III, IV et V) correspond aux cas de l'évolution de l'accélération en trapèze et la troisième ligne (types VI, VII, VIII et IX), au jerk en trapèze. Les colonnes définissent les conditions finales du mouvement prises en compte par les générateurs. Nous utilisons cette classification pour situer nos travaux par rapport à l'état de l'art.

Andersson propose l'utilisation des polynômes d'ordre 5 (quintiques) pour résoudre le problème de

TABLE 2.3 – Les différents types de générateurs de trajectoire en ligne

	$V_F = 0$ $A_F = 0$ $J_F = 0$	$V_F \in \mathbb{R}$ $A_F = 0$ $J_F = 0$	$V_F \in \mathbb{R}$ $A_F \in \mathbb{R}$ $J_F = 0$	$V_F \in \mathbb{R}$ $A_F \in \mathbb{R}$ $J_F \in \mathbb{R}$
$A^{max} \in \mathbb{R}$	Type I	Type II	-	-
$A^{max} \in \mathbb{R}$ $J^{max} \in \mathbb{R}$	Type III	Type IV	Type V	-
$A^{max} \in \mathbb{R}$ $J^{max} \in \mathbb{R}$ $D^{max} \in \mathbb{R}$	Type VI	Type VII	Type VIII	Type IX

génération de trajectoire avec des conditions initiales et finales d'accélération, de vitesse et de position imposées et des temps de déplacement fixes [Andersson 89].

Bestaoui propose une méthode qui utilise des polynômes du quatrième degré avec des contraintes en vitesse et en accélération pour des applications en ligne par actualisation de la trajectoire à chaque périodes [Bestaoui 89]. Cette méthode est reprise par les travaux de [Bazaz 97].

Lloyd considère l'utilisation des polynômes de degré 5 pour des mouvements en ligne droite entre deux points et propose une méthode pour calculer les mouvements de transition aux points intermédiaires en utilisant une fonction polynomiale [Lloyd 93].

Llibre propose de définir les trajectoires en deux étapes. D'abord définir un chemin composé de courbes continues à l'ordre 2 (droites, cercles, interpolation entre trois points par deux quadriques, interpolation entre deux points par des polynômes de degré 5 ou des splines cubiques). Le type de chaque segment de courbe est choisi pour respecter la continuité d'ordre 2. Llibre propose ensuite de générer une loi d'évolution respectant également des contraintes en jerk, accélération et vitesse. La fonction décrivant la loi d'évolution est strictement croissante et est basée sur la méthode à jerk borné introduit par Castain [Castain 84].

Bazaz et Tondu introduisent une méthode à trois splines cubiques dans [Bazaz 98], une description plus détaillée de ces travaux est présentée dans [Tondu 99]. Ils proposent un générateur de trajectoire mono-dimensionnelle interpolant une suite de points de passage associés à des vitesses tout en limitant la vitesse et l'accélération maximale de la trajectoire. A chaque point de passage, l'accélération est nulle. L'algorithme développé est analytique pour être utilisé en ligne.

Liu [Liu 02] propose une méthode à sept segments cubiques pour un mouvement point à point (conditions initiale et finale en vitesse et accélération nulles). Les évolutions du jerk, de l'accélération et de la vitesse sont respectivement représentées par les fonctions $J(t)$, $A(t)$, $V(t)$ et $X(t)$ (figure 2.15). Cette méthode, traitant uniquement les trajectoires mono-dimensionnelles, permet de prendre en compte les contraintes en jerk J_{max} , accélération A_{max} et vitesse V_{max} (eq. 2.13).

$$\begin{aligned}
 |J(t)| &\leq J_{max} \\
 |A(t)| &\leq A_{max} \\
 |V(t)| &\leq V_{max}
 \end{aligned} \tag{2.13}$$

L'approche de Liu est basée sur la correspondance entre chaque position de la trajectoire point à point et le segment auquel elle appartient. C'est à dire que pendant l'exécution, l'algorithme de contrôle peut déterminer dans quel segment se situe l'état courant. Si la destination change pendant le déplacement, l'algorithme calcule un point virtuel à l'état d'arrêt (accélération et vitesse nulle) en amont de l'état courant de la trajectoire. Puis une nouvelle trajectoire point à point est calculée entre ce point virtuel et la nouvelle destination.

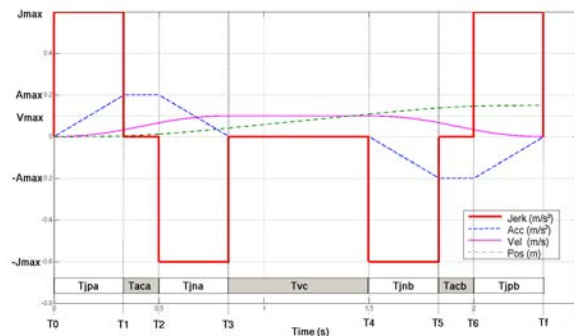


FIGURE 2.15 – Mouvement point à point composé de sept segments

Macfarlane propose de remplacer les sept segments de cubiques par des segments de quintiques [Macfarlane 03]. Cette extension est utilisée pour des mouvements en point à point dans l'espace cartésien. Pour une trajectoire, la méthode proposée considère des mouvements en ligne droite entre les points successifs et propose des fonctions polynomiales qui respectent les contraintes (jerk, accélération et vitesse) pour les relier et éviter le mouvement d'arrêt en chaque point. Nous détaillerons cette méthode dans la section 5.3.

Lambrechts [Lambrechts 05] augmente d'un degré en bornant le snap (dérivée du jerk) pour un mouvement point à point mono-dimensionnel (figure 2.16).

Des travaux récents basés sur les courbes polynomiales cubiques ont été mené par Herrera, Haschke et Kröger. Par rapport au tableau 2.3, les planificateurs introduits par Herrera et Haschke sont de type III.

Herrera et Sidobre [Herrera-Aguilar 05, Herrera-Aguilar 07] proposent, sur la base des travaux de Liu, sept cubiques pour planifier un *Mouvement Souple* pour les applications à la robotique de service. La méthode permet de prendre en compte des contraintes en jerk, accélération et vitesse pour un mouvements entre un point initial avec une vitesse et une accélération non nulle et un point final à l'arrêt (vitesse et accélération nulle). Cette approche a été validée sur un bras à six degré de liberté.

Haschke, Weitnauer et Ritter [Haschke 08] proposent un planificateur de trajectoire à jerk borné ayant les mêmes caractéristiques que celui de Herrera. Cette méthode permet de calculer un profil d'accélération en déterminant d'abord qualitativement le type du profil (cela permet de déterminer le nombre de segments de cubiques) puis quantitativement (e.g. résolution des équations correspondant au profil sélectionné). La figure 2.17 illustre les différents profils d'accélération dans le cas où deux profils en trapèze sont nécessaire. Ce profil étant déterminé sans prendre en compte la position finale (mais uniquement le signe), il est ensuite adapté en réduisant ou en augmentant les longueurs des segments d'accélération. Pour conserver la vitesse et l'accélération nulle au point final, cette modification est réalisée symétriquement par rapport au segment à vitesse constante. Dans ces travaux, la méthode est

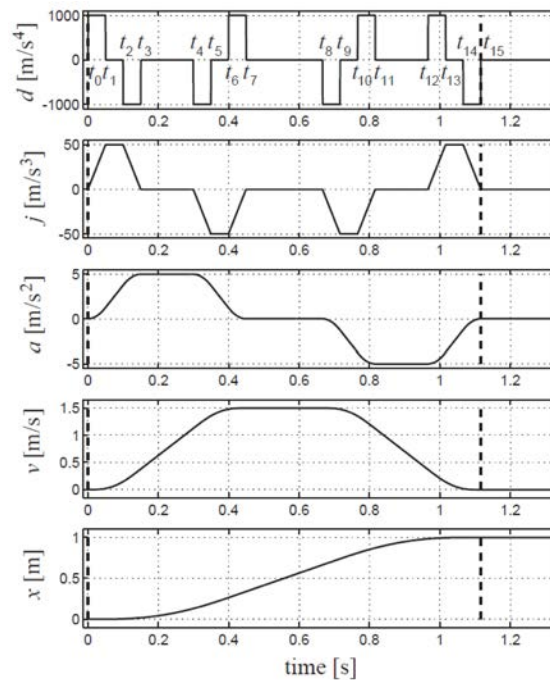


FIGURE 2.16 – Mouvement point à point composé de 15 segments de courbes de degré 4 [Lambrechts 05]

étendue au cas multidimensionnel pour la synchronisation des axes.

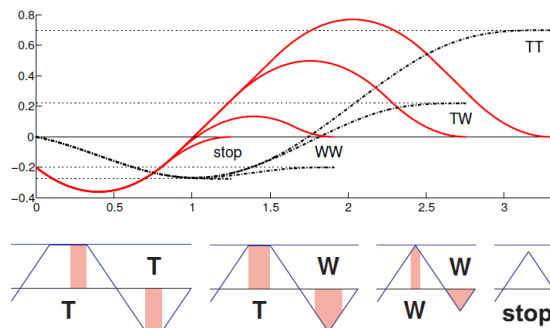


FIGURE 2.17 – Calcul du profil d'accélération par la méthode de calcul d'aire [Haschke 08]

Enfin, Kröger et al. proposent un générateur de trajectoire de type IV multi-dimensionnel, c'est à dire avec des conditions initiales cinématiques quelconques et des conditions finales avec une vitesse quelconque et une accélération nulle pour chaque dimension [Kröger 10b].

Nous remarquons que ces derniers travaux vont dans le sens des résultats de Flash et Hogan [Flash 84, Hogan 84]. En effet, les mouvements effectués par les humains étant limités en jerk et en accélération, un robot qui respecte ces deux limites aura un comportement acceptable par l'Homme.

2.3 Conclusion

Ce chapitre présente des méthodes pour la planification de mouvement. Ces dix dernières années des travaux ont été effectués sur la planification de chemin pour permettre de résoudre des tâches com-

plexes dans de grandes dimensions et dans des temps raisonnables. Des méthodes ont été introduites pour prendre en compte explicitement l'Homme dès le calcul du chemin afin de satisfaire les contraintes de sécurité et de confort. Des travaux montrent que pour respecter ces contraintes, le planificateur de trajectoire doit non seulement prendre en compte les limites de vitesse pour la sécurité mais aussi les limites d'accélération et de jerk pour le confort.

Un robot évoluant dans le même environnement que les humains doit adapter en ligne ses mouvements pour s'adapter aux mouvements des humains. Cependant les planificateurs actuels de trajectoire ne permettent pas de respecter ces contraintes. En effet, lors de la modification d'une trajectoire entre deux points appartenant à celle-ci, la nouvelle sous trajectoire aura des conditions initiale et finale en accélération et vitesse non nulle. Cette sous-trajectoire, selon le tableau 2.3, est de type V et les planificateurs actuels ne sont capables de calculer que des trajectoires de type IV. Dans le chapitre 3, nous proposons une méthode de type V permettant de satisfaire les contraintes imposées. Cette approche constituera la base de la méthode présentée dans le chapitre 5 concernant la planification et le contrôle d'exécution des trajectoires pour l'interaction Homme-robot.

3

La génération de trajectoire sous contraintes cinématiques

Comme nous l'avons vu dans le chapitre précédent, la présence de l'Homme dans l'espace de travail du robot introduit des contraintes de sécurité et de confort (Section 2.1.3). Pour respecter la contrainte de sécurité, la vitesse doit être bornée sur l'ensemble de la trajectoire. L'accélération et le jerk doivent également être bornés pour satisfaire la contrainte de confort. Pour représenter ce type de mouvement, une solution est d'utiliser des courbes polynomiales d'un degré supérieur ou égal à trois. Les trajectoires composées de suites de segments de courbes polynomiales cubiques répondent aux contraintes imposées.

Dans un contexte d'environnement dynamique, les trajectoires doivent être calculées en-ligne. En d'autres termes, le temps de calcul de la trajectoire doit être suffisamment petit pour permettre une interaction acceptable entre l'Homme et le robot. Des travaux récents ont été menés sur ce type de trajectoires [Herrera-Aguilar 07, Haschke 08, Kröger 10b] mais aucun ne traitent le cas général des trajectoires de type V (tableau 2.3, conditions initiales et finales en accélération et vitesse non nulles et contrainte en jerk, accélération et vitesse). Dans ce chapitre, nous présentons le premier, à notre connaissance, planificateur de trajectoires de type V. Ce générateur est basé sur des suites de segments de courbes polynomiales cubiques. Ces travaux ont été initialement publiés dans IROS (*IEEE/RSJ International Conference on Intelligent Robots and Systems*) [Broquère 08]. Nous introduisons le chapitre par la présentation du modèle de trajectoire utilisé et par une description de l'espace accélération-vitesse.

3.1 Modèle de trajectoire pour l'interaction Homme-robot

3.1.1 Notations

Dans la section 2.2, nous avons introduit les notations utilisées dans le domaine de la planification de chemin. Pour faciliter la compréhension des développements, nous introduisons ici les notations utilisées pour présenter la génération de trajectoire de systèmes quelconques de n degrés de liberté. Soit $\mathbf{X}_i \in \mathbb{R}^n$ un point dans un espace de dimension n décrivant la configuration du système à un instant t_i . Le point X_i est représenté par un vecteur de scalaires de dimension n :

$$\vec{X}_i = [{}_1X_i \ {}_2X_i \ \dots \ {}_nX_i]^T \quad (3.1)$$

De même pour la vitesse, l'accélération et le jerk :

$$\vec{V}_i = [{}_1V_i \ {}_2V_i \ \dots \ {}_nV_i]^T \quad (3.2)$$

$$\vec{A}_i = [{}_1A_i \ {}_2A_i \ \dots \ {}_nA_i]^T \quad (3.3)$$

$$\vec{J}_i = [{}_1J_i \ {}_2J_i \ \dots \ {}_nJ_i]^T \quad (3.4)$$

Dans le cas où le système est mono-dimensionnel ($n = 1$), \vec{X}_i , \vec{V}_i , \vec{A}_i et \vec{J}_i sont des scalaires.

3.1.2 Des suites de segments de courbes cubiques pour représenter les trajectoires

Nous choisissons de représenter la trajectoire $\text{TR}(t)$ par une combinaison de n suites de courbes polynomiales cubiques car ce type de trajectoire permet de limiter le jerk, l'accélération et la vitesse, tout en ayant une forme simple. La trajectoire ${}_j\text{TR}(t)$ correspond à l'évolution de l'axe j et est composée de N segments (figure 3.1).

Nous considérons, sans perte de généralité, que la division en segments est la même pour tous les axes. En effet, en considérant l'axe qui a le plus de segments, il est possible de diviser les segments des autres axes pour en obtenir le même nombre. Les fonctions ${}_jJ_k(t)$, ${}_jA_k(t)$, ${}_jV_k(t)$, ${}_jX_k(t)$ représentent respectivement les évolutions du jerk, de l'accélération, de la vitesse et de la position sur le segment k pour l'axe j .

Les segments sont définis par l'équation (3.5). Ils dépendent de leur durée T_k et de 5 paramètres :

- le temps initial t_{lk} avec $t_{lk} = t_l + \sum_{i=1}^{k-1} T_i$
- les conditions initiales (3 paramètres : ${}_jA_k(t_{lk})$, ${}_jV_k(t_{lk})$, ${}_jX_k(t_{lk})$);
- la valeur du jerk ${}_jJ_k$;

$\forall t \in [t_{lk}, t_{lk} + T_k]$:

$${}_jX_k(t) = \frac{{}_jJ_k}{6}(t - t_{lk})^3 + \frac{{}_jA_k(t_{lk})}{2}(t - t_{lk})^2 + {}_jV_k(t_{lk})(t - t_{lk}) + {}_jX_k(t_{lk}) \quad (3.5)$$

avec ${}_jJ_k$, ${}_jA_k(t_{lk})$, ${}_jV_k(t_{lk})$, ${}_jX_k(t_{lk})$ et t_{lk} des constantes $\in \mathbb{R}$.

Les conditions initiales de la trajectoire ${}_j\text{TR}(t)$ sont :

$$\begin{aligned} {}_jA_1(t_I) &= {}_jA_I \\ {}_jV_1(t_I) &= {}_jV_I \\ {}_jX_1(t_I) &= {}_jX_I \end{aligned} \quad (3.6)$$

et les conditions finales :

$$\begin{aligned} {}_jA_N(t_F) &= {}_jA_F \\ {}_jV_N(t_F) &= {}_jV_F \\ {}_jX_N(t_F) &= {}_jX_F \end{aligned} \quad (3.7)$$

avec $t_F - t_I = \sum_{i=1}^N T_i$.

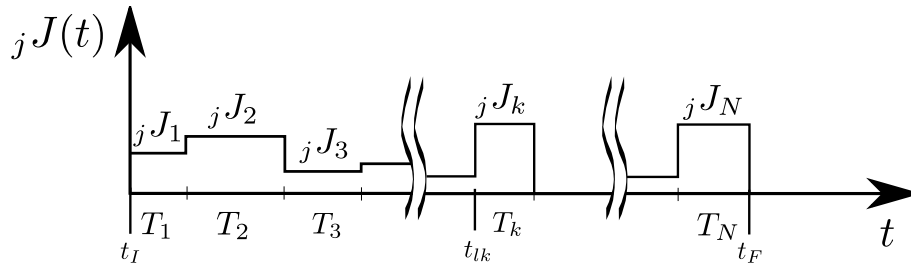


FIGURE 3.1 – Profil du jerk d'une suite de segments de courbes polynomiales cubiques associé à l'axe j d'une trajectoire ${}_j\text{TR}(t)$

La trajectoire multidimensionnelle est une composée des trajectoires de chaque axe :

$$\text{TR}(t) = [{}_1\text{TR}(t) \ {}_2\text{TR}(t) \ \dots \ {}_n\text{TR}(t)]^T \quad (3.8)$$

où n est le nombre d'axes.

Notons que la nature des axes j varie avec le type de courbe : translation ou rotation dans \mathbb{R}^3 , coordonnées articulaires ou combinaisons.

Les N couples $({}_jJ_k, T_k)$ et les conditions initiales (eq. 3.6) de la trajectoire ${}_j\text{TR}(t)$ permettent de calculer l'état de l'axe j à un instant donné avec les équations 3.9, 3.10 et 3.11. Pour simplifier l'écriture, l'indice j correspondant à l'axe considéré sera souvent omis dans l'écriture de ces équations. Le lecteur se reportera à l'annexe A pour les détails des calculs.

$\forall t \in [t_{lk}, t_{lk} + T_k]$, avec $t_I = 0$:

$$A_k(t) = J_k \left(t - \sum_{i=1}^{k-1} T_i \right) + \sum_{i=1}^{k-1} J_i T_i + A_I \quad (3.9)$$

$$V_k(t) = \frac{J_k}{2} \left(t - \sum_{i=1}^{k-1} T_i \right)^2 + \sum_{i=1}^{k-1} J_i T_i \left(t - \sum_{j=1}^i T_j \right) + \sum_{i=1}^{k-1} \frac{J_i T_i^2}{2} + A_I t + V_I \quad (3.10)$$

$$X_k(t) = \frac{J_k}{6} \left(t - \sum_{i=1}^{k-1} T_i \right)^3 + \sum_{i=1}^{k-1} \frac{J_i T_i}{2} \left(t - \sum_{j=1}^i T_j \right)^2 + \sum_{i=1}^{k-1} \frac{J_i T_i^2}{2} \left(\sum_{j=1}^i T_j \right) + \sum_{i=1}^{k-1} \frac{J_i T_i^3}{6} + \frac{A_I}{2} t^2 + V_I t + X_I \quad (3.11)$$

3.1.3 Les contraintes cinématiques

La méthode de planification proposée dans ces travaux est basée sur la satisfaction de contraintes cinématiques (vitesse, accélération et jerk). Ces contraintes sont supposées constantes pour un mouvement donné. Dans le cas multi-dimensionnel, chaque axe pourra avoir des contraintes différentes. Nous avons également supposé que ces contraintes sont symétriques :

$$\begin{aligned} {}_j J_{min} &= -{}_j J_{max} \\ {}_j A_{min} &= -{}_j A_{max} \\ {}_j V_{min} &= -{}_j V_{max} \end{aligned} \quad (3.12)$$

Les évolutions du jerk, de l'accélération et de la vitesse devront donc respecter :

$$\begin{aligned} |{}_j J(t)| &\leq {}_j J_{max} \\ |{}_j A(t)| &\leq {}_j A_{max} \\ |{}_j V(t)| &\leq {}_j V_{max} \end{aligned} \quad (3.13)$$

3.1.4 Notion de trajectoire en temps minimal

Par la suite, la notion de *de mouvement en temps optimal ou minimal* est associée aux contraintes cinématiques (jerk, accélération et vitesse). Dans le cas d'un mouvement mono-dimensionnel en temps minimal, le mouvement est sous contraintes saturées ; à chaque instant, une des contraintes est saturée. Dans le cas du mouvement multidimensionnel en temps minimal, au moins un axe évolue sous contraintes saturées. Dans nos travaux, sans perte de généralité, nous avons également supposé que les contraintes sont admissibles sur tout l'espace de travail du robot.

3.1.5 Le mouvement dans le plan accélération-vitesse

L'algorithme décrit dans la suite de ce chapitre est basé sur des calculs géométriques directement représentables dans le plan accélération-vitesse (figure 3.2).

L'axe horizontal de ce plan est la vitesse et l'axe vertical est l'accélération. La position n'est pas représentée. Pour un axe considéré, le point A correspond à l'état d'arrêt (pas de mouvement : accélération

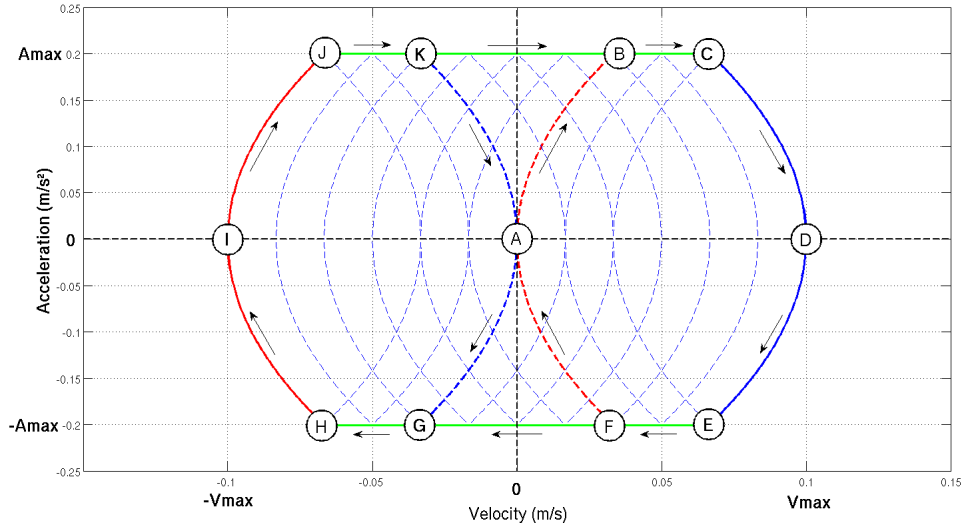


FIGURE 3.2 – Plan accélération-vitesse

et vitesse nulles). Les lignes *JC* et *EH* définissent respectivement les états à accélération maximale (A_{max}) et minimale ($-A_{max}$). Le système peut rester indéfiniment sur un point de la ligne *IAD* en raison de l'accélération nulle. La vitesse est maximale (V_{max}) au point *D* et minimale ($-V_{max}$) en *I*. Les autres états du système ne sont pas des états d'équilibres, comme par exemple, à partir du point *C* où la seule évolution possible est de rejoindre le point *D*. La parabole *HIJ* représente une évolution à jerk maximal J_{max} tandis que la parabole *CDE* représente une évolution à jerk minimal ($-J_{max}$).

Les paraboles à jerk saturé ont pour équations :

$$V(t) = V_0 \pm \frac{1}{2J_{max}} A(t)^2 \tag{3.14}$$

où $V_0 \in [-V_{max}, V_{max}]$ est la vitesse quand $A(t) = 0$.

Dans ce plan, seulement trois types de mouvements élémentaires sont nécessaires pour définir n'importe quel mouvement. Ils correspondent aux évolutions à vitesse, accélération et jerk saturés ($t_I = 0$ pour chaque segment) :

Jerk saturé $\pm J_{max}$ segments <i>AB, CD, DE, FA, AG, HI, IJ</i> et <i>KA</i>)	Accélération saturée $\pm A_{max}$ segments <i>BC, EF, GH</i> et <i>JK</i>	Vitesse saturée $\pm V_{max}$ points <i>D</i> et <i>I</i>
$J(t) = \pm J_{max}$ $A(t) = A_i \pm J_{max}t$ $V(t) = V_i + A_i t \pm \frac{1}{2} J_{max} t^2$ $X(t) = X_i + V_i t + \frac{1}{2} A_i t^2 \pm \frac{1}{6} J_{max} t^3$	$J(t) = 0$ $A(t) = A_i = \pm A_{max}$ $V(t) = V_i \pm A_{max} t$ $X(t) = X_i + V_i t \pm \frac{1}{2} A_{max} t^2$	$J(t) = 0$ $A(t) = 0$ $V(t) = V_i = \pm V_{max}$ $X(t) = X_i \pm V_{max} t$

3.1.6 Symétrie du plan accélération-vitesse

Considérons un mouvement quelconque représenté dans le plan accélération-vitesse (figure 3.3) avec pour conditions initiales et finales :

$$\begin{aligned} A(t_I) &= A_{I1} & A(t_F) &= A_{F1} \\ V(t_I) &= V_{I1} & V(t_F) &= A_{F1} \\ X(t_I) &= 0 & X(t_F) &= X_{F1} \end{aligned}$$

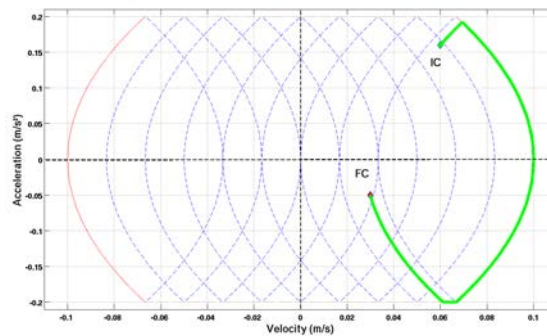


FIGURE 3.3 – Un mouvement représenté dans le plan accélération-vitesse

Si l'on inverse les conditions initiales et finales telles que :

$$\begin{aligned} A(t_I) &= -A_{I1} & A(t_F) &= -A_{F1} \\ V(t_I) &= -V_{I1} & V(t_F) &= -A_{F1} \\ X(t_I) &= 0 & X(t_F) &= -X_{F1} \end{aligned}$$

Le mouvement obtenu sera un mouvement symétrique par rapport au point A du plan accélération-vitesse en raison de la symétrie des contraintes cinématiques (eq. 3.12). Ce mouvement est représenté par la figure 3.4.

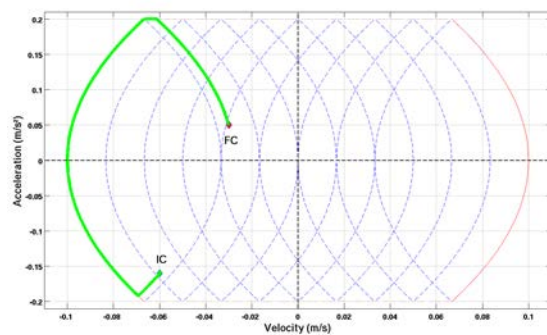


FIGURE 3.4 – Le mouvement illustrant la symétrie dans le plan accélération-vitesse

Cette propriété sera utilisée par l'algorithme général de calcul du mouvement de type V (section 3.3.4).

3.2 Les trajectoires mono-dimensionnelles point à point en temps minimal

Avant de présenter le planificateur que nous avons développé dans le cas général, nous introduisons la méthode de calcul d'un mouvement point à point mono-dimensionnel en temps minimal sous contraintes cinématiques. Cette méthode a déjà été présentée dans la littérature [Liu 02, Herrera-Aguilar 07] et nous proposons au lecteur de se reporter à ces travaux pour les détails des calculs.

Dans ce cas, les conditions initiales et finales sont définies par :

$$\begin{aligned} A(t_I) &= 0 & A(t_F) &= 0 \\ V(t_I) &= 0 & V(t_F) &= 0 \\ X(t_I) &= 0 & X(t_F) &= X_F \end{aligned}$$

La figure 3.5 représente un mouvement optimal (par rapport aux contraintes cinématiques) qui se compose au plus de sept segments :

- $T_{jpa} = t_1 - t_I$ Temps à Jerk positif
- $T_{aca} = t_2 - t_1$ Temps à Accélération constante
- $T_{jna} = t_3 - t_2$ Temps à Jerk négatif
- $T_{vc} = t_4 - t_3$ Temps à Vitesse constante
- $T_{jnb} = t_5 - t_4$ Temps à Jerk négatif
- $T_{acb} = t_6 - t_5$ Temps à Accélération constante
- $T_{jpb} = t_F - t_6$ Temps à Jerk positif

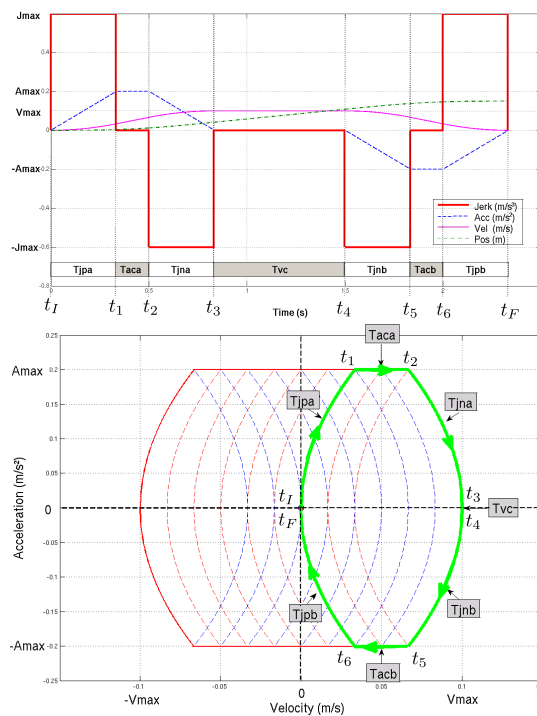


FIGURE 3.5 – Les courbes du Jerk, de l'accélération, de la vitesse et de la position en fonction du temps (haut) et le mouvement dans le plan accélération-vitesse (bas)

Le système d'équation à résoudre est le suivant :

$$\begin{aligned}
 & A_1(t) = \text{dir}J_{\max}(t - t_I) + A_I \\
 t \in [t_I, t_1] \quad & V_1(t) = \text{dir}\frac{J_{\max}}{2}(t - t_I)^2 + A_I(t - t_I) + V_I \\
 & X_1(t) = \text{dir}\frac{J_{\max}}{6}(t - t_I)^3 + \frac{1}{2}A_I(t - t_I)^2 + V_I(t - t_I) + X_I \\
 & A_2(t) = A_1(t_1) \\
 t \in [t_1, t_2] \quad & V_2(t) = A_1(t_1)(t - t_1) + V_1(t_1) \\
 & X_2(t) = \frac{1}{2}A_1(t_1)(t - t_1)^2 + V_1(t_1)(t - t_1) + X_1(t_1) \\
 & A_3(t) = -\text{dir}J_{\max}(t - t_2) + A_2(t_2) \\
 t \in [t_2, t_3] \quad & V_3(t) = -\text{dir}\frac{J_{\max}}{2}(t - t_2)^2 + A_2(t_2)(t - t_2) + V_2(t_2) \\
 & X_3(t) = -\text{dir}\frac{J_{\max}}{6}(t - t_2)^3 + \frac{1}{2}A_2(t_2)(t - t_2)^2 + V_2(t_2)(t - t_2) + X_2(t_2) \\
 & A_4(t) = 0 \\
 t \in [t_3, t_4] \quad & V_4(t) = V_3(t_3) \\
 & X_4(t) = X_3(t_3) + V_3(t_3)(t - t_3) \\
 & A_5(t) = -\text{dir}J_{\max}(t - t_4) + A_4(t_4) \\
 t \in [t_4, t_5] \quad & V_5(t) = -\text{dir}\frac{J_{\max}}{2}(t - t_4)^2 + A_4(t_4)(t - t_4) + V_4(t_4) \\
 & X_5(t) = -\text{dir}\frac{J_{\max}}{6}(t - t_4)^3 + \frac{1}{2}A_4(t_4)(t - t_4)^2 + V_4(t_4)(t - t_4) + X_4(t_4) \\
 & A_6(t) = A_5(t_5) \\
 t \in [t_5, t_6] \quad & V_6(t) = A_5(t_5)(t - t_5) + V_5(t_5) \\
 & X_6(t) = \frac{1}{2}A_5(t_5)(t - t_5)^2 + V_5(t_5)(t - t_5) + X_5(t_5) \\
 & A_7(t) = \text{dir}J_{\max}(t - t_6) + A_6(t_6) \\
 t \in [t_6, t_F] \quad & V_7(t) = \text{dir}\frac{J_{\max}}{2}(t - t_6)^2 + A_6(t_6)(t - t_6) + V_6(t_6) \\
 & X_7(t) = \text{dir}\frac{J_{\max}}{6}(t - t_6)^3 + \frac{1}{2}A_6(t_6)(t - t_6)^2 + V_6(t_6)(t - t_6) + X_6(t_6)
 \end{aligned} \tag{3.15}$$

avec $\text{dir} = \text{sign}(X_F - X_I)$.

Le cas particulier du mouvement point à point avec des contraintes cinématiques symétriques introduit une antisymétrie pour l'accélération et une symétrie pour le jerk par rapport au segment à vitesse constante T_{vc} . Cette symétrie apparaît aussi pour la courbe de vitesse. Ainsi, seulement trois durées de segment sont inconnues :

$$\boxed{T_j = T_{jpa} = T_{jna} = T_{jnb} = T_{jpb} \quad T_a = T_{aca} = T_{acb} \quad T_{vc}}$$

Dans ce cas, le système calcule les trois temps T_j , T_a et T_{vc} pour obtenir le déplacement désiré entre les positions initiale et finale. La planification s'effectuant sous contraintes (J_{\max} , A_{\max} et V_{\max}), nous obtenons un *mouvement en temps minimal sous contraintes cinématiques*.

A partir des contraintes cinématiques, nous pouvons déduire les temps T_{jmax} et T_{amax} des évolutions à jerk et à accélération saturées lorsque A_{\max} et V_{\max} sont atteints :

$$T_{jmax} = \frac{A_{\max}}{J_{\max}} \quad T_{amax} = \frac{V_{\max}}{A_{\max}} - \frac{A_{\max}}{J_{\max}} \tag{3.16}$$

En fonction du déplacement $D = X_F - X_I$ à effectuer, on distingue trois types de mouvement. Le premier cas, lorsque D est suffisamment long pour que V_{\max} soit atteint. Dans ce cas, nous avons $T_a = T_{amax}$

et $T_j = T_{jmax}$. La durée du segment à vitesse constante T_{vc} est fonction de la distance à parcourir . Dans le deuxième cas, l'accélération maximale est seulement atteinte et pour le troisième cas, ni l'accélération maximale, ni la vitesse maximale ne sont atteintes.

Pour séparer ces trois mouvements, deux seuils de distance à parcourir sont calculés. Le premier D_{thr1} correspond au mouvement où V_{max} est atteint avec $T_{vc} = 0$:

$$D_{thr1} = \frac{A_{max}V_{max}}{J_{max}} + \frac{V_{max}^2}{A_{max}} \quad (3.17)$$

Le second seuil D_{thr2} correspond au mouvement où l'accélération maximale est atteinte avec $T_a = 0$:

$$D_{thr2} = 2\frac{A_{max}^3}{J_{max}^2} \quad (3.18)$$

Le calcul du mouvement mono-dimensionnel point à point en temps minimal est effectué par l'algorithme 1, basé sur les équations 3.17 et 3.18 et sur la définition des trois types de mouvement.

Algorithme 1 : Mouvement mono-dimensionnel point à point en temps minimal

début

Calcul de la distance $D = X_F - X_I$

Calcul des durées T_{jmax} et T_{amax}

Calcul des seuils D_{thr1} et D_{thr2}

si $D \geq D_{thr1}$ **alors**

$$\begin{array}{l} T_j = T_{jmax} \quad T_a = T_{amax} \\ T_{vc} = \frac{D - D_{thr1}}{V_{max}} \end{array}$$

sinon si $D \geq D_{thr2}$ **alors**

$$\begin{array}{l} T_{vc} = 0 \quad T_j = T_{jmax} \\ T_a = \sqrt{\frac{A_{max}^2}{4(J_{max})^2} + \frac{D}{A_{max}}} - \frac{3A_{max}}{2J_{max}} \end{array}$$

sinon

$$\begin{array}{l} T_{vc} = 0 \quad T_a = 0 \\ T_j = \sqrt[3]{\frac{D}{2J_{max}}} \end{array}$$

fin

Ainsi la durée totale du mouvement T_F est :

$$\begin{aligned} T_F &= t_F - t_I \\ &= 4T_j + 2T_a + T_{vc} \end{aligned} \quad (3.19)$$

Remarque : Les segments à accélération saturée n'apparaissent que si la relation 3.20 est vérifiée. Cette relation est directement déduite de la relation 3.16.

$$\frac{V_{max}}{A_{max}} > \frac{A_{max}}{J_{max}} \quad (3.20)$$

Dans la suite, nous considérons essentiellement des situations où les contraintes cinématiques respectent

cette relation.

3.3 Les trajectoires mono-dimensionnelles de type V en temps minimal

Contrairement au mouvement point à point, le calcul du mouvement dans le cas général de type V est beaucoup plus complexe car les symétries n'apparaissent plus.

Les conditions initiales *IC* et finales *FC* de ce type de mouvement sont :

$$\begin{aligned} A(t_I) &= A_I & A(t_f) &= V_F \\ V(t_I) &= V_I & V(t_f) &= A_F \\ X(t_I) &= X_I & X(t_f) &= X_F \end{aligned} \tag{3.21}$$

3.3.1 Les types de mouvements

Considérons un mouvement uniquement composé de mouvements élémentaires saturés. Supposons également que la distance à parcourir est suffisamment longue pour que le mouvement comporte un mouvement élémentaire à vitesse constante saturée. Un tel mouvement est en temps minimal et passe par le point *D* ou par le point *I* du plan accélération-vitesse.

A partir d'un état quelconque, deux mouvements sont possibles. Si la distance à parcourir est positive, le mouvement passe par le point *D*. Inversement, si la distance à parcourir est négative, le mouvement passe par le point *I*.

Nous distinguons donc deux types de mouvements :

- Les mouvements débutant par un segment à jerk maximum pour rejoindre le point *D*. Le point *D* est atteint par au plus 3 déplacements élémentaires : celui à jerk maximum, suivi éventuellement d'un mouvement à accélération maximum puis un mouvement à jerk minimum. Nous qualifions ces **mouvement de type 1** (figure 3.6)
- Les mouvements débutant par un segment à jerk minimum pour rejoindre le point *I*. Le point *I* est atteint par au plus 3 déplacements élémentaires : celui à jerk minimal, suivi éventuellement d'un mouvement à accélération minimum puis un mouvement à jerk maximum. Nous qualifions ces **mouvement de type 2** (figure 3.7)

Dans le cas général, un mouvement comporte au plus sept segments [Herrera-Aguilar 07] :

Segments	Mouvement de type 1	Mouvement de type 2
1	jerk maximum	jerk minimum
2	accélération maximum	accélération minimum
3	jerk minimum	jerk maximum
4	vitesse maximum	vitesse minimum
5	jerk minimum	jerk maximum
6	accélération minimum	accélération maximum
7	jerk maximum	jerk minimum

Lorsque la distance à parcourir diminue de telle sorte que le mouvement n'atteint plus la vitesse maximale, le mouvement comporte moins de sept segments.

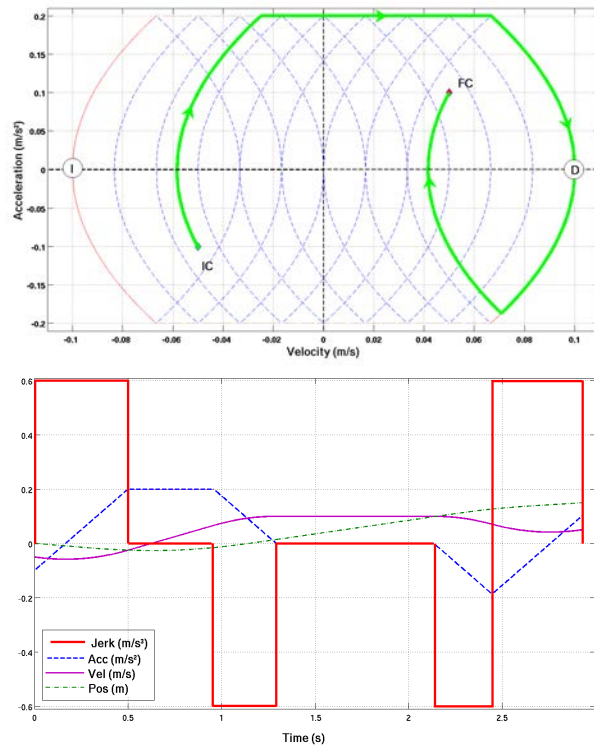


FIGURE 3.6 – Mouvement de type 1 avec V_{max} atteint

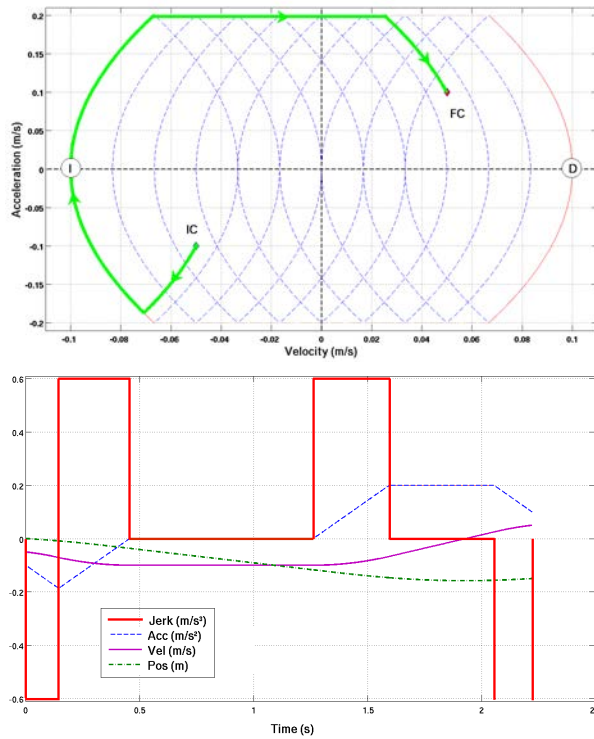


FIGURE 3.7 – Mouvement de type 2 avec $-V_{max}$ atteint

3.3.2 Les types de mouvements composés de moins de sept segments

La composition des segments de mouvement varie en fonction des conditions cinématiques initiales et finales et de la distance à parcourir. L'ensemble de ces types sont décrits dans la section suivante (Section 3.3.4). Un mouvement peut être, par exemple, composé d'un seul segment lorsque les conditions initiales et finales sont nulles en accélération et égales en vitesse (V_{max} ou V_{min}).

Comme nous l'avons introduit dans le paragraphe précédent, le mouvement sera de type 1 ou de type 2, en fonction de la distance à parcourir. Lorsque cette distance diminue les accélérations maximales et minimales ne sont pas forcément atteintes.

Nous pouvons isoler un mouvement remarquable associé à une longueur particulière pour chaque paire de conditions initiales et finales du plan vitesse-accélération. Ce mouvement particulier sépare les mouvements de type 1 de ceux de types 2. La figure 3.8 illustre ce mouvement pour les conditions initiales et finales des figures 3.6 et 3.7. Nous appellerons ce mouvement : *Mouvement critique*. Il correspond à une distance particulière à parcourir, la *distance critique*.

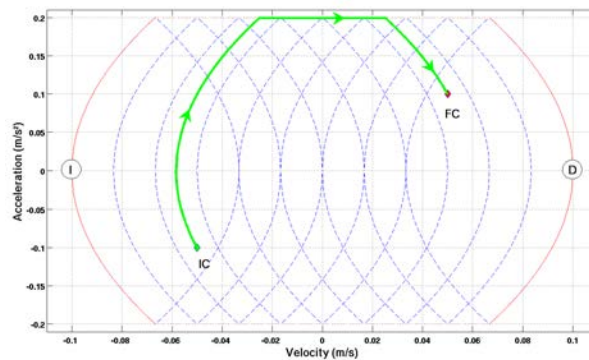


FIGURE 3.8 – Un *Mouvement critique*

3.3.3 La distance critique : définition et calculs

La *distance critique* dc est une longueur caractéristique pour le calcul du mouvement avec des conditions cinématiques initiales et finales non nulles. La définition introduite par Herrera ne traite pas l'ensemble des cas existants notamment quand le mouvement n'atteint ni $-V_{max}$ ni V_{max} . Nous proposons une définition générale de la distance critique :

3.3.3.1 Définition

La distance critique dc est la distance effectuée entre les conditions initiales IC ((V_I, A_I)) et les conditions finales FC ((A_F, V_F)) définie de telle sorte que le temps de déplacement soit minimal. Le mouvement critique sépare continument les *mouvements de type 1* des *mouvements de type 2*.

Lorsque la distance à parcourir devient supérieure à dc , on bascule sur un mouvement de type 1 et inversement lorsque la distance à parcourir est inférieure à dc , on bascule sur un mouvement de type 2.

3.3.3.2 Calculs

Les données nécessaires pour le calcul de la distance critique dc sont :

$$\begin{aligned} A(t_I) &= A_I & A(t_F) &= A_F \\ V(t_I) &= V_I & V(t_F) &= V_F \end{aligned}$$

Les positions initiale et finale n'interviennent pas dans le calcul de la distance critique. Dans la suite du manuscrit, l'état initial est aussi représenté par le point P_1 et l'état final par le point P_2 . Nous avons divisé le plan accélération-vitesse en sept zones. La figure 3.9 illustre cette décomposition du plan pour les conditions initiales particulières $IC(V_I, A_I)$.

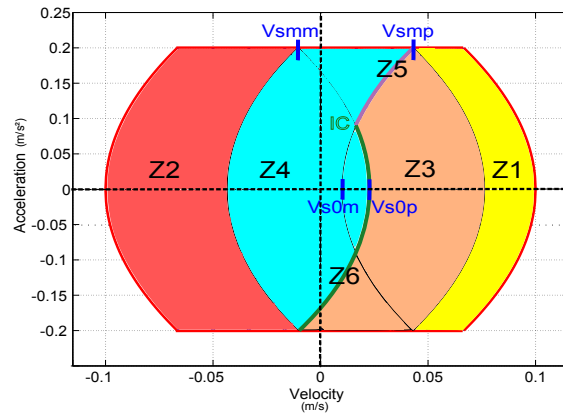


FIGURE 3.9 – Une décomposition du plan accélération-vitesse en sept zones

Le premier cas à prendre en compte est celui avec des conditions initiales et finales identiques. Ce cas correspond à la **zone 0 (Z0)**. Dans ce cas la distance critique est nulle ($dc = 0$). Remarquons que le mouvement point à point ($A_I = A_F = V_I = V_F = 0$) rentre dans ces conditions. Donc pour effectuer un déplacement positif, le mouvement doit commencer par un segment à jerk positif pour éventuellement rejoindre la vitesse maximale ; et inversement pour effectuer un déplacement négatif.

Chaque zone est définie par rapport aux conditions initiales $IC(V_I, A_I)$. Pour définir ces zones, nous introduisons des points particuliers du plan accélération-vitesse (V_{smm}, A_{max}) , (V_{smp}, A_{max}) , $(V_{s0m}, 0)$ et $(V_{s0p}, 0)$ (figure 3.9).

La signification des abréviations relatives aux vitesses est :

- V_{smp} = Vitesse atteinte lorsque l'accélération est maximale sur la parabole à jerk positif passant par $IC(V_I, A_I)$;
- V_{smm} = Vitesse atteinte lorsque l'accélération est maximale sur la parabole à jerk négatif passant par $IC(V_I, A_I)$;
- V_{s0p} = Vitesse atteinte lorsque l'accélération est nulle sur la parabole à jerk négatif passant par $IC(V_I, A_I)$;
- V_{s0m} = Vitesse atteinte lorsque l'accélération est nulle sur la parabole à jerk positif passant par $IC(V_I, A_I)$;

Pour effectuer le calcul de la distance critique, nous définissons également les points particuliers (V_{fmm}, A_{max}) , (V_{fmp}, A_{max}) , $(V_{f0m}, 0)$ et $(V_{f0p}, 0)$ pour les conditions finales $FC(V_F, A_F)$ (figure 3.10).

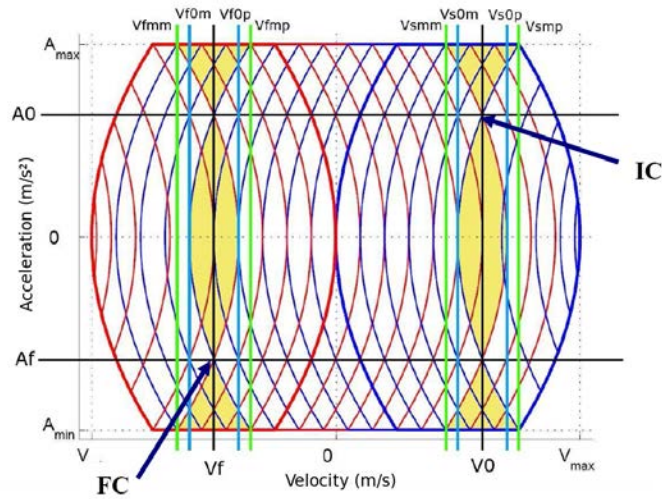


FIGURE 3.10 – Les points particuliers du plan accélération-vitesse pour des conditions initiales IC et finales FC données

La signification des abréviations relatives aux vitesses est :

- V_{fmp} = Vitesse atteinte lorsque l'accélération est maximale sur la parabole à jerk positif passant par $FC(V_F, A_F)$;
- V_{fmm} = Vitesse atteinte lorsque l'accélération est maximale sur la parabole à jerk négatif passant par $FC(V_F, A_F)$;
- V_{f0p} = Vitesse atteinte lorsque l'accélération est nulle sur la parabole à jerk négatif passant par $FC(V_F, A_F)$;
- V_{f0m} = Vitesse atteinte lorsque l'accélération est nulle sur la parabole à jerk positif passant par $FC(V_F, A_F)$;

Le calcul de la longueur dc dépend de la zone où se situe le point (V_F, A_F) par rapport au point (V_I, A_I) . Pour cela il faut calculer et comparer, les points (V_{smm}, A_{max}) , (V_{smp}, A_{max}) , (V_{fmm}, A_{max}) , (V_{fmp}, A_{max}) , $(V_{s0m}, 0)$, $(V_{s0p}, 0)$, $(V_{f0m}, 0)$ et $(V_{f0p}, 0)$ définis plus haut.

Nous définissons V_{lim} (figure 3.11) la vitesse atteinte lorsque l'accélération est nulle pour la parabole à jerk positif passant par le point C du plan accélération-vitesse (figure 3.2). Cette vitesse est calculée par la relation 3.22

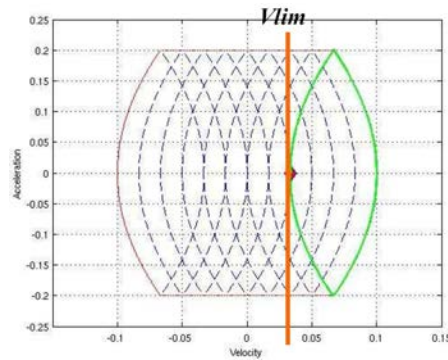
$$V_{lim} = V_{max} - \frac{A_{max}^2}{J_{max}} \quad (3.22)$$

La suite de segments composant le mouvement critique change avec la zone. La figure 3.12 résume les différents cas qui sont détaillés dans l'annexe B.

Remarque :

Pour certaines conditions initiales et finales particulières, il existe un mouvement plus court qui court-circuite le mouvement critique. La figure 3.13 en illustre un cas particulier. Pour rejoindre les conditions initiales et finales, il existe un mouvement en temps minimal composé de deux segments (jerk positif puis négatif) (noir sur la figure). Le mouvement critique est représenté par la courbe jaune.

Dans cet exemple, les conditions initiales et finales sont :

FIGURE 3.11 – La vitesse V_{lim} dans le plan accélération-vitesse

$$\begin{aligned} A(t_I) &= 0.2m.s^{-2} & A(t_F) &= 0.25m.s^{-2} \\ V(t_I) &= 0.021m.s^{-1} & V(t_F) &= 0.043m.s^{-1} \end{aligned}$$

Pour effectuer la distance particulière de 0.002908 m, il existe deux solutions (figure 3.14). La première est la solution en temps minimal (courbe noire, constituée de deux segments de jerk saturé). La seconde est un mouvement passant par les états à vitesse négative (courbe verte), c'est celui qui est calculé par notre algorithme.

Ce mouvement particulier ne sera pas pris en compte par l'algorithme général pour les mouvements de type V présenté dans la section suivante. En effet, ce mouvement est seulement valide pour une longueur particulière (ici 0.002908 m). Si la distance à parcourir est augmentée ou diminuée, la durée du mouvement résultat sera très largement différente de celle de ce mouvement minimal. L'algorithme proposé ignore ces cas particuliers afin de ne pas introduire de discontinuité temporelle.

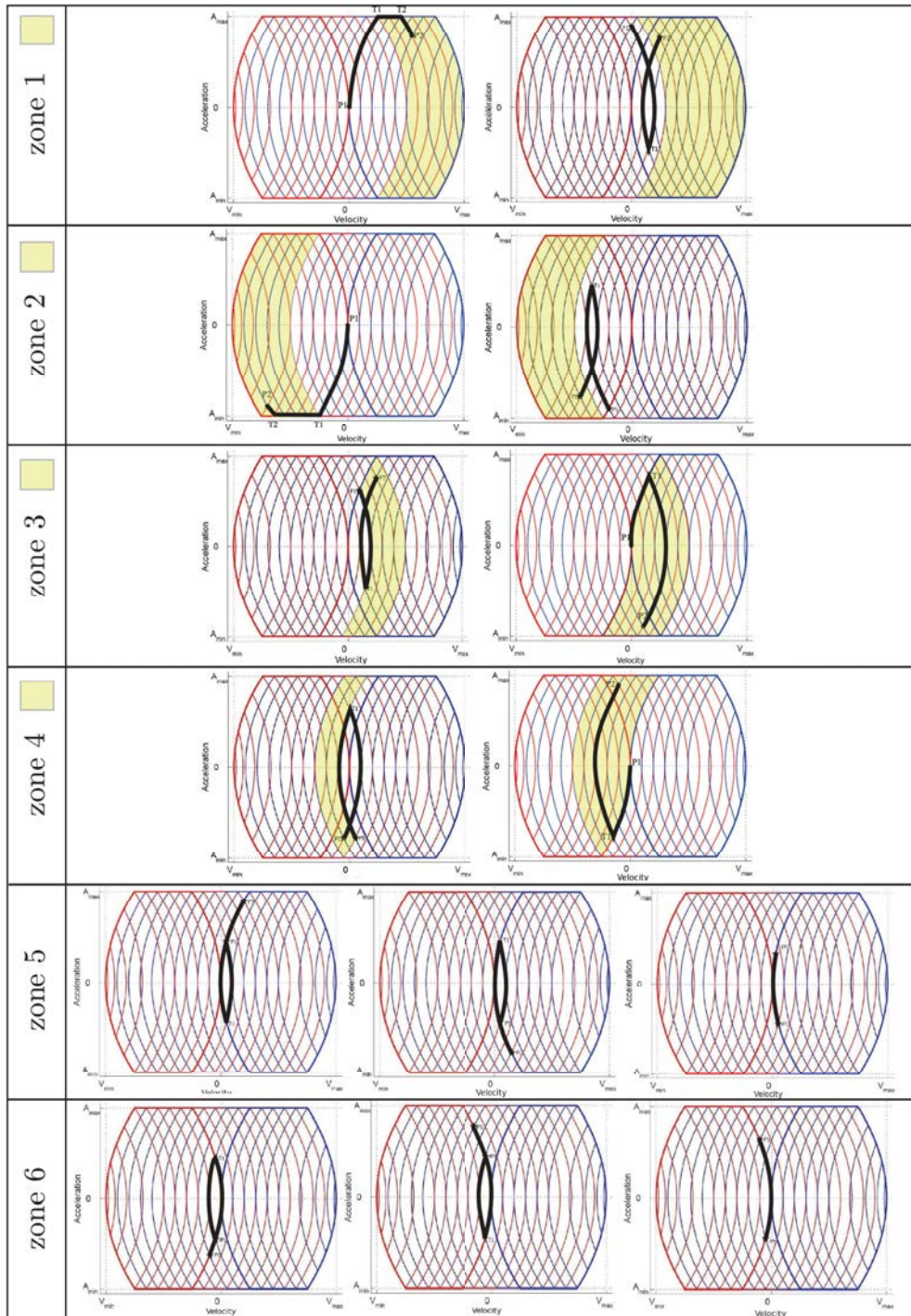


FIGURE 3.12 – Les différents types de mouvements critiques en fonction des zones

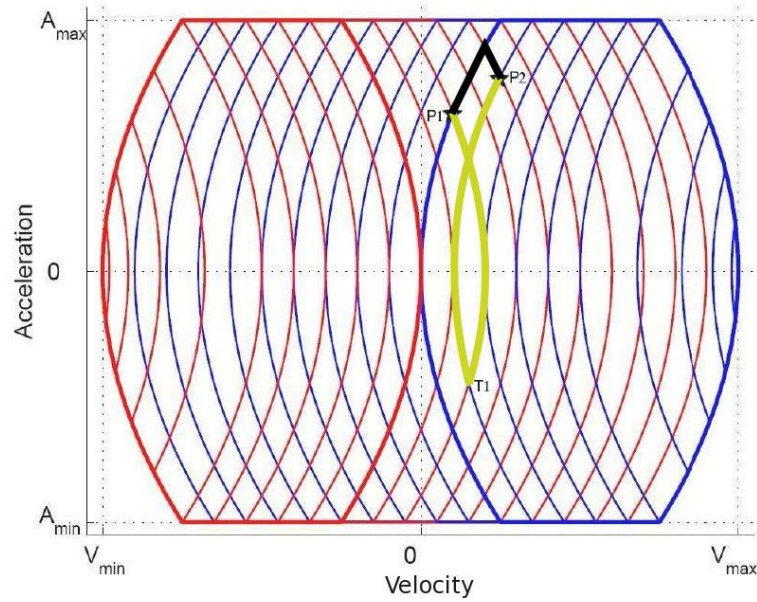


FIGURE 3.13 – Illustration de l'existence de mouvement en temps minimal particulier (noir)

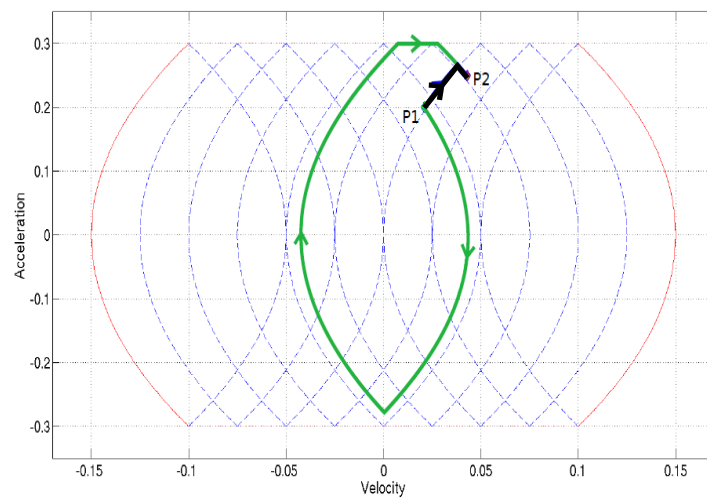


FIGURE 3.14 – Illustration de l'existence de deux solutions pour un mouvement avec des conditions initiales et finales quelconques

3.3.4 L'algorithme général de planification en temps minimal

Ce paragraphe présente l'algorithme de génération de trajectoire avec des conditions initiales et finales quelconques (en accélération, vitesse et position) et avec contraintes cinématiques (eq. 3.12 et 3.13).

Méthodes de calcul :

L'algorithme que nous proposons est conçu pour être utilisé en ligne. Il est basé sur des calculs de seuils de distance pour déterminer la suite de segments composant le mouvement. Deux seuils sont facilement identifiables. Le premier DSV_{max} est la distance parcouru de telle sorte que le mouvement comporte un segment à vitesse saturée. Le second DSA_{max} , correspond à la distance quand les accélérations A_{max} et A_{min} sont atteintes.

Utilisation de la propriété de symétrie du plan accélération-vitesse :

Pour réduire de moitié l'algorithme, nous avons utilisé la propriété de symétrie du plan accélération-vitesse introduite précédemment (Section 3.1.6). Ainsi une planification pour un chemin de type 2 pourra être calculé comme celle d'un chemin du premier type en inversant judicieusement certaines variables :

- les conditions initiales et finales, la distance critique sont multipliées par -1 (la distance étant relative et $X_I = 0$) ;
- inversion de la zone pour le calcul de la distance critique suivant le cas :
 - zone 1 \Rightarrow zone 2
 - zone 2 \Rightarrow zone 1
 - zone 3 \Rightarrow zone 4
 - zone 4 \Rightarrow zone 3
 - zone 5 \Rightarrow zone 6
 - zone 6 \Rightarrow zone 5
- re-calculation des couples de référence : $(V_{smm}, A_{max}), (V_{smp}, A_{max})...$

L'algorithme :

Ainsi, à partir de la définition de la distance critique dc (Section 3.3.3.1), nous proposons l'algorithme 2 de génération de mouvement. Cet algorithme se décompose en trois niveaux de sous-algorithmes. L'algorithme principal (alg. 2) fait appel, si besoin, à l'algorithme de génération de mouvement de type 1 (alg. AT1 3).

L'algorithme pour les trajectoires de type 1 se décompose ensuite en trois autres algorithmes : AT1-1 (alg. 4), AT1-2 (alg. 5) et enfin AT1-3 (alg. 6). L'algorithme AT1-3 correspond au cas où ni la vitesse maximale, ni les accélérations maximales ne sont atteintes. La décomposition de ce dernier est détaillée dans l'annexe C. Les cas les plus complexes correspondent aux mouvements composés de trois segments à jerk saturé ; c'est à dire au calcul de l'intersection de trois paraboles (équation de degré six). Ces cas sont résolus numériquement par la méthode de Raphson-Newton. Nous proposons au lecteur de se reporter à l'ouvrage de Press pour les détails de cette méthode [Press 07].

Algorithme 2 : Génération de trajectoire mono-dimensionnelle de type V

Calcul des vitesses particulière (V_{smp}, \dots)

Détermination des sept zones

Calcul de la distance critique dc

Calcul de la longueur du mouvement $D = X_F - X_I$

début

si $D > dc$ **alors**

 Le mouvement est de type 1

 Calcul du mouvement de type 1 (alg. 3 page 50)

 Vérification des temps des sept segments

sinon si $D < dc$ **alors**

 Le mouvement est de type 2

 Inversion des conditions initiales et finales (la propriété de symétrie)

 Calcul des nouvelles vitesses particulières (V_{smp}, \dots)

 Calcul du mouvement de type 1 (alg. 3 page 50)

 Détermination du mouvement de type 2 à partir du mouvement de type 1

 Vérification des temps des sept segments

sinon si $D == dc$ **alors**

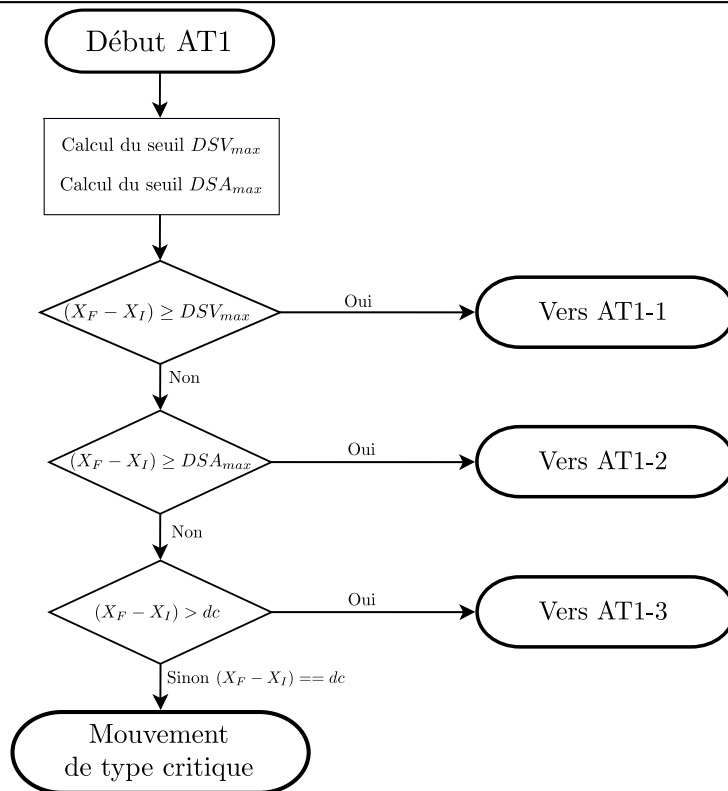
 Le mouvement est le mouvement critique

 Ce mouvement a déjà été calculé lors du calcul de la distance critique

fin

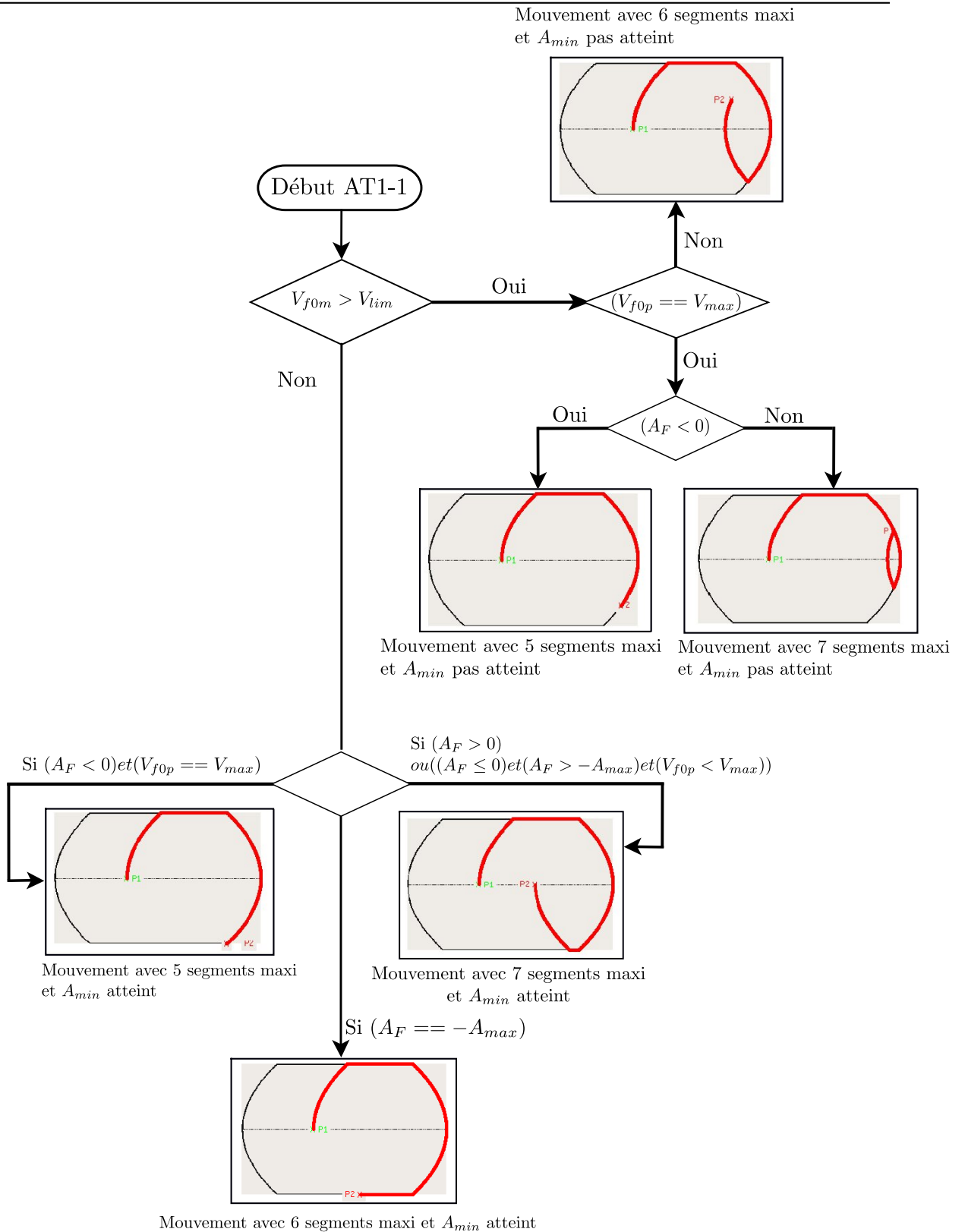
fin

Algorithme 3 : AT1 : Algorithme général pour les mouvements de type 1

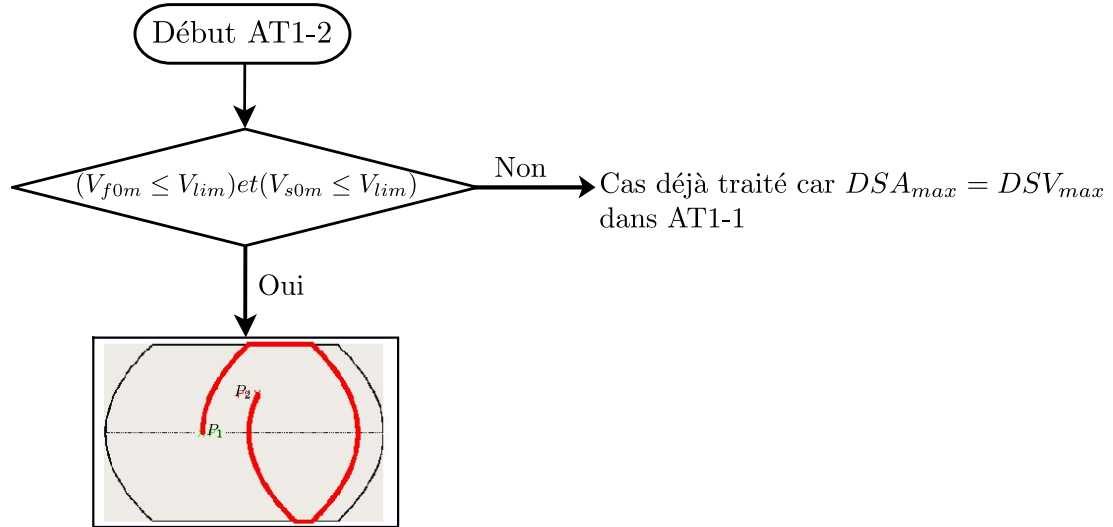


Les algorithmes AT1-1, AT1-2 et AT1-3 sont respectivement décrits aux pages 51, 52 et 52.

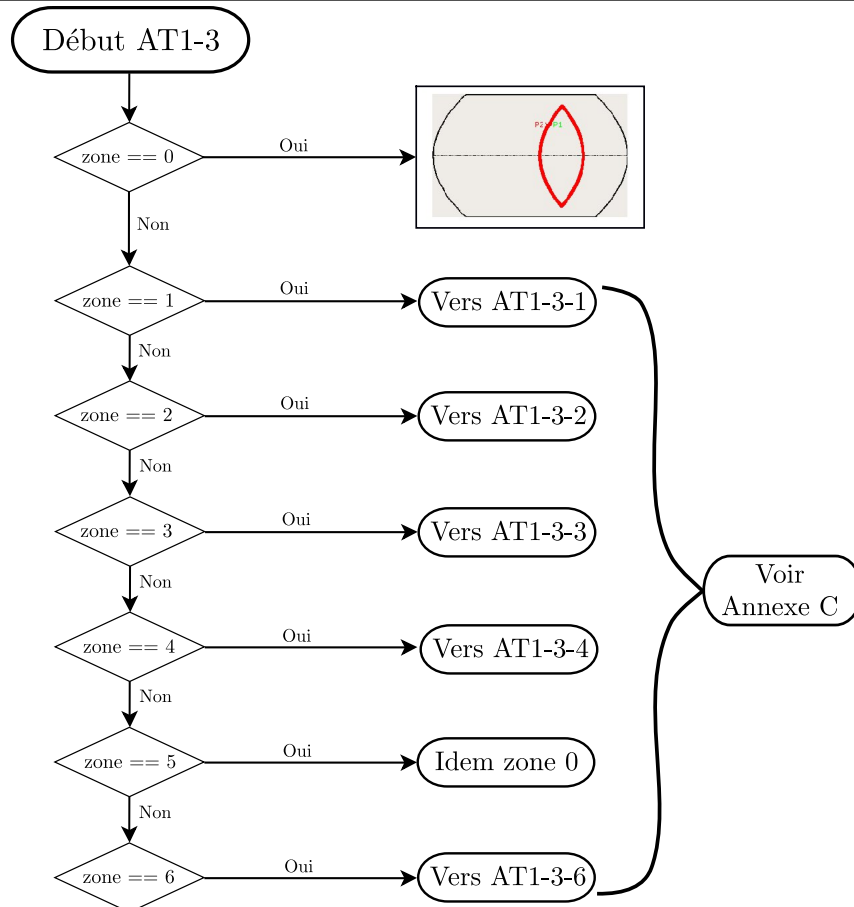
Algorithme 4 : AT1-1 : Algorithme pour les mouvements atteignant V_{max}



Algorithme 5 : AT1-2 : Algorithme pour les mouvements atteignant A_{max} et $-A_{max}$



Algorithme 6 : AT1-3 : Algorithme pour les mouvements n'atteignant pas A_{max} et $-A_{max}$



3.3.5 Exemples d'applications

Nous présentons deux applications de la méthode de génération de mouvement de type V en temps minimal. La première est le calcul d'un mouvement entre deux états (position, vitesse, accélération). La deuxième est une extension de cet algorithme qui permet de générer un mouvement suivant une consigne cinématique (vitesse,accélération).

3.3.5.1 Exemple de génération d'un mouvement de type V

Considérons les conditions initiales et finales suivantes :

$$\begin{aligned} A(t_I) &= -0.25m.s^{-2} & A(t_f) &= 0.19m.s^{-2} \\ V(t_I) &= -0.07m.s^{-1} & V(t_f) &= -0.01m.s^{-1} \\ X(t_I) &= 0 & X(t_f) &= -0.048m \end{aligned}$$

Nous imposons les contraintes cinématiques :

$$\begin{aligned} J_{max} &= 0.9m.s^{-3} \\ A_{max} &= 0.3m.s^{-2} \\ V_{max} &= 0.15m.s^{-1} \end{aligned}$$

Les conditions initiales et finales en vitesse et accélération permettent de déterminer que le point FC se situe dans la zone 1 par rapport au point IC. La figure 3.15 représente le mouvement critique. La distance critique est $dc = -0.060473m$. La distance à parcourir $D = X_F = -0.048m$ est supérieure à dc donc le mouvement est de type 1 (départ à jerk maximal). Le seuil DSV_{max} qui est la distance nécessaire de telle sorte que le mouvement possède un segment à vitesse constante non nul est $DSV_{max} = 0.0519m$. Le seuil DSA_{max} , qui représente un mouvement qui atteint A_{min} et A_{max} vaut $-0.0334 m$. On est donc dans la cas de l'algorithme AT1 (alg. 3) :

$$dc < D < DSA_{max}$$

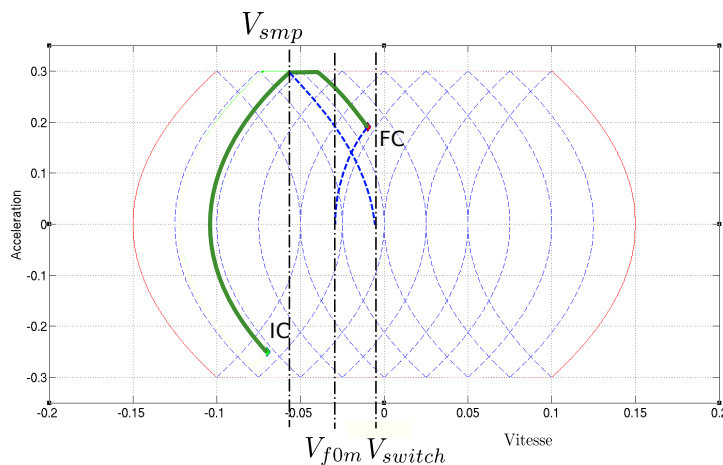


FIGURE 3.15 – Mouvement critique de l'exemple

La solution est donc traitée par l'algorithme AT1-3-1 (alg. C.1) fourni en annexe C (car la zone est de type 1). Dans AT1-3-1, nous avons introduit une vitesse particulière nommée V_{switch} . Cette vitesse correspond à la vitesse à accélération nulle de la parabole à jerk négatif passant par V_{smp} . Dans notre cas, la vitesse $V_{switch} = -0.0047$ m/s est supérieure à V_{f0m} . Ceci implique que la parabole à jerk négatif passant par V_{switch} et la parabole à jerk positif passant par les conditions finales se croisent. Les mouvements des trois seuils DSA_{max31} , DSA_{max31} et DSA_{max31} peuvent donc être calculés. Les résultats du calcul des trois seuils nécessaires pour déterminer la composition du mouvement sont :

$$\begin{aligned} DSA_{max31} &= -0.0731m \\ DSA_{max311} &= -0.0615m \\ DSA_{max33} &= -0.0696m \end{aligned}$$

Ces trois seuils permettent de déterminer la composition du mouvement. La distance D à parcourir est supérieure à DSA_{max31} , on obtient donc un mouvement composé de 5 mouvements élémentaires. La figure 3.16 présente le résultat. Le système s'écrit sous la forme d'une équation à une inconnu de degré 4. Nous avons choisi comme inconnu la durée du mouvement élémentaire de la parabole à jerk positif passant par les conditions finales.

Le mouvement obtenu est composé de cinq segments dont les durées sont respectivement :

$$\begin{aligned} T_{jpa} &= 0.611111s \\ T_{aca} &= 0.182863s \\ T_{jna} &= 0.333333s \\ T_{vc} &= 0s \\ T_{jnb} &= 0.2985s \\ T_{acb} &= 0s \\ T_{jpb} &= 0.509612s \end{aligned}$$

3.3.5.2 Génération du mouvement de suivi d'une consigne (vitesse, accélération)

La méthode de calcul de mouvement présentée ci-après permet de générer un mouvement en ligne à partir d'une consigne. Un exemple pratique de cette méthode est présenté dans la section 6.2. Le mouvement est calculé à chaque actualisation de la consigne en respectant les contraintes cinématiques (J_{max} , A_{max} et V_{max}) à partir de l'état courant du système.

Dans la section 3.3.3.1, nous avons vu que le mouvement en temps minimal sous contraintes cinématiques est le mouvement critique correspondant à la distance critique dc . Ainsi, pour que le système mono-dimensionnel atteigne la consigne en un temps minimal, le mouvement critique est calculé à chaque itération. La figure 3.17 illustre l'évolution de la vitesse générée (courbe bleue) à partir de la consigne en vitesse (courbe noire pointillé). Dans l'exemple, la consigne d'accélération est nulle. La méthode agit comme un filtre sur l'accélération. Nous appellerons f_{smooth} cette fonction de lissage.

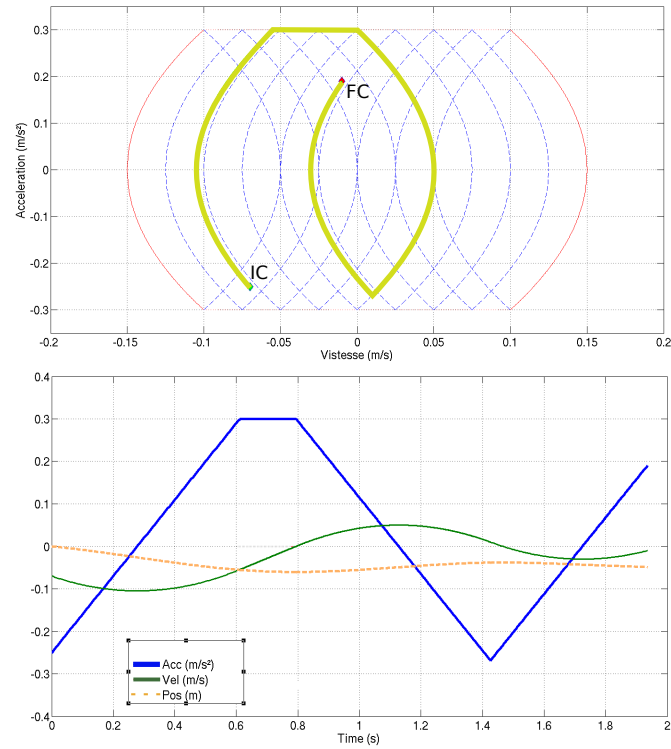


FIGURE 3.16 – Exemple de mouvement de type V

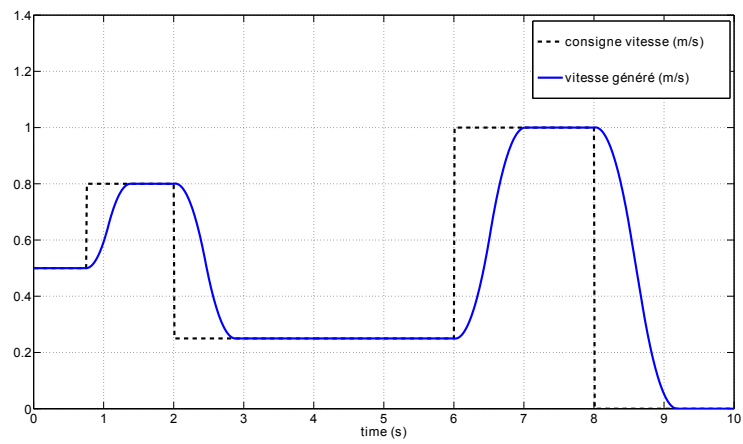


FIGURE 3.17 – Vitesse générée pour une consigne en vitesse-accelération

3.3.6 Implémentation et résultats

Nous avons implémenté l’algorithme de génération de mouvement de type V (alg. 2) en langage C. La figure 3.18 représente l’interface graphique utilisateur optionnelle.

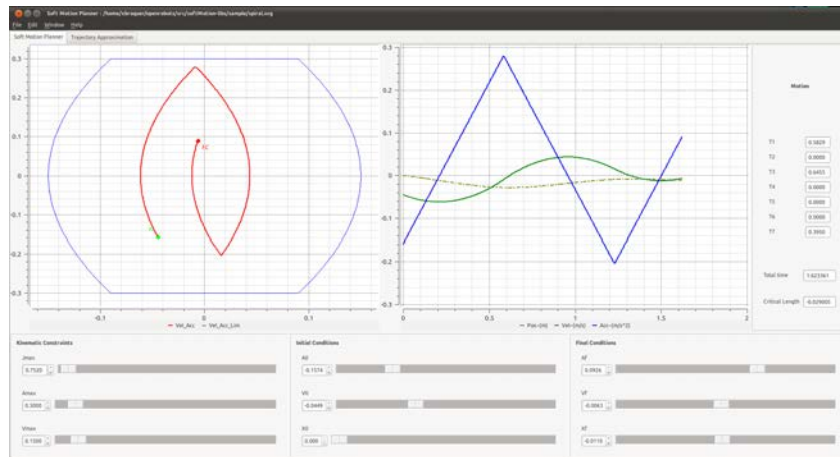


FIGURE 3.18 – Interface graphique utilisateur du générateur de trajectoire mono-dimensionnelle

Pour valider l’implémentation, nous avons échantillonné aléatoirement des conditions initiales et finales respectant les contraintes cinématiques imposées (J_{max} , A_{max} et V_{max}). Sur une moyenne d’un milliard de tests, le temps de calcul moyen d’un mouvement mono-dimensionnel est de **0.0139 ms** (machine utilisée : Intel Core i5 2.67 GHz). Les temps de calculs les plus longs correspondant aux cas les plus complexes sont de l’ordre de la milliseconde. Ce résultats montrent la validité de l’approche pour les applications temps réels. Dans le chapitre 6, nous présenterons une expérimentation basée sur l’utilisation en-ligne de cet algorithme sur un robot manipulateur.

Dans la première partie de ce chapitre, nous avons proposé une méthode de calcul de mouvement en temps minimal mono-dimensionnel. Dans la suite, nous allons nous intéresser aux mouvements en temps imposé pour terminer par les mouvements multidimensionnels.

3.4 Les trajectoires mono-dimensionnelles en temps imposé

3.4.1 Cas du mouvement point à point

Pour effectuer un mouvement point à point en un temps imposé T_{imp} ($T_{imp} >$ temps minimal), il existe une infinité de solutions. Par exemple, parcourir la moitié de la distance, s’arrêter le temps nécessaire et enfin effectuer la distance restante.

Une méthode est décrite dans l’ouvrage de Biagiotti pour un mouvement atteignant V_{max} et A_{max} [Biagiotti 08] (section 3.4.5). Cette méthode consiste à calculer les contraintes J_{max} , A_{max} et V_{max} permettant de réaliser le mouvement en une durée imposée T_{imp} . Dans le paragraphe 3.2, nous avons vu que le calcul du mouvement en temps minimal revient à calculer trois temps caractéristiques T_{vc} , T_a et T_j . La durée totale du mouvement étant $T = T_{vc} + 2(T_a + 2T_j)$. Pour un mouvement avec des contraintes symétriques et une distance D à parcourir, la durée T du mouvement est donnée par (voir annexe D) :

$$T = \frac{D}{V_{max}} + \frac{V_{max}}{A_{max}} + \frac{A_{max}}{J_{max}} \quad (3.23)$$

Pour une durée imposée $T_{imp} > T$, on obtient un paramètre δ tel que :

$$\delta = \frac{T}{T_{imp}} \quad (3.24)$$

L'équation 3.23 permet d'obtenir les nouvelles contraintes :

$$\begin{aligned} V_{max}' &= \delta V_{max} \\ A_{max}' &= \delta^2 A_{max} \end{aligned} \quad (3.25)$$

$$J_{max}' = \delta^3 J_{max} \quad (3.26)$$

Cette méthode permet d'obtenir facilement une solution en temps imposé lorsque les durées des sept segments sont non nulles.

Une autre solution consiste à imposer les trois temps T_{vc} , T_a et T_j (chacun de ces trois temps étant supérieur aux temps minimum). A partir de ces trois temps et des équations générales du mouvement point à point, nous pouvons recalculer le jerk J_{imp} à imposer (équations 3.27, 3.28 et 3.29) :

Si le segment à vitesse constante est non null ($T_{vc} > 0$) :

$$J_{imp} = \frac{X_F - X_I}{T_j(T_a^2 + 3T_a T_j + T_{vc} T_a + 2T_j^2 + T_{vc} T_j)} \quad (3.27)$$

si le segment à accélération constante est non null $T_a > 0$ et $T_{vc} = 0$:

$$J_{imp} = \frac{X_F - X_I}{T_j(T_a^2 + 3T_a T_j + 2T_j^2)} \quad (3.28)$$

et si $T_a = 0$ et $T_{vc} = 0$:

$$J_{imp} = \frac{X_F - X_I}{2T_j^3} \quad (3.29)$$

3.4.2 Cas du mouvement de type V

Les conditions cinématiques initiales et finales étant non nulles, la planification du mouvement en temps imposé est plus complexe que dans le cas point à point, le mouvement n'est plus symétrique. Pour ajuster la durée du mouvement, nous ne pouvons plus modifier uniquement les contraintes J_{max} , A_{max} et V_{max} . En effet, considérons un mouvement en temps minimal seulement constitué d'un mouvement élémentaire à vitesse constante saturée. La diminution de la vitesse maximale n'est plus possible car le mouvement deviendrait invalide (e.g. les conditions initiales et finales seraient en dehors de l'espace accélération-vitesse valide).

La solution que nous avons choisi consiste alors à calculer un mouvement avec une évolution à vitesse constante non saturée. Nous proposons un algorithme itératif pour déterminer la valeur de la

vitesse du segment à vitesse constante. Pour effectuer un mouvement en temps infini, la valeur de cette vitesse tendra vers zéro. Donc la valeur de la vitesse à déterminer sera comprise entre la vitesse nulle et la vitesse maximale atteinte par le mouvement en temps minimal calculé par l’algorithme de la section précédente (section 3.3.4).

La méthode proposée permet de prendre en compte les accélérations et vitesses initiales et finales non nulles. La première étape de l’algorithme consiste à calculer le mouvement en temps optimal. La durée de ce mouvement est notée T_{opt} et la vitesse maximale atteinte, V_m ($V_m = \max(V(t))$). Si la vitesse V_m est positive alors la valeur de la vitesse du segment à vitesse constante (notée V_c) appartiendra à l’intervalle $[0, V_m]$, pour tout temps imposé $T_{imp} \geq T_{opt}$. Inversement, si V_m est négatif, alors V_c appartiendra à l’intervalle $[V_m, 0]$. La seconde phase de la méthode consiste à faire varier V_c dans l’intervalle précédemment déterminé. Pour chaque valeur de V_c , le mouvement joignant les conditions cinématiques initiales et finales et passant par V_c est calculé. Ce dernier mouvement est composé de six segments (il ne comprend pas de segment à vitesse constante). La somme des durées de ces six segments est notée T_{6seg} et permet de déterminer la durée du mouvement du segment à vitesse constante ($T_{vc} = T_{imp} - T_{6seg}$). La solution est trouvée si la distance parcourue par le mouvement composé des six segments et du segment à vitesse constante est égale à la distance initiale. La figure 3.20 illustre ce type de mouvement, le mouvement en temps optimal étant tracé sur la figure 3.19.

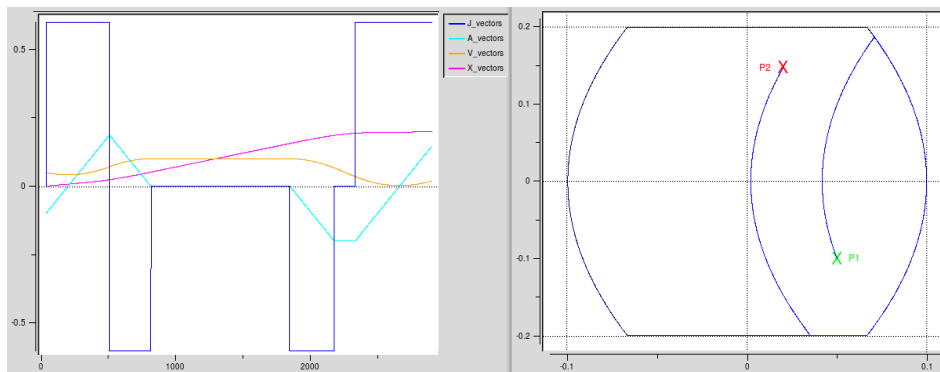
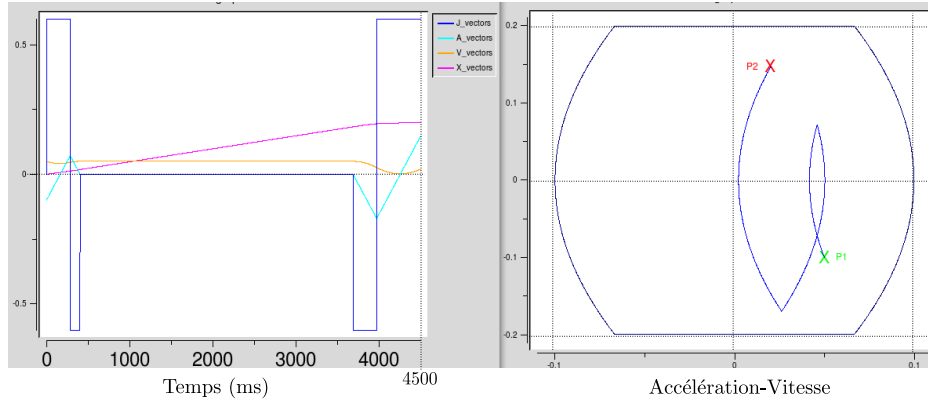


FIGURE 3.19 – Mouvement en temps optimal de durée T_{opt} ($T_{opt} = 2.903s$)

Remarque : La somme des six segments T_{6seg} peut être supérieure à T_{imp} , dans ce cas il n’y aura pas de solution. Ce problème apparait pour les petites distances proche de la distance critique. Une solution serait d’augmenter la contrainte J_{max} pour réduire le temps de parcours du mouvement à six segments. Nous ne retenons pas cette solution car augmenter le jerk maximal, introduirait des variations d’accélération trop brutales pour l’interaction avec l’Homme. Dans la section suivante nous détaillons la méthode utilisée pour synchroniser un mouvement multidimensionnel.

3.5 Les trajectoires multidimensionnelles en temps minimal

Dans cette section, nous présentons la planification en temps minimal pour les trajectoires multidimensionnelles.

FIGURE 3.20 – Mouvement en temps imposé de durée $T_{imp} = 4.5s$

3.5.1 Cas du mouvement point à point

Dans un espace de dimension n , le mouvement est obtenu par la combinaison linéaire du mouvement des n axes. Soit la direction de déplacement définie par le vecteur $v = \sum_{i=1}^n \alpha_i v_i$ où $\sum_{i=1}^n \alpha_i^2 = 1$. Les vecteurs v_i sont les n vecteurs directeurs unitaires de l'espace considéré. La synchronisation des n mouvements point à point mono-dimensionnels permet d'obtenir une ligne droite entre le point initial et le point final si les instants de commutations entre les segments sont identiques pour tous les axes. Les différents jerks de chaque axe sont reliés linéairement par $J_{iM} = \alpha_i J_{max}$. J_{max} correspond au jerk maximum sur la droite.

Le mouvement en temps minimal est alors directement obtenu par :

$$J_{max} = \min_{1 \leq i \leq n} \frac{1}{\alpha_i} J_{iM} \quad (3.30)$$

$$A_{max} = \min_{1 \leq i \leq n} \frac{1}{\alpha_i} A_{iM} \quad (3.31)$$

$$V_{max} = \min_{1 \leq i \leq n} \frac{1}{\alpha_i} V_{iM} \quad (3.32)$$

Pour chaque segment, l'une des fonctions vitesse, accélération ou jerk de l'un des axes est saturée. Les autres se trouvent à l'intérieur de leur domaine de validité.

Une autre méthode pour calculer un mouvement point à point multidimensionnel en temps minimal consiste à calculer, dans un premier temps, les mouvements optimaux pour chaque dimension avec les contraintes cinématiques correspondantes. Les durées des segments du mouvement de chaque dimension sont ensuite ajustées au mouvement qui a la plus grande durée par la deuxième méthode décrite au paragraphe 3.4.1.

3.5.2 Cas du mouvement de type V

Contrairement au mouvement point à point, dans le cas du mouvement de type V, les segments des n mouvements en temps minimal ne sont pas forcément synchronisés. La solution que nous avons choisi se décompose en trois étapes. La première étape consiste à calculer le mouvement en temps minimal

pour les n dimensions à partir de l’algorithme présenté dans la section précédente 3.3.4. Les n temps des mouvements sont stockés dans un vecteur de dimension n que nous notons T_{opt} . On a donc pour l’axe j :

$$T_{opt}[j] = \sum_{i=1}^7 jT_i \quad (3.33)$$

La deuxième étape consiste à calculer pour chaque axe, le mouvement qui passe par l’état à l’arrêt (point A du plan accélération-vitesse). Ce mouvement sera constitué de 11 segments au plus (3 segments pour rejoindre le point d’arrêt, le segment à l’arrêt et les 7 segments pour effectuer le déplacement restant). De même que pour les mouvements optimaux, les durées de ces mouvements sont stockées dans un vecteur T_{stop}

$$T_{stop}[j] = \sum_{i=1}^{11} jT_i \quad (3.34)$$

La troisième étape consiste à déterminer la durée de synchronisation T_{imp} . Notons que cette durée est différente de celle introduite dans la section précédente (Section 3.4). Cette durée n’est pas forcément égale à la plus grande des durées des mouvements des axes. En effet, comme nous l’avons vu dans la section sur la planification en temps imposé 3.4.2, il existe des intervalles de temps où il n’y a pas de solution de planification en temps imposé dans la famille de solutions que nous avons choisi.

Ainsi, pour déterminer la durée T_{imp} , nous déterminons pour chaque axe j les intervalles solutions de la planification en temps imposé pour l’intervalle de temps $[T_{opt}[j], T_{stop}[j]]$ (figure 3.21). La durée T_{imp} sera égale à la durée minimale valide pour l’ensemble des axes.

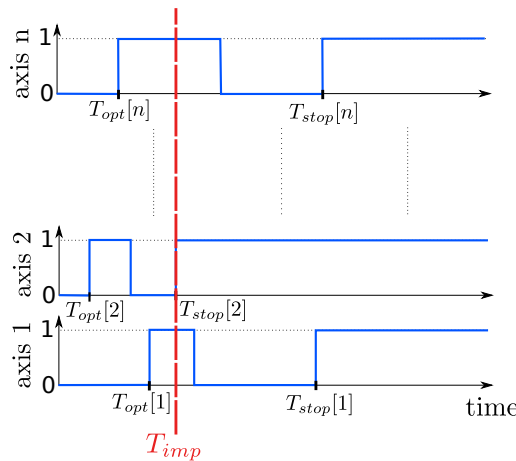


FIGURE 3.21 – Détermination du temps de synchronisation pour un mouvement multidimensionnel

3.6 Conclusion

Dans ce chapitre, nous avons détaillé notre contribution concernant la planification de trajectoires sous contraintes cinématiques. Nous avons développé un planificateur de trajectoire permettant de générer des trajectoires avec le jerk, l’accélération et la vitesse bornés. Ce générateur permet de calculer des mouvements multidimensionnels avec des conditions initiales et finales cinématiques non nulles. Pour cela,

nous avons utilisé un modèle de trajectoire composé de suites de segments de courbes polynomiales cubiques. Dans le chapitre suivant, nous utilisons ce modèle pour approximer des trajectoires.

4

Approximation de trajectoire

Dans le chapitre précédent, nous avons présenté une approche originale pour la génération de trajectoire sous contraintes cinématiques par interpolation entre deux états (position, vitesse et accélération). La trajectoire ainsi générée est composée de suites de segments de courbes polynomiales cubiques. Dans ce chapitre, nous allons étudier l'approximation de trajectoire par ce même type de trajectoire. Le but de cette approximation est d'obtenir une représentation unique pour l'ensemble des trajectoires que le contrôleur du robot devra exécuter. Une partie des résultats présentés ci-après a été présentée à la conférence ICRA (*IEEE International Conference on Robotics and Automation*) [Broquère 10].

4.1 Définition du problème

Considérons une trajectoire quelconque TR_{in} définie par un chemin $\mathcal{P} = \mathcal{P}(u)$ et par une loi de mouvement $u = u(t)$. Cette trajectoire peut être définie par une grande variété de courbes (Beziers, NURBS, sinusoides ...). Dans le domaine industriel, ces courbes sont généralement représentées par des composées de primitives telles que les cercles et les droites. L'inconvénient de la représentation par des cercles et des droites provient des discontinuités dans le profil d'accélération. En effet, lors du passage d'une droite à un cercle, ou inversement, une discontinuité en accélération apparaît à cause de l'accélération centripète introduite par le cercle. Cette discontinuité peut être évitée en imposant des vitesses nulles aux points liant les différentes primitives. Cependant, cette solution n'est généralement pas retenue car elle augmente les temps de parcours.

Outre le fait d'obtenir une représentation uniforme des trajectoires pouvant être exécutées par un unique contrôleur, l'approximation de trajectoire permet de lisser ces discontinuités.

Nous proposons une méthode générale d'approximation pour les trajectoires définies par un chemin

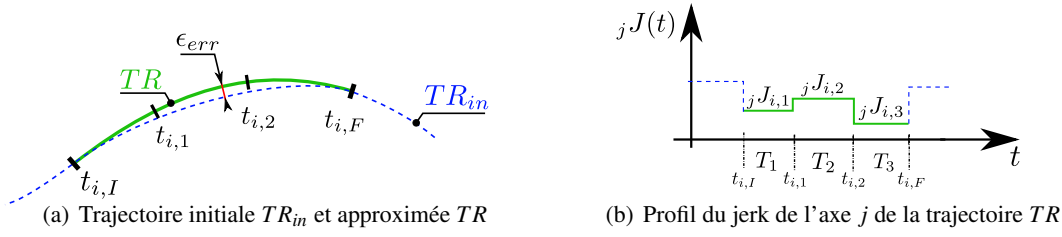


FIGURE 4.1 – Approximation en temps imposé de trajectoire par trois segments de courbes cubiques

\mathcal{P} et une loi de mouvement $u(t)$. L'approximation est effectuée à partir d'un ensemble d'instantés ou calculés de la trajectoire initiale TR_{in} . Entre deux instantés consécutifs, un mouvement en temps imposé est calculé pour approximer la trajectoire. La section suivante présente la méthode proposée pour calculer ce mouvement en temps imposé.

4.2 Mouvement en temps imposé sans contrainte cinématique

Un mouvement mono-dimensionnel entre deux états (position, vitesse, accélération) introduit six contraintes :

- les trois conditions initiales ;
- les trois conditions finales.

Une cubique a seulement quatre coefficients et ne peut donc pas satisfaire les six contraintes. L'utilisation de deux cubiques pour réaliser ce mouvement est aussi impossible (8 coefficients pour 9 contraintes : les conditions initiales et finales et les 3 contraintes de continuité entre les deux cubiques). Trois segments de cubiques (figure 4.1(a) et 4.1(b)) introduisent 12 coefficients qui permettent de résoudre le problème :

- Six contraintes pour les conditions initiales (A_I, V_I, X_I) et finales (A_F, V_F, X_F) ;
- Trois contraintes de continuité à l'instant $t_{i,1}$:

$$TR(t_{i,1}^-) = TR(t_{i,1}^+), \quad \left. \frac{dTR(t)}{dt} \right|_{t_{i,1}^-} = \left. \frac{dTR(t)}{dt} \right|_{t_{i,1}^+}, \quad \left. \frac{d^2TR(t)}{dt^2} \right|_{t_{i,1}^-} = \left. \frac{d^2TR(t)}{dt^2} \right|_{t_{i,1}^+}$$

- Trois contraintes de continuité à l'instant $t_{i,2}$:

$$TR(t_{i,2}^-) = TR(t_{i,2}^+), \quad \left. \frac{dTR(t)}{dt} \right|_{t_{i,2}^-} = \left. \frac{dTR(t)}{dt} \right|_{t_{i,2}^+}, \quad \left. \frac{d^2TR(t)}{dt^2} \right|_{t_{i,2}^-} = \left. \frac{d^2TR(t)}{dt^2} \right|_{t_{i,2}^+}$$

Ce qui donne donc 12 contraintes. La durée imposée du mouvement est $T_{imp} = t_{i,F} - t_{i,I}$.

Le système à résoudre est ensuite obtenu à partir des équations 3.9, 3.10 et 3.11 représentant les évolutions de l'accélération, de la vitesse et de la position en fonction de l'indice du segment et du temps (page 34).

Nous obtenons ainsi le système suivant :

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} \quad (4.1)$$

où les coefficients des matrices A et B sont :

$$\begin{aligned}
 A_{11} &= T_1 \\
 A_{12} &= T_2 \\
 A_{13} &= T_3 \\
 A_{21} &= \frac{T_1^2}{2} + T_1 T_2 + T_1 T_3 \\
 A_{22} &= \frac{T_2^2}{2} + T_2 T_3 \\
 A_{23} &= \frac{T_3^2}{2} \\
 A_{31} &= \frac{T_1^3}{6} + \frac{T_1^2}{2}(T_2 + T_3) + \frac{T_1}{2}(T_2^2 + T_3^2) + T_1 T_2 T_3 \\
 A_{32} &= \frac{T_2^3}{6} + \frac{T_2^2}{2} T_3 + \frac{T_2}{2} T_3^2 \\
 A_{33} &= \frac{T_3^3}{6}
 \end{aligned}$$

$$\begin{aligned}
 B_1 &= A_F - A_I \\
 B_2 &= V_F - V_I - A_I T_{imp} \\
 B_3 &= X_F - X_I - V_I T_{imp} - A_I \frac{T_{imp}^2}{2}
 \end{aligned}$$

Si nous fixons arbitrairement la durée de chaque segment au tiers de la durée totale imposée T_{imp} :

$$T_1 = T_2 = T_3 = \frac{T_{imp}}{3} \quad (4.2)$$

Alors la matrice A devient :

$$A = \begin{bmatrix} \frac{1}{3} T_{imp} & \frac{1}{3} T_{imp} & \frac{1}{3} T_{imp} \\ \frac{5}{18} T_{imp}^2 & \frac{3}{18} T_{imp}^2 & \frac{1}{18} T_{imp}^2 \\ \frac{19}{162} T_{imp}^3 & \frac{7}{162} T_{imp}^3 & \frac{1}{162} T_{imp}^3 \end{bmatrix} \quad (4.3)$$

Les trois jerks solutions se calculent alors directement à partir de l'inverse de la matrice A :

$$\begin{bmatrix} J_1 \\ J_2 \\ J_3 \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} \quad (4.4)$$

avec

$$A^{-1} = \frac{1}{T_{imp}} \begin{bmatrix} 1 & -9 & 27 \\ -7/2 & 27 & -54 \\ 11/2 & -18 & 27 \end{bmatrix} \quad (4.5)$$

Dans la suite du manuscrit, nous nommerons cette méthode de calcul en temps imposé *3-segments*.

Remarque : Le rapport des durées de chaque segment par rapport à la durée imposée peut être optimisé en fonction d'une contrainte comme la minimisation des trois jerk. Cependant, nous avons choisi de fixer les durées de chaque segment au tiers de T_{imp} pour obtenir un calcul rapide.

4.3 Algorithmes d'approximation proposés

Nous proposons deux méthodes pour approximer une trajectoire. Ces méthodes sont basées sur le même principe. La trajectoire initiale est d'abord découpée en plusieurs sous trajectoires. Ensuite, pour chaque sous trajectoire caractérisée par une durée et des conditions initiale et finale (position, vitesse et accélération), nous appliquons la méthode de planification en temps imposée *3-segments*. Pour la première méthode, nommée *approximation par division a priori*, le nombre de sous trajectoires est fixé a priori. Pour la seconde méthode, le nombre de sous trajectoires est déduit de l'erreur maximale admissible $\mathcal{E}_{errPosMaxDes}$ donnée par l'utilisateur. Pour calculer l'erreur, la trajectoire initiale est discrétisée à ΔT_s .

4.4 Calculs des erreurs d'approximation

Les erreurs d'approximations sont calculées à partir des trajectoires TR_{in} et TR échantillonnées à ΔT_s . Trois erreurs sont calculées (où n est le nombre d'axes) :

– l'erreur en position de trajectoire :

$$\forall t \in [t_I, t_F], \mathcal{E}_{errPos}(t) = \sqrt{\sum_{j=1}^n ({}_jTR(t) - {}_jTR_{in}(t))^2} \quad (4.6)$$

$$\mathcal{E}_{errPosMax} = \max(\forall t \in [t_I, t_F] | \mathcal{E}_{errPos}(t)) \quad (4.7)$$

– l'erreur en vitesse de trajectoire :

$$\forall t \in [t_I, t_F], \mathcal{E}_{errVel}(t) = \sqrt{\sum_{j=1}^n \left(\frac{d_j TR(t)}{dt} - \frac{d_j TR_{in}(t)}{dt} \right)^2} \quad (4.8)$$

$$\mathcal{E}_{errVelMax} = \max(\forall t \in [t_I, t_F] | \mathcal{E}_{errVel}(t)) \quad (4.9)$$

– la distance de Hausdorff entre les trajectoires discrétisées TR et TR_{in} :
cette distance représente la distance maximale entre les deux chemins.

$$d_{Haus}(TR_{in}, TR) = \max(f_d(TR_{in}, TR), f_d(TR, TR_{in})) \quad (4.10)$$

où f_d est la distance de Hausdorff relative :

$$f_d(TR_{in}, TR) = \max(\forall p \in TR_{in} \mid \min(\forall q \in TR \mid d(p, q))) \quad (4.11)$$

et

$$f_d(TR, TR_{in}) = \max(\forall q \in TR \mid \min(\forall p \in TR_{in} \mid d(p, q))) \quad (4.12)$$

La distance d utilisée est la distance euclidienne.

Pour effectuer les calculs d'erreur, sauf précision contraire, nous utilisons une discrétisation dix fois plus fine que l'incrément de temps utilisé. Nous supposons dans cette partie que la période d'exécution est fixée à 10 ms. La discrétisation est donc effectuée à $\Delta T_s = 1ms$. Notons que le choix de cette période influence la précision du calcul de l'erreur, notamment dans le cas de la distance de Hausdorff.

La distance de Hausdorff est calculée à titre indicatif ; les algorithmes proposés ne nécessitent pas son calcul.

4.5 Approximation de trajectoire multidimensionnelle avec un nombre imposé de triplets de segments

Pour approximer une trajectoire, nous présentons d'abord une méthode *par division a priori* (alg. 7), qui consiste à diviser la trajectoire en N_{st} sous-trajectoires. Chaque sous trajectoire TR^i est définie par un instant initial $t_{i,I}$ et un instant final $t_{i,F}$. Ces instants sont identiques pour les n dimensions de TR_{in} .

Algorithme 7 : Méthode *par division a priori*

Données d'entrées : TR_{in} discrétisée à ΔT_s et N_{st}

début

- ├ Détermination des conditions initiales IC et finales FC et de la durée des N_{st} sous-trajectoires
- ├ Application de la méthode '3-segments' pour chaque sous-trajectoire
- ├ Calcul des erreurs d'approximation de la trajectoire

fin

La trajectoire TR obtenue est composée de $3 * N_{st}$ segments de cubiques pour chaque dimension.

Remarque : Le calcul de l'approximation s'effectuant avec trois segments et sur des trajectoires discrétisées à ΔT_s , le nombre d'échantillons de la trajectoire TR_{in} doit être un multiple de trois. Si ce n'est pas le cas, nous ajoutons un ou deux échantillons à la fin de la trajectoire avec des conditions égales à celles de la position finale de TR_{in} . La durée totale de la trajectoire TR aura ainsi au maximum une durée de $2\Delta T_s$ plus longue que TR_{in} .

4.5.1 Application au cas simple du segment de droite

Considérons le segment de droite défini par les points P_1 et P_2 tels que :

$$P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad P_2 = \begin{bmatrix} 0.15 \\ 0.1 \\ 0 \end{bmatrix}$$

La loi de mouvement imposée (figure 4.2) est un mouvement point à point à sept segments avec les paramètres :

$$J_{max} = 0.12m/s^3 \quad A_{max} = 0.04m/s^2 \quad V_{max} = 0.02m/s$$

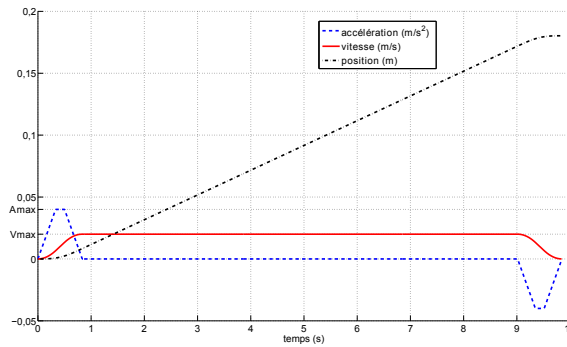


FIGURE 4.2 – Loi de mouvement imposée

Pour cet exemple, le nombre de sous trajectoires est fixé à sept (figure 4.3). Pour les calculs, la trajectoire est échantillonnée à $\Delta T_y = 1$ ms.

Ainsi dans le cas du déplacement en ligne droite, les erreurs en vitesse de la loi de mouvement et en position de la trajectoire sont tracées sur la figure 4.4. L'erreur maximale de position sur la trajectoire est 0.395 mm et l'erreur de Hausdorff 0.01060 mm.

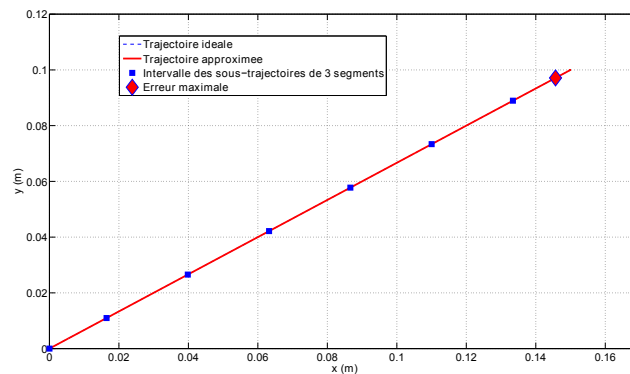


FIGURE 4.3 – Approximation d'un segment de droite par sept sous trajectoires 3-segments

4.5.2 Application au cas du cercle

Considérons un cercle de rayon 0.10 m, centré sur le point $P_1 = [0 \ 0 \ 0]^T$. La loi de mouvement imposée a les mêmes paramètres que celle de la droite du cas précédent (figure 4.2). Le nombre de sous trajectoire est également fixé à sept (figure 4.5).

Les erreurs en vitesse de la loi de mouvement et en position de la trajectoire sont tracées sur la figure 4.6. L'erreur maximale de position sur la trajectoire est de 13.815 mm, l'erreur de vitesse 15.45 mm/s et l'erreur de Hausdorff 2.99215 mm.

Comme l'on pouvait s'y attendre, l'erreur de trajectoire est très grande. Nous proposons dans la section suivante un algorithme d'approximation en fonction de l'erreur maximale désirée.

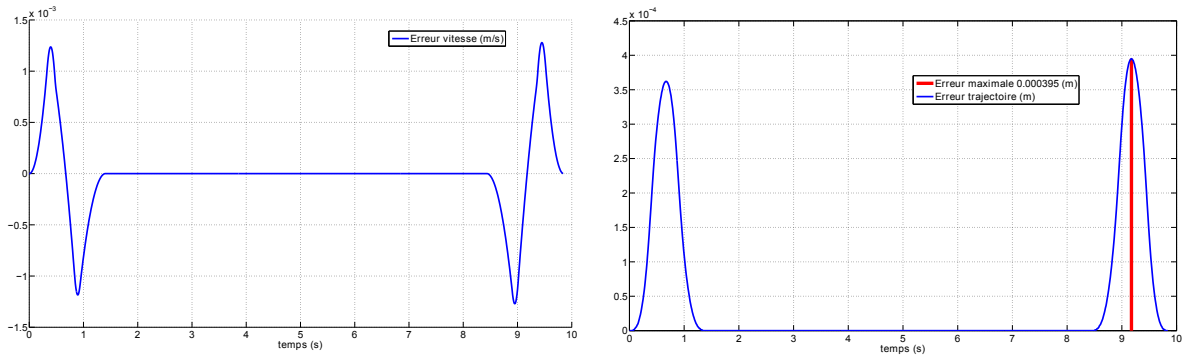


FIGURE 4.4 – Erreur en vitesse de la loi de mouvement (gauche) et Erreur en position de la trajectoire (droite) du segment de droite approximé

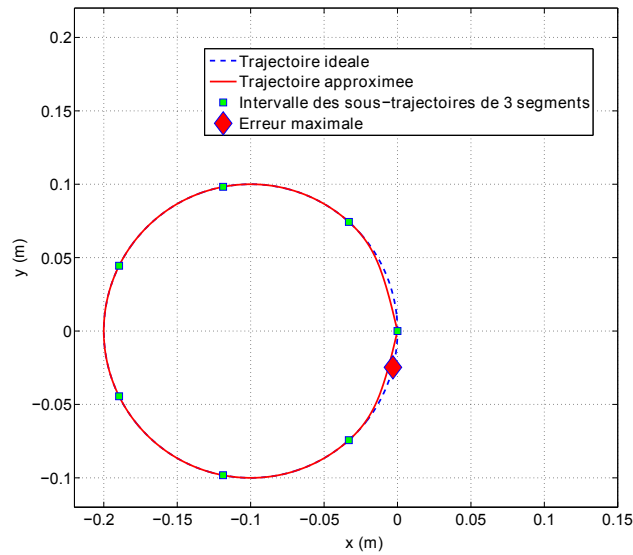


FIGURE 4.5 – Approximation d'un cercle par sept sous trajectoires 3-segments

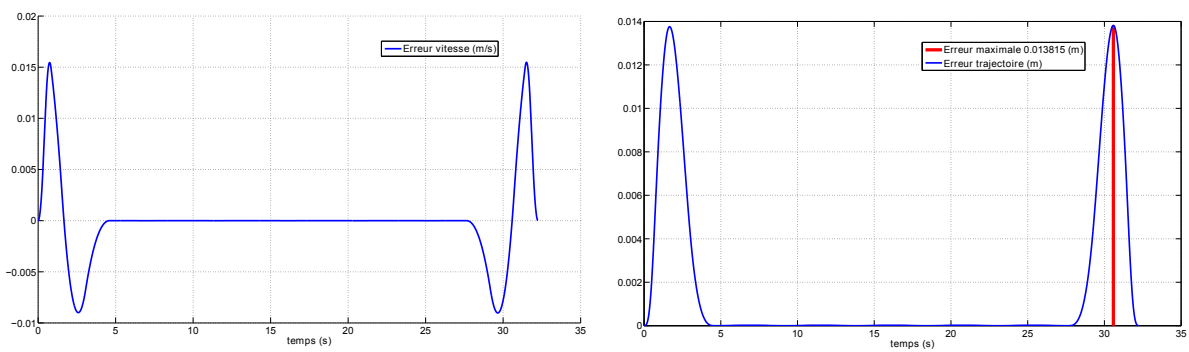


FIGURE 4.6 – Erreur en vitesse de la loi de mouvement (gauche) et Erreur en position de la trajectoire (droite) du cercle approximé

4.6 Approximation de trajectoire en fonction d'une erreur maximale imposée

La seconde méthode proposée permet d'approximer une trajectoire en contraignant l'erreur maximale de trajectoire $\epsilon_{errPosMax}$. La division en sous trajectoires de la trajectoire initiale TR_{in} est effectuée par dichotomie. TR_{in} est initialement divisée en deux sous trajectoires. La méthode *3-segments* est ensuite appliquée à chaque sous trajectoire (alg. 8). Si une sous trajectoire ne satisfait pas la contrainte d'erreur, elle est alors re-divisée. L'algorithme s'arrête lorsque les erreurs des sous trajectoires sont inférieures à $\epsilon_{errPosMaxDes}$ où que la longueur temporelle des sous trajectoires est égale à trois échantillons (ΔT_s).

Remarque : Lors de la division par dichotomie, la trajectoire n'est pas exactement divisée en deux sous trajectoires de durée égale. Ces durées sont ajustées à des durées multiples de trois échantillons.

Algorithme 8 : Méthode *par division automatique de la trajectoire en fonction de l'erreur*

$\epsilon_{errPosMaxDes}$

Données d'entrées : TR_{in} discrétisée à ΔT_s

début

si nombreEchantillons(TR_{in}) modulo 3 \neq 0 **alors**

 | Ajout d'échantillons à la fin de la trajectoire TR_{in}

fin

 Division de TR_{in} en 2 sous trajectoires $subTR_1$ et $subTR_2$

$listeSubTR_{in}.ajout(subTR_1)$

$listeSubTR_{in}.ajout(subTR_2)$

répéter

pour Chaque élément $subTR_i$ dans la liste des sous trajectoires $listeSubTR_{in}$ **faire**

 | Détermination des conditions initiales IC et finales FC et de la durée de chaque sous-trajectoire

 | Application de la méthode '*3-segments*'

 | Calcul de l'erreur $\epsilon_{errPosMax}$ de la sous trajectoire

si $\epsilon_{errPosMax} < \epsilon_{errPosMaxDes}$ **alors**

 | La sous trajectoire est valide

 | $listeSubTR.ajout(subTR_i)$

 | $listeSubTR_{in}.supr(subTR_i)$

sinon

 | Division de $subTR_i$ en deux sous trajectoires $subTR_a$ et $subTR_b$

 | $listeSubTR_{in}.ajout(subTR_a)$

 | $listeSubTR_{in}.ajout(subTR_b)$

 | $listeSubTR_{in}.supr(subTR_i)$

fin

fin

jusqu'à $listeSubTR_{in}.vide() = vraie;$

fin

4.6.1 Application au cas du segment de droite

Nous reprenons l'exemple de la Section 4.5.1 ainsi que les paramètres de la loi de mouvement imposée. La trajectoire est échantillonnée à 1 ms et l'erreur maximale admissible est fixée à 10^{-6} m. Pour satisfaire ces contraintes, le segment de droite a été divisé en 16 intervalles (figure 4.7). On constate que la durée des intervalles est faible au début et à la fin de la trajectoire pour respecter l'erreur imposée. Ces intervalles correspondent à la zone où les variations de vitesse et d'accélération sont les plus importantes.

Les erreurs en vitesse de la loi de mouvement et en position de la trajectoire sont tracées sur la figure 4.8, elle est de 1.9×10^{-5} m/s soit 0.095 %. L'erreur maximale de position sur la trajectoire est de 0.59×10^{-6} m et l'erreur de Hausdorff est 0.98×10^{-6} m.

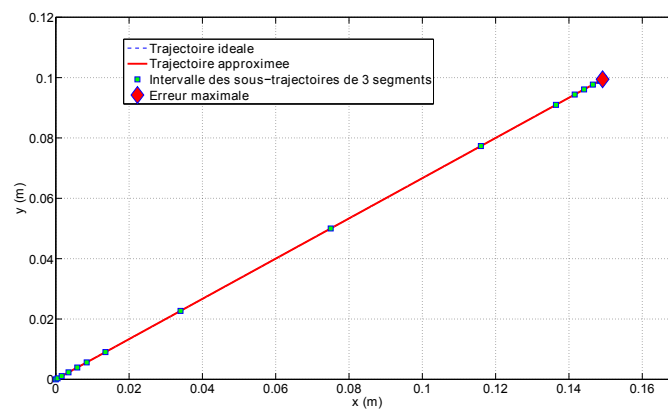


FIGURE 4.7 – Approximation d'un segment de droite en fonction de l'erreur maximale de trajectoire imposée

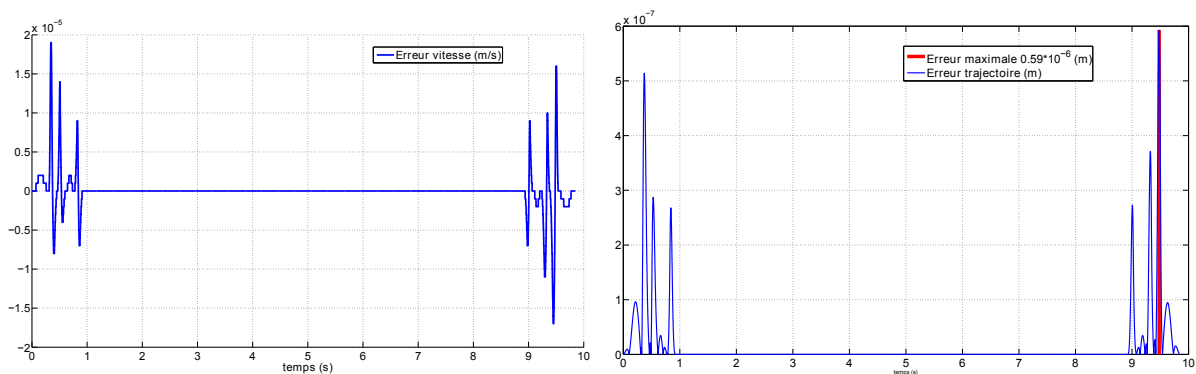


FIGURE 4.8 – Erreur en vitesse de la loi de mouvement (gauche) et Erreur en position de la trajectoire (droite) de la droite approximée par la méthode de division automatique

4.6.2 Application au cas du cercle

Le cercle et la loi de mouvement ont les mêmes paramètres que ceux de la section 4.5.2. La trajectoire est échantillonnée à 0.1 ms et l'erreur maximale admissible est fixée à 10^{-6} m.

Le cercle a été divisé en 24 intervalles (figure 4.9). Comme dans le cas précédent, la durée des intervalles est faible au début et à la fin de la trajectoire. La faible erreur imposée permet de faire apparaître les oscillations autour du cercle sur la partie où la vitesse de la loi de mouvement est constante. Ces oscillations n'apparaissent pas dans la division a priori car les erreurs de début et de fin étaient prépondérantes.

Les erreurs en vitesse de la loi de mouvement et en position de la trajectoire sont tracées dans la figure 4.10. L'erreur maximale de position sur la trajectoire est 0.752×10^{-6} mm et celle en vitesse de 1.26×10^{-5} m/s. L'erreur de Hausdorff est de 0.98×10^{-6} m.

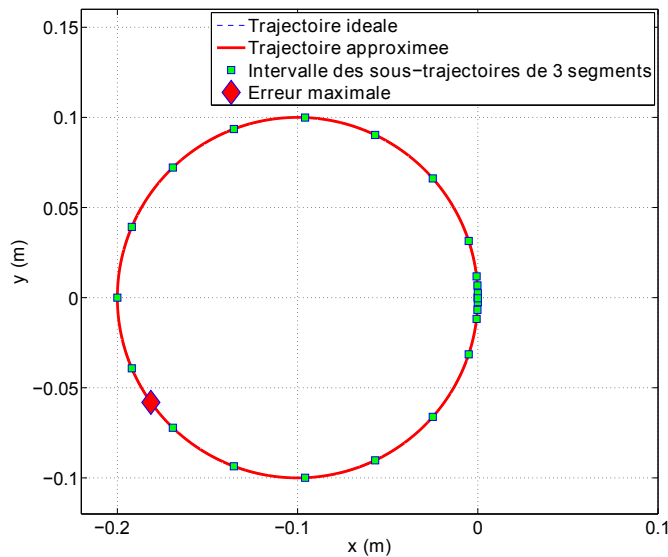


FIGURE 4.9 – Approximation d'un cercle en fonction de l'erreur maximale de trajectoire imposée

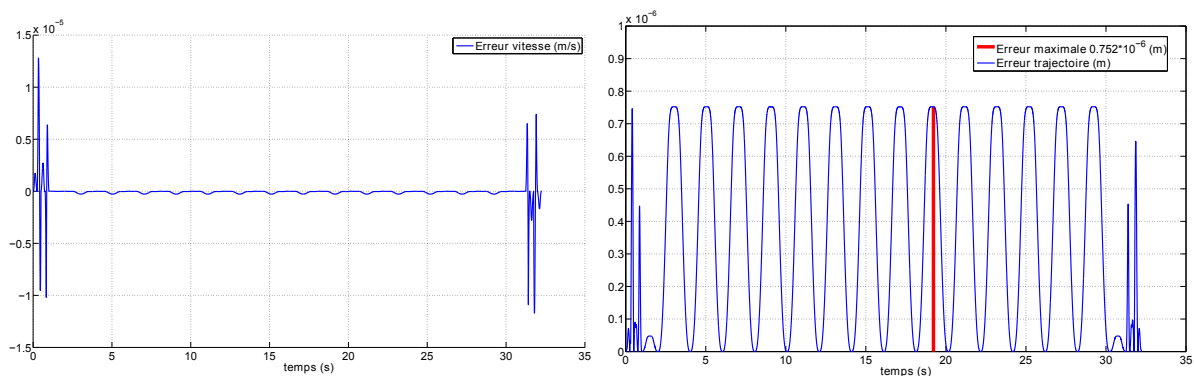


FIGURE 4.10 – Erreur en vitesse de la loi de mouvement (gauche) et Erreur en position de la trajectoire (droite) du cercle approximée par la méthode de division automatique

4.7 Cas des trajectoires non C^2

Dans cette section, nous appliquons la méthode d'approximation a priori à une trajectoire qui n'est pas continue en accélération. Pour cela, nous reprenons le cas présenté dans la Section 9.6.4 de l'ouvrage de Biagiotti [Biagiotti 08]. La courbe initiale d'origine $P_0 = [0 \ 0 \ 0]^T$ est composée de droites et d'arcs de cercle 4.12 :

- droite de P_0 à $P_1 = [1 \ 0 \ 0]^T$;
- arc de cercle de rayon 1.0 m de P_1 à $P_2 = [2 \ 1 \ 0]^T$;
- arc de cercle de rayon 1.0 m de P_2 à $P_3 = [2 \ 1 \ 2]^T$;
- droite de P_3 à $P_4 = [2 \ 0 \ 2]^T$;

La loi de mouvement est un mouvement à sept segments (figure 4.14) avec les paramètres :

$$V_{max} = 2\text{m/s} \quad A_{max} = 40\text{m/s}^2 \quad J_{max} = 200\text{m/s}^3$$

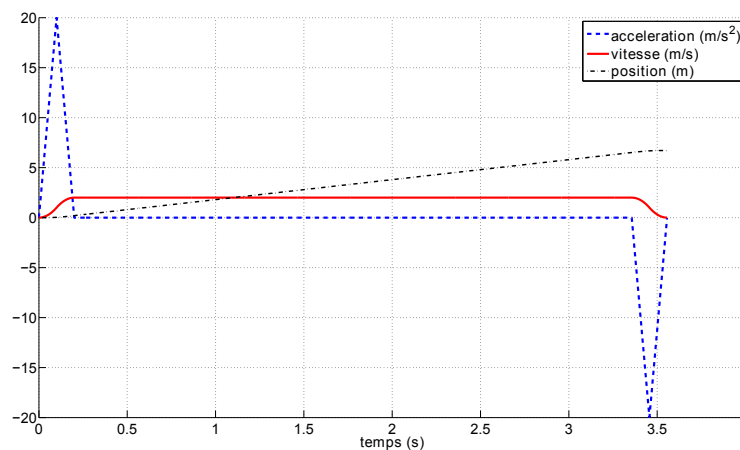


FIGURE 4.11 – Loi de mouvement imposée

4.7.1 Approximation par un nombre donné d'intervalles

Dans [Biagiotti 08], cette trajectoire est approximée par une B-spline (Basic-spline) de degré trois. Deux solutions d'approximations sont présentées. La première est calculée avec une B-spline définie par 10 points de contrôle et la seconde avec 42 points. Dans les deux cas la distance de Hausdorff donnée dans l'ouvrage est comparée à notre approche (tableau 4.1).

Nous appliquons l'algorithme d'approximation *par division a priori* avec d'abord 10 puis 41 sous trajectoires. Seul le cas avec 42 points est représenté sur les figures ci-après. Les erreurs en vitesse de la loi de mouvement et en position de la trajectoire sont tracées sur la figure 4.13.

La figure 4.14 représente la loi de mouvement (valeur absolue) de la trajectoire approximée obtenue. On remarque l'effet de l'accélération centripète due au cercle entre les instants $t=0.6\text{s}$ et $t=3\text{s}$. L'erreur de trajectoire la plus grande se situe à l'instant $t=1.4\text{s}$ où la plus grande discontinuité en accélération apparaît.

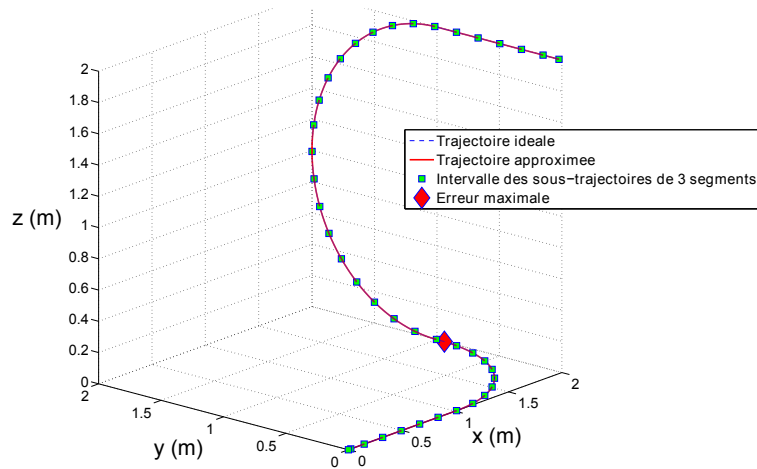


FIGURE 4.12 – Approximation d’une trajectoire composée de segments de droite et d’arcs de cercle

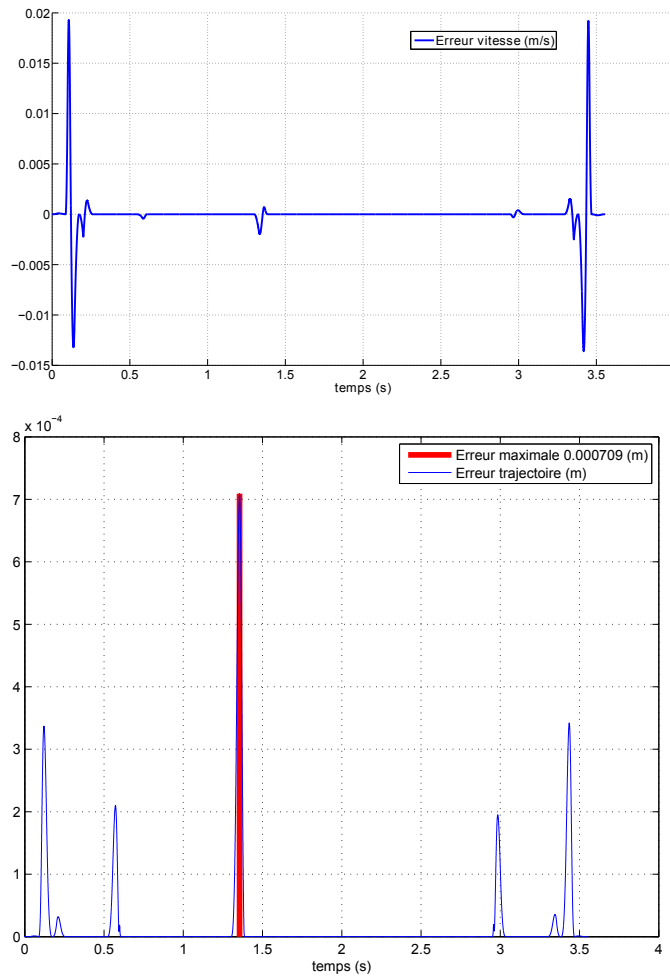


FIGURE 4.13 – Erreur en vitesse de la loi de mouvement (haut) et Erreur en position de la trajectoire (bas) de la droite approximée par la méthode de division a priori avec 42 points

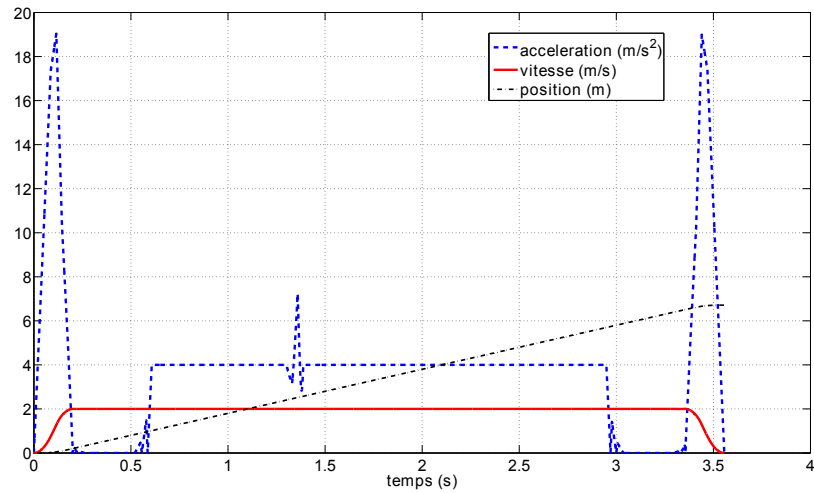


FIGURE 4.14 – Loi de mouvement de la trajectoire approximée

TABLE 4.1 – Comparaison entre l'approximation par la méthode *3-segment* et par B-spline cubique

Nb points	Type d'erreur	Méthode <i>3-segment</i>	Méthode B-spline cubique
10 points	Hausdorff (mm)	1.5	19.4
	Vitesse de la loi de mouvement (mm/s)	333	-
	Position sur la trajectoire (mm)	27.2	-
42 points	Hausdorff (mm)	0.7	3.2
	Vitesse de la loi de mouvement (mm/s)	19.2	-
	Position sur la trajectoire (mm)	0.7	-

4.7.2 Approximation par la méthode limitant l'erreur

Nous avons appliqué l'algorithme par division automatique (alg. 8) en fixant une erreur maximale $\epsilon_{errPosMaxDes} = 1mm$. Seulement 18 triplets sont nécessaires pour satisfaire cette erreur. L'erreur maximale de trajectoire obtenue étant de 0.87 mm et l'erreur de Hausdorff de 0.71 mm. La vitesse maximale de la loi de mouvement approximée est de 2.016 m/s soit 1% d'erreur, l'accélération maximale 17.90 m/s^2 et le jerk maximal 315 m/s^3 . Ce dépassement de jerk ($> 200m/s^3$) apparaît aux phases de démarrage et d'arrêt de la trajectoire. Le temps de calcul de cette approximation est de 0.13s (machine utilisée : Intel Core i5 2.67 GHz).

4.8 Conclusion

Nous avons présenté une méthode d'approximation de trajectoires multidimensionnelles par des trajectoires capables de limiter un paramètre d'erreur donné. Elle est basée sur un calcul simple de mouvement en temps imposé composé d'un triplet de segments de cubiques (méthode que nous avons nommé *3-segments*). L'erreur la plus pertinente pour nos problèmes est l'erreur de trajectoire (erreur de position à un instant donné entre les trajectoires). Nous avons montré que cette méthode permet de lisser les discontinuités d'accélération pour les trajectoires composées de segments de droites et de cercles. La comparaison avec la méthode d'approximation par des B-splines montre la validité de l'approche. La trajectoire approximée est composée de suites de segments de courbes cubiques comme pour le générateur de trajectoire présenté dans le chapitre précédent ; ce qui permet d'utiliser le même contrôleur de trajectoire pour les exécuter.

5

Planification et exécution de mouvement pour l'interaction Homme-robot

Dans le chapitre 2, nous avons vu que la présence de l'Homme dans l'espace de travail du robot introduit des contraintes, non seulement au niveau de la planification du chemin, mais aussi au niveau de la génération et de l'exécution de la trajectoire.

Nous nous appuyons sur les travaux de Akin Sisbot [Sisbot 07] et de Jim Mainprice [Mainprice 11] pour définir des chemins qui servent de point de départ aux calculs des trajectoires. La planification du chemin ne fait pas partie des travaux menés dans le cadre de cette thèse. De même, nous ne traitons pas les mouvements non holonomes des plateformes mobiles.

Dans ce chapitre nous présentons d'abord le planificateur de chemin utilisé pour calculer le chemin d'entrée de notre générateur de trajectoire. Dans une deuxième partie nous détaillons notre approche de génération de trajectoire à partir du chemin précédemment obtenu. Dans ces deux premières parties, la planification est effectuée dans l'espace des configurations du robot.

La troisième section présente la planification dans l'espace cartésien. Nous y proposons notamment une méthode originale pour approximer une trajectoire définie dans l'espace cartésien par une trajectoire définie dans l'espace articulaire.

Nous introduisons enfin une méthode de contrôle de trajectoire permettant de prendre en compte des contraintes de l'interaction Homme-robot pendant l'exécution de la trajectoire.

Une partie des travaux présentés dans ce chapitre a été publiée dans la revue *Journal of Social Robotics*, Volume 2, Number 3, 2010.

5.1 Planification du chemin

La première étape de la planification du mouvement est le calcul du chemin \mathcal{P} . Dans notre cas, nous utilisons le planificateur proposé initialement par Sisbot [Sisbot 07]. Ce planificateur de mouvement prend explicitement en compte les contraintes d'interaction Homme-robot (HRI) pour synthétiser des mouvements de navigation (mouvement de la plateforme) et de manipulation (plateforme du robot fixe). Les contraintes HRI sont par exemple la distance Homme-robot ou encore le champ de vue de l'Homme. Ce planificateur est basé sur des études de cas d'interaction Homme-robot [Koay 07], ainsi que sur les théories existantes sur le partage de l'espace inter-humains [Hall 63]. Les contraintes d'interaction Homme-robot sont représentées par des fonctions de coût basées respectivement sur la posture de l'Homme, son champ de vision et son accessibilité. Ces différents coûts sont représentés par des cartes de coûts définies dans l'espace de travail du robot. Pour résoudre une tâche de manipulation comme donner un objet à l'Homme, le chemin de l'objet est d'abord planifié en utilisant des techniques de grilles définies dans l'espace de travail. Le chemin du robot est ensuite planifié à partir de la cinématique inverse du robot. Cependant, le chemin de l'objet étant fixé par la première étape de la méthode, ce planificateur n'est pas performant dans des espaces encombrés. Nous utilisons l'extension proposée par Mainprice [Mainprice 11]. Cette extension consiste à étendre les capacités du planificateur par l'utilisation d'algorithmes de planification à base d'échantillonnage aléatoire (Section 2.2.3) pour calculer des mouvements prenant en compte l'Homme dans des environnements encombrés.

Ainsi, lorsque le robot partage un espace avec l'Homme, le planificateur de chemin prend en compte les contraintes HRI en utilisant la méthode probabiliste T-RRT [Jaillet 08]. Lorsque la tâche à réaliser ne nécessite pas de prendre en compte l'Homme, la méthode probabiliste RRT [Kuffner 00] est utilisée.

5.1.1 Prise en compte des contraintes géométriques

Les contraintes pour l'interaction Homme-robot proposées dans [Sisbot 07] sont modélisées par des fonctions de coûts. Ces coûts sont importants lorsque la configuration du robot n'est pas confortable pour l'Homme ou qu'elle est dangereuse. Nous retenons trois contraintes :

- **Contrainte de sécurité** (figure 5.1(a)) : cette contrainte permet d'assurer la sécurité de l'interaction en contrôlant la distance entre le robot et l'humain. L'Homme est modélisé par un volume approximatif délimitant le corps (sans tenir compte de la géométrie du bras). Cette contrainte de sécurité maintient le robot loin de la tête et du corps, pour limiter tout risque de collision entre l'humain et le robot. La distance prise en compte est la distance minimale entre le robot (l'ensemble des parties du robot sont prises en compte) et le modèle simplifié de l'Homme (cylindre + sphère). Lorsque cette distance est faible, le coût est élevé ; et inversement lorsque cette distance augmente le coût diminuera jusqu'à un seuil minimal au delà duquel il devient nul ;
- **Contrainte de visibilité** (figure 5.1(b)) : cette contrainte a pour but de limiter l'effet de surprise que peut ressentir l'Homme pendant que le robot bouge dans l'espace de travail. Un Homme se sent moins surpris si le robot reste visible et l'interaction est plus sûre et plus confortable. Ainsi, chaque point de l'espace de travail a un coût proportionnel à l'angle entre le regard de l'Homme et sa position dans l'espace cartésien ;

- **Contrainte de "confort du bras"** : cette troisième contrainte est prise en compte pour des tâches d'échange d'objet entre le robot et l'Homme. Elle permet de déterminer un point de transfert de l'objet dans l'espace de travail. Cette contrainte est également prise en compte lors de la planification de chemin de la tâche d'échange pour faciliter l'échange de l'objet à tout moment pendant le déplacement. Pour cela, le robot doit raisonner sur les capacités d'accessibilité de l'Homme et sa cinématique. Le volume supposé atteignable de l'Homme peut être pré-calculé en utilisant la cinématique inverse généralisée. Pour chaque point à l'intérieur du volume atteignable de l'Homme, la configuration du torse déterminée reste aussi près que possible d'une posture de repos donnée. La détection de collision avec l'environnement est ensuite utilisée pour valider ces postures. A chaque posture valide est attribué un coût de confort grâce au modèle de prédiction de posture humaine introduit dans [Marler 05].

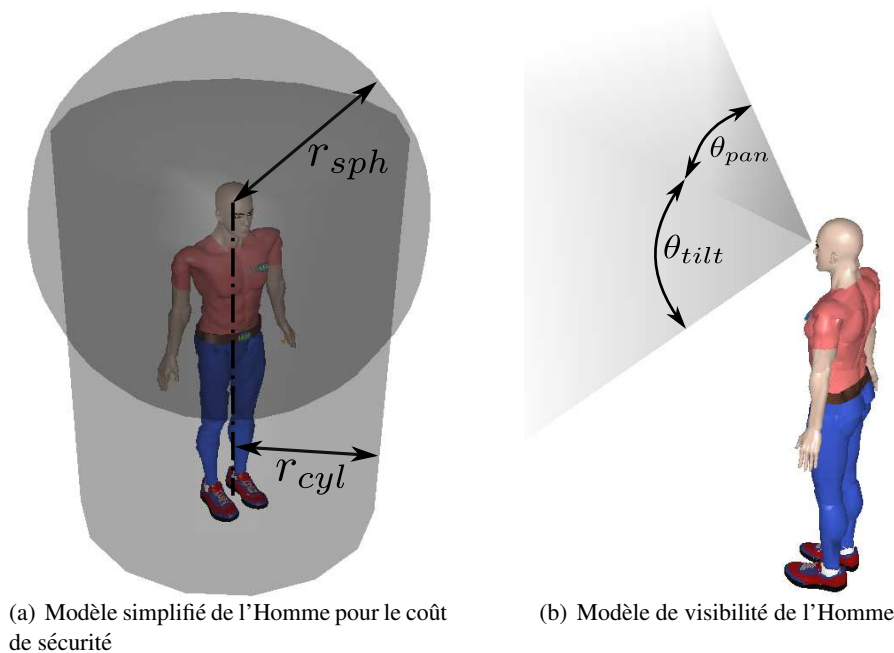


FIGURE 5.1 – Modèles des coûts de sécurité et de visibilité

Chaque contrainte est représentée par une carte de coût tridimensionnelle, ces cartes de coût élémentaires sont ensuite combinées par une somme pondérée :

$$cost(h, q) = \sum_{i=1}^3 w_i c_i(h, q) \quad (5.1)$$

où les w_i sont les poids, h la posture de l'Homme, q la configuration du robot et c_i , les coûts.

Dans l'implémentation actuelle, les poids sont définis empiriquement et les fonctions de coût sont évaluées "à la volée" lors de la planification.

5.1.2 Planification du chemin

En fonction de la présence ou non de l'Homme dans la scène, la planification du chemin est réalisée par les méthodes T-RRT, si présence de l'Homme, ou RRT sinon (Section 2.2.3).

Dans les deux cas, le chemin $\mathcal{P}(s)$ ($s \in \mathbb{R}^+$) résultant est composé d'un ensemble de configurations du robot (noeuds) reliées par des lignes droites (arrêtes). Prenons l'exemple d'un système à deux dimensions résolu par la méthode RRT. Les figures 5.2(a) et 5.2(b) représentent respectivement les positions initiales et finales du palet jaune. Le chemin obtenu est représenté sur la figure 5.2(c) (lignes vertes reliant les sphères). Les sphères représentent les configurations intermédiaires du chemin.

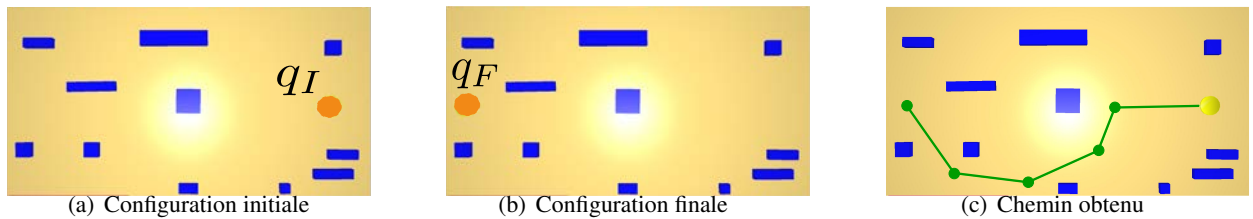


FIGURE 5.2 – Exemple de planification de chemin dans un espace à 2 dimensions

5.2 Du chemin à la trajectoire

Nous proposons de générer une trajectoire à partir d'un chemin en utilisant les méthodes présentées dans les deux chapitres précédents (Chapitres 3 et 4). La génération de la trajectoire est réalisée en deux étapes illustrées sur la figure 5.3.

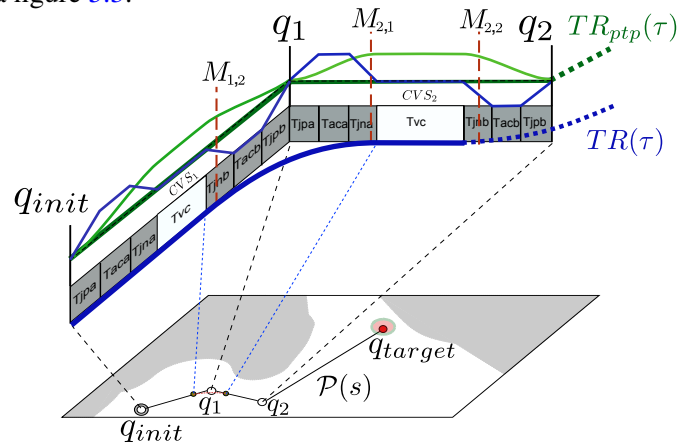


FIGURE 5.3 – Du chemin \mathcal{P} à la trajectoire lissée TR

La première étape consiste à calculer une trajectoire passant par l'ensemble des configurations du chemin \mathcal{P} . Cette trajectoire, que nous nommons TR_{ptp} est composée de mouvements point à point (Section 3.5.1) et comporte donc des mouvements d'arrêt à chaque configuration définissant un sommet.

La deuxième étape consiste à lisser ces mouvements d'arrêt afin d'obtenir une trajectoire TR plus courte en temps. Le lissage utilise le même modèle 3D que celui de l'étape de la recherche du chemin. Les collisions sont ainsi testées lors du calcul du mouvement de transition en chaque noeud. Si une

collision apparait lors du lissage du mouvement d'arrêt pour un noeud q_i , alors le mouvement ne sera pas lissé pour ce noeud et l'arrêt sera conservé.

Nous détaillons dans la suite une méthode de lissage des mouvements d'arrêt basée sur le calcul du mouvement en temps imposée par la méthode *3-segments* (Section 4.2).

5.3 Lissage des mouvements d'arrêt

5.3.1 Méthodes de Lloyd et al. et de Macfarlane

Dans [Lloyd 93], Lloyd et Hayward présentent une méthode pour calculer un mouvement de transition à un instant t_s (figure 5.4) entre deux polynômes $x_1(t)$ et $x_2(t)$ linéarisés aux extrémités du mouvement de transition.

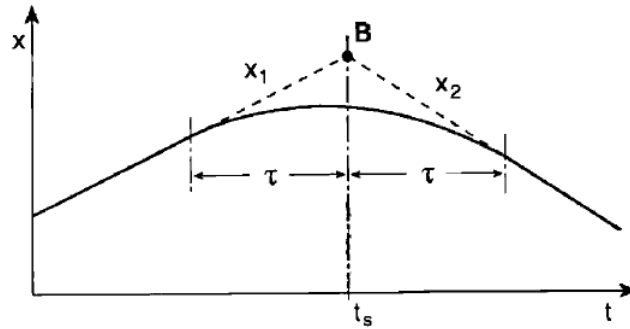


FIGURE 5.4 – Mouvement de transition mono-dimensionnel [Lloyd 93]

La durée du mouvement de transition $2 * \tau$ est estimée pour obtenir une accélération moyenne a_r sur le mouvement de transition tel que :

$$2\tau \approx \frac{\| \dot{x}_2(t_s + \tau) - \dot{x}_1(t_s - \tau) \|}{a_r} \quad (5.2)$$

Le mouvement de transition est paramétré à l'aide d'une variable adimensionnée :

$$\forall s \in [0, 1] \quad s = \frac{t - t_s + \tau}{2\tau} \quad (5.3)$$

Dans une première solution, Lloyd propose d'utiliser un polynôme de degré 5 pour assurer la continuité en accélération et vitesse sur l'intervalle de temps $t \in [t_s - \tau, t_s + \tau]$. Cependant pour permettre l'adaptation du mouvement de transition, il propose d'utiliser un polynôme $x(s)$ défini par la moyenne convexe des polynômes $x_1(s)$ et $x_2(s)$ linéarisés respectivement en $t = t_s - \tau$ et $t = t_s + \tau$:

$$x(s) = x_1(s) + \alpha(s)(x_2(s) - x_1(s)) - k\beta(s)v_d \quad (5.4)$$

avec :

- $\alpha(s) = 6s^5 - 15s^4 + 10s^3$, un polynôme de degré 5 avec des dérivées première et seconde nulles pour $s = 0$ et $s = 1$ et $\alpha(0) = 0$, $\alpha(1) = 1$;
- $v_d = \dot{x}_2(t_s + \tau) - \dot{x}_1(t_s - \tau)$;

- $\beta(s) = s^6 - 3s^5 + 3s^4 - s^3$ un polynôme de degré 6 (de mêmes conditions aux limites que $\alpha(s)$) permettant de réduire la variation d'accélération du mouvement de transition ;
- k un paramètre pour modifier le chemin.

La figure 5.5 illustre le résultat pour différente valeur du paramètre k .

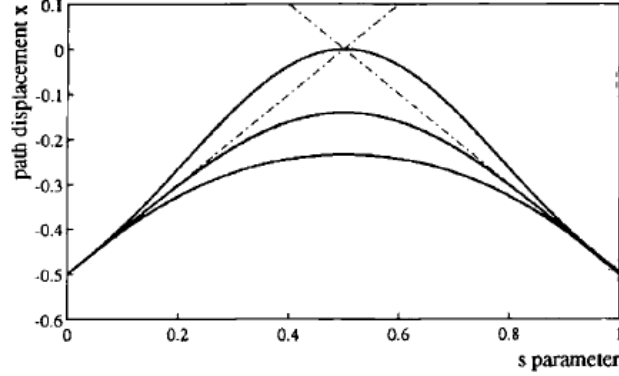


FIGURE 5.5 – Mouvement de transition de deux segments de droite avec $k=0$ (haut), $k=6$ (milieu) et $k=15/2$ (bas) [Lloyd 93]

Ainsi, Lloyd propose une méthode permettant d'obtenir un mouvement de transition assurant la continuité de l'accélération et de la vitesse à ses extrémités. Avec cette méthode l'accélération moyenne et sa variation sont limitées. Cependant, l'accélération maximale et le jerk maximal ne peuvent pas être imposés. Remarquons que cette méthode s'applique à des trajectoires quelconques multidimensionnelles.

Macfarlane [Macfarlane 03] propose une restriction de cette méthode pour le cas particulier où les chemins x_1 et x_2 sont des segments de droite. Les vitesses aux limites du mouvement de transition doivent être identiques (notée s_{blend}) et les accélérations nulles. Les instants de début et de fin du mouvement de transition sont calculés à partir d'un paramètre *tightness* donné qui représente la distance maximale entre les extrémités p_{1a} et p_{1b} du mouvement de transition et le point d'intersection des deux segments de droite p_1 (figure 5.6). Le paramètre τ définissant les instants des points p_{1a} et p_{1b} est déterminé en considérant que l'évolution de la vitesse est linéaire :

$$\tau = \frac{tightness}{s_{blend}} \quad (5.5)$$

A partir de l'accélération maximale a_{max} et du jerk maximal j_{max} désirés, du paramètre *tightness* et de l'angle θ formé par les deux segments de droite, la vitesse s_{blend} est calculée :

$$s_{blendMaxAcc} = \sqrt{\frac{4a_{max}tightness}{5\sqrt{\frac{1+\cos(\theta)}{2}}}} \quad (5.6)$$

$$s_{blendMaxJerk} = \left(\frac{2j_{max}tightness^2}{15\cos(\frac{\theta}{2})}\right)^{1/3} \quad (5.7)$$

$$s_{blendMax} = \min(s_{blendMaxAcc}, s_{blendMaxJerk}) \quad (5.8)$$

Cette méthode est uniquement appliquée aux translations de l'espace cartésien. Outre la nécessité d'impo-

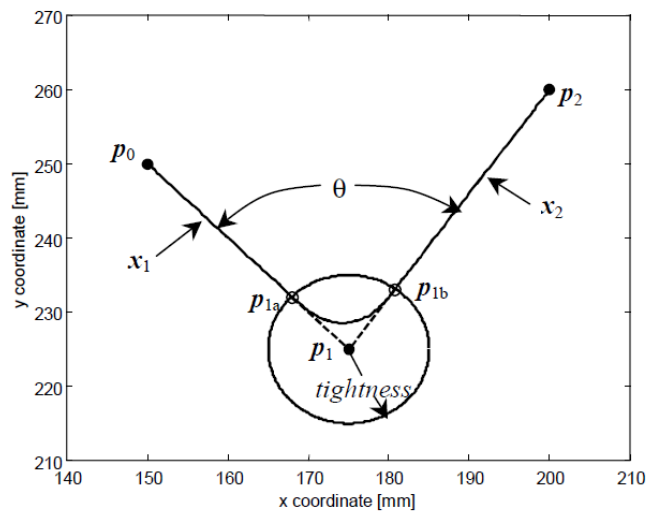


FIGURE 5.6 – Mouvement de transition de deux segments de droite en fonction d'un paramètre de distance *tightness* [Macfarlane 03]

ser des accélérations nulles et des vitesses égales aux extrémités du mouvement de transition, la méthode proposée par Macfarlane nécessite de connaître le mouvement x_2 pour pouvoir planifier le mouvement x_1 .

5.3.2 Méthode de lissage proposée

Nous proposons une méthode basée sur l'algorithme de génération de trajectoire en temps minimal (alg. 2) et sur la méthode *3-segments* de calcul en temps imposé pour le lissage des arrêts (Section 4.2).

La trajectoire TR_{ptp} (figure 5.3) entre les deux premiers noeuds q_{init} et q_1 est un mouvement point à point en ligne droite de durée $T_{(q_{init}q_1)}$. De même le mouvement entre q_1 et q_2 est un mouvement point à point de durée $T_{(q_1q_2)}$. Le mouvement d'arrêt est lissé entre les points $M_{1,2}$ et $M_{2,1}$.

Notation : Nous notons les points qui limitent le lissage $M_{i,j}$, l'indice i correspond à l'index du mouvement point à point (le premier de la trajectoire a un index de 1). L'indice $j \in \{1,2\}$ vaut 1 si ce point est l'extrémité finale du mouvement de transition avec le mouvement point à point précédent et inversement pour $j = 2$.

Choix des points $M_{i,j}$

Considérons le mouvement de transition au voisinage du point q_1 situé à l'instant t_{q_1} :

$$t_{q_1} = t_I + T_{(q_{init}q_1)} \quad (5.9)$$

Pour simplifier, nous avons posé $t_I = 0$ pour l'origine temporelle de la trajectoire.

Les positions temporelles $t_{M_{1,2}}$ et $t_{M_{2,1}}$ des points $M_{1,2}$ et $M_{2,1}$ sont déterminées à partir d'un paramètre de temps donné τ tel que :

$$M_{1,2} = TR_{ptp}(t_{q1} - \max(\tau, \frac{T_{(q_{init}q_1)}}{2})) \quad (5.10)$$

$$M_{2,1} = TR_{ptp}(t_{q1} + \max(\tau, \frac{T_{(q_1q_2)}}{2})) \quad (5.11)$$

Ainsi, lorsque τ est nul, le mouvement s'arrête au point q_1 . Lorsque τ est satisfait l'équation 5.12, le mouvement de transition relie les milieux des segments de droite (q_{init}, q_1) et (q_1, q_2) car le profil de vitesse est symétrique par rapport à ce point (Section 3.2).

$$\tau \geq \max\left(\frac{T_{(q_{init}q_1)}}{2}, \frac{T_{(q_1q_2)}}{2}\right) \quad (5.12)$$

En pratique, sauf précision contraire, nous choisissons par défaut les points $M_{i,j}$ tel que le mouvement de transition commence à la fin du segment à vitesse constante du premier mouvement point à point (P_1, P_2) ; le mouvement de transition se termine au début du segment à vitesse constante du deuxième mouvement point à point (P_2, P_3) .

Remarque : Pour une même valeur du paramètre τ , la distance euclidienne entre les points $M_{i,j}$ et le point q_i correspondant varie en fonction des paramètres cinématiques des mouvements point à point.

Calcul du mouvement de transition

Considérons une trajectoire de dimension n . Les instants $t_{M_{i-1,2}}$ et $t_{M_{i,1}}$ de début et de fin du mouvement de transition à la configuration q_i sont identiques pour l'ensemble des n dimensions. La méthode de calcul est décrite par l'algorithme 9. La première étape consiste à calculer les mouvements en temps optimaux pour chaque axe pour déterminer la durée T_{imp} du mouvement de transition. La méthode 3-segments de calcul de mouvement en temps imposé est ensuite appliquée pour chaque axe.

Algorithme 9 : Calcul d'un mouvement de transition au voisinage du noeud q_i

début

Détermination des points de commutation $M_{i-1,2}$ et $M_{i,1}$ (eq. 5.11 et 5.10)

pour *Pour chaque dimensions n_i* **faire**

 Calcul du mouvement mono-dimensionnel en temps minimal (alg. 2)

 Calcul de la durée du mouvement mono-dimensionnel en temps minimal $T_{opt}[i]$

fin

Détermination de la durée du mouvement de transition $T_{imp} = \max(\forall i \in [1, n] \mid T_{opt}[i])$

pour *Pour chaque dimensions n_i* **faire**

 Calcul des triplets de segments de courbes cubiques à partir de la méthode 3-segments
 (Section 4.2)

fin

fin

La figure 5.7 illustre une application de la méthode pour le cas d'un mouvement défini par trois points P_1, P_2, P_3 et par les contraintes cinématiques $V_{max} = 0.1m/s, A_{max} = 0.3m/s$ et $J_{max} = 0.9m/s$. Les mouvements de transitions sont calculés pour plusieurs valeurs du paramètre τ .

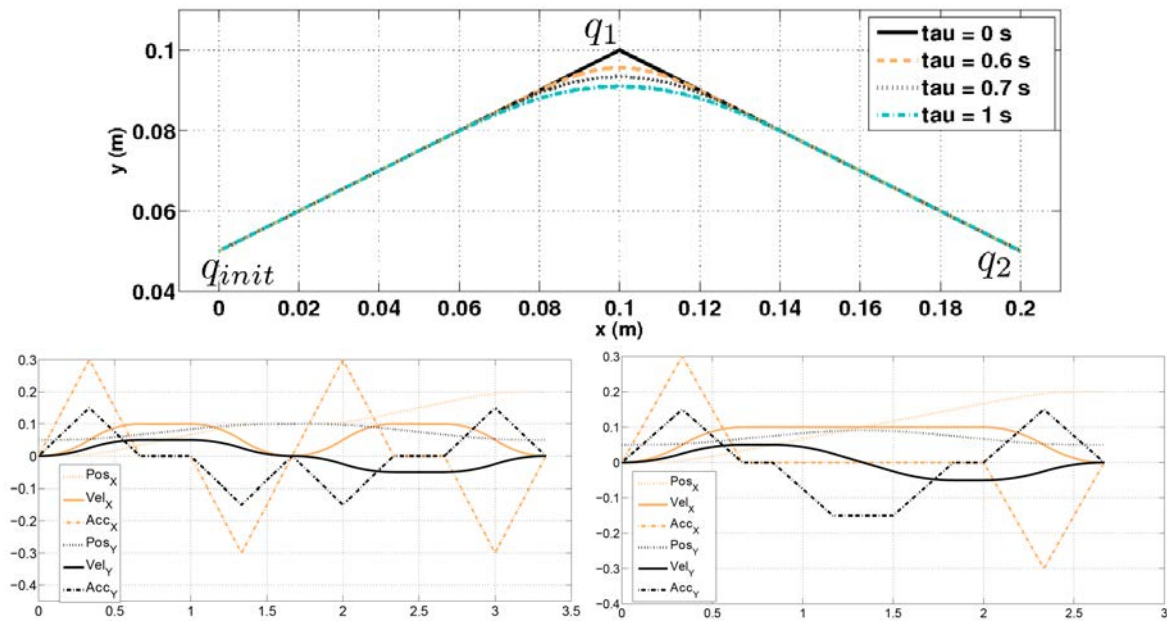


FIGURE 5.7 – Mouvement de transition pour un angle entre les droites d'environ 127° (haut), Évolutions de la position, de la vitesse et de l'accélération en fonction du temps pour un mouvement point à point ($\tau = 0$) (bas-gauche) et $\tau = 1$ s (bas-droite)

La méthode que nous proposons assure la continuité en vitesse et accélération pour chaque dimension. Les vitesses initiales et finales des mouvements de transition peuvent être différentes et les accélérations non nulles. La durée des mouvements de transition est calculée en prenant en compte les contraintes cinématiques de chaque dimension en utilisant l'algorithme 2. Cette méthode garantit donc que les évolutions de la vitesse, de l'accélération et du jerk sont limités. Cependant, dans certains cas, les contraintes peuvent ne pas être respectées par l'approximation *3-segments*. En pratique, nous introduisons un pourcentage (10%) de dépassement pour chaque contrainte. Si pour un mouvement de transition le dépassement des contraintes cinématiques est trop important, ce mouvement n'est pas lissé pour respecter la contrainte de confort de l'Homme.

5.4 Application aux robots manipulateurs

Afin de mieux expliquer la méthode, nous l'appliquons ici à un exemple de tâche de saisie d'objet, la cassette grise de la figure 5.8. Le chemin du centre de l'organe effecteur du robot (la main) est décrit par les segments de droites verts sur la figure 5.9. Sur ce chemin, les sphères représentent les configurations initiale, finale et intermédiaires. Le chemin de la trajectoire point à point TR_{ptp} est identique au chemin planifié. Cette trajectoire s'arrête à chaque noeud intermédiaire. La trajectoire lissée TR est représentée par la courbe noire. Nous remarquons que la trajectoire s'arrête au premier noeud car un lissage à ce voisinage aurait induit une collision¹ entre la main du robot et l'environnement. La planification du chemin et de la trajectoire a été réalisée dans l'espace cartésien du robot en considérant la plateforme

1. Remarque : une autre solution serait de calculer un chemin qui passe plus loin de l'obstacle mais ce n'est pas l'objet de nos travaux.

fixe. La section suivante présente la méthode permettant de prendre en compte la redondance du robot.

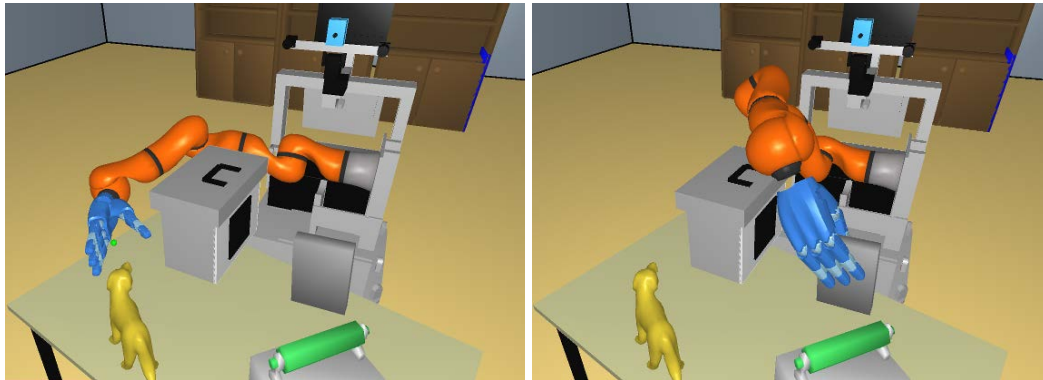


FIGURE 5.8 – Configuration initiale et configuration de prise du robot Jido

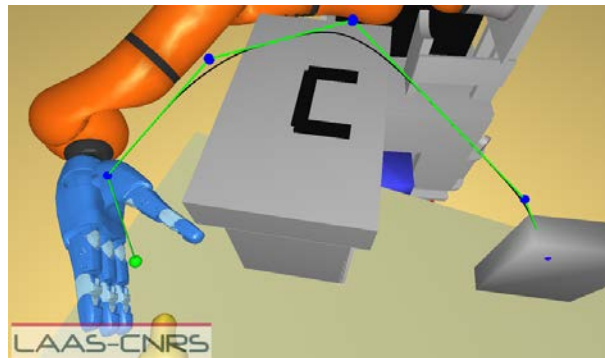


FIGURE 5.9 – Trajectoires TR_{ptp} et TR dans l'espace cartésien pour saisir la cassette

5.5 Planification dans l'espace cartésien

5.5.1 Génération de la trajectoire TR lissée dans l'espace cartésien

Dans la section 2.2.2, nous avons vu que pour représenter la posture du robot dans l'espace cartésien, il est nécessaire d'utiliser un vecteur X_I comprenant :

- la situation de la base du robot ;
- la/les situation(s) du/des organe(s) effecteur(s) ;
- la configuration des axes redondants des bras s'ils possèdent plus de six degrés de liberté ;
- la/les configuration(s) de la/des main(s) ;
- la configuration de la tête.

Dans la suite, nous considérons que la plateforme est fixe. Pour un système évoluant dans l'espace, six paramètres indépendants permettent de définir la situation de l'organe effecteur. Pour la planification, le système est décomposé en parties *passives* et *actives* correspondant respectivement aux variables dépendantes et indépendantes [Cortés 04, Han 01]. Ainsi un robot manipulateur à six degrés de liberté, est décomposé de la façon suivante : les variables indépendantes (actives) sont les six degrés de liberté

(position et orientation) de l'organe effecteur et les variables articulaires sont les variables dépendantes (passives).

Dans le cas du robot Jido, le bras manipulateur est constitué de sept degrés de liberté et est donc redondant. En plus de la situation de l'organe effecteur, une articulation du bras est choisie comme redondante et devient une variable *active*. Remarque : Si le mouvement d'une plateforme holonome était considérée, les degrés de liberté de celle-ci seraient des variables actives.

Lors de la planification du chemin dans l'espace cartésien, seules les variables *actives* sont échantillonnées en utilisant, suivant les cas, l'algorithme RRT ou T-RRT. Les variables passives sont calculées dans un deuxième temps en résolvant la cinématique inverse du bras avant de tester la validité de la configuration échantillonnée du robot (butées et collision) (Section 2.2.3). Lors du test de la validité d'une arrête entre deux configurations, la fonction de cinématique inverse est également appelée.

Pour effectuer l'interpolation entre deux configurations, nous représentons la situation de l'organe effecteur par un déplacement : 3 paramètres pour la position et 3 paramètres pour l'orientation (représentation vecteur-angle avec la norme du vecteur égal à l'angle) (Section 2.2.1.2). Nous avons implémenté une méthode locale d'interpolation entre deux configurations. Cette méthode locale prend en paramètre deux configurations (avec leurs conditions cinématiques) et les contraintes cinématiques imposées (J_{max} , A_{max} et V_{max}) de chaque axe actif. Après application de la méthode locale entre chaque configuration intermédiaire, la trajectoire TR_{ptp} obtenue est composée de mouvements point à point de dimension n (n , nombre d'axes actifs) soit, pour Jido, $n = 22$ dimensions (6 pour l'organe effecteur, 1 pour l'axe redondant du bras manipulateur, 13 pour la main et 2 pour la tête).

La trajectoire TR lissée dans l'espace cartésien est ensuite obtenue par la méthode décrite dans le paragraphe précédent appliquée sur les axes actifs (Section 5.3).

5.5.2 Conversion de la trajectoire dans l'espace articulaire du robot

Dans le cas du robot Jido équipé d'un bras Kuka, il n'est pas possible d'utiliser le contrôleur cartésien de Kuka car il ne permet pas de spécifier l'évolution de l'articulation indépendante. Pour pouvoir exécuter ce mouvement, les trajectoires des axes passifs sont obtenues en discrétisant la trajectoire TR définie dans l'espace cartésien et en effectuant la cinématique inverse pour chaque échantillon. La trajectoire TR est discrétisée à la période de fonctionnement du contrôleur du robot. Cette opération permet d'obtenir la position, et par dérivation, la vitesse et l'accélération de l'ensemble des degrés de liberté du robot.

Cependant, cette discrétisation supprime, d'une part, la notion temporelle et nécessite d'autre part une grande quantité de données pour représenter la trajectoire.

Nous utilisons la méthode d'approximation de trajectoire présentée dans le chapitre 4 pour approximer cette trajectoire discrétisée et ainsi obtenir une description compacte de la trajectoire. A la différence de l'approximation dans l'espace cartésien, l'erreur de trajectoire prise en compte par l'algorithme d'approximation est la valeur maximale des erreurs de trajectoire de chaque degré de liberté.

La trajectoire approximée TR_{app} obtenue est une fonction du temps composée de suite de segments de courbes cubiques pour chaque variable articulaire du robot. Chaque suite est composée du même nombre de segments synchronisés en temps.

Cependant, les mouvements des axes passifs n'étant pas planifiés, ils peuvent dépasser les limites cinématiques du robot. Dans ce cas, la trajectoire ne peut pas être exécutée directement. Pour adapter

la trajectoire, si la tâche le permet, nous remplaçons le paramètre de temps t de la trajectoire par une application $\alpha, \mathbb{R} \rightarrow \mathbb{R}$. La fonction α va permettre de modifier l'incrément de temps pendant l'exécution de la trajectoire et donc permettre de ralentir (ou d'accélérer) l'exécution.

La période du contrôleur de trajectoire est notée ΔT . Dans le cas d'une exécution classique, l'application α est définie par :

$$\alpha(t) = t \quad (5.13)$$

soit, en notation discrète :

$$\alpha(k\Delta T) = \alpha((k-1)\Delta T) + \Delta T \quad (5.14)$$

La trajectoire exécutée est $TR_{app}(\alpha(t))$.

L'introduction de l'application α permet ainsi de modifier la loi de mouvement de la trajectoire TR_{app} et donc permet d'adapter l'évolution de chaque articulation du robot de manière synchronisée.

Pour déterminer l'application α dans le cas où l'on désire adapter la loi de mouvement, nous déterminons d'abord pour chaque instant de la trajectoire TR_{app} le dépassement β des vitesses de chaque axe par rapport aux vitesses maximales correspondantes (les valeurs maximales utilisées ici sont par défaut les limites acceptées par le système). On obtient ainsi :

$$\forall k\Delta T \in [t_I, t_F],$$

$$\beta(k\Delta T) = \begin{cases} 1 & \text{si } \forall j \in [1, n], {}_jV(k\Delta T) \leq {}_jV_{max}^{mot} \\ \min \left(\forall j \in [1, n] \mid \frac{{}_jV_{max}^{mot}}{{}_jV(k\Delta T)} \right) & \text{sinon} \end{cases} \quad (5.15)$$

avec n le nombre de degrés de liberté contrôlés, ${}_jV(t)$, l'évolution de la vitesse de l'articulation j et ${}_jV_{max}^{mot}$, la vitesse maximale de l'articulation j .

Nous obtenons ainsi :

$$\alpha(k\Delta T) = \alpha((k-1)\Delta T) + \beta(k\Delta T)\Delta T \quad (5.16)$$

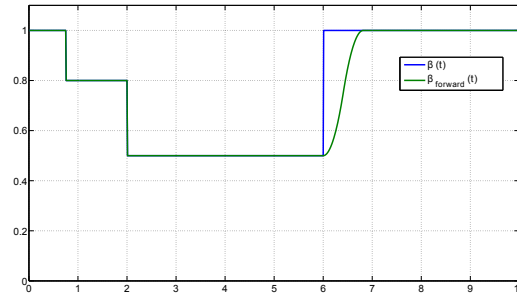
avec $\alpha(0) = 0$.

Cependant, la trajectoire $TR_{app}(\alpha(t))$ ne peut pas être exécutée directement car elle introduirait des discontinuités en vitesse du fait de la discontinuité de β . Pour lisser l'évolution de β , nous appliquons une variante de la méthode décrite dans la section 3.3.5.2 qui permet d'anticiper la variation de β . La fonction β lissée est notée β_{smooth} . Le lissage est effectué en trois étapes par l'algorithme 10.

Algorithme 10 : Lissage de la loi de variation de la loi de mouvement**début**

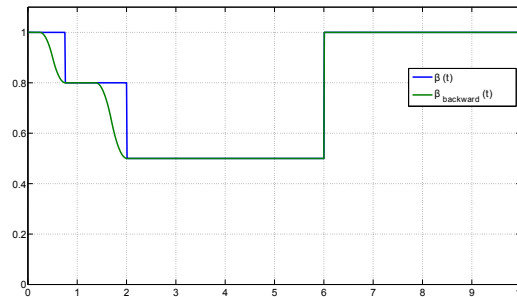
Application de la méthode 3.3.5.2 sur l'évolution de β en faisant varier le temps de t_I à t_F avec un pas ΔT , la courbe obtenue est nommée $\beta_{forward}$:

$$\beta_{forward}(k\Delta T) = \begin{cases} \beta(k\Delta T) & \text{si } \beta(k\Delta T) < \beta((k-1)\Delta T) \\ f_{smooth}(\beta) & \text{sinon} \end{cases} \quad (5.17)$$



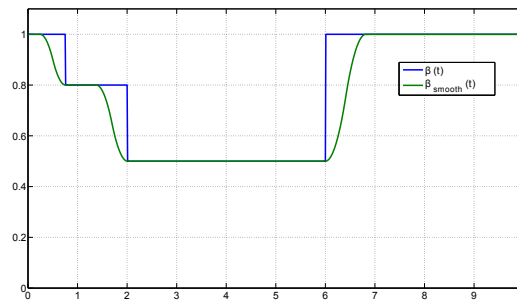
Application de la méthode 3.3.5.2 sur l'évolution de β en faisant varier le temps de t_F à t_I avec un pas $-\Delta T$, la courbe obtenue est nommée $\beta_{backward}$:

$$\beta_{backward}(k\Delta T) = \begin{cases} \beta(k\Delta T) & \text{si } \beta(k\Delta T) > \beta((k+1)\Delta T) \\ f_{smooth}(\beta) & \text{sinon} \end{cases} \quad (5.18)$$



β_{smooth} est enfin obtenu en prenant le minimum entre $\beta_{forward}$ et $\beta_{backward}$:

$$\beta_{smooth}(k\Delta T) = \min(\beta_{forward}(k\Delta T), \beta_{backward}(k\Delta T)) \quad (5.19)$$

**fin**

La méthode présentée ci-dessus permet de modifier la vitesse de chaque articulation du robot pour satisfaire les limites en vitesse. Nous avons supposé que la trajectoire résultante respecte les contraintes d'accélération. Dans le cas contraire, il est possible de déterminer une fonction β^{acc} équivalent à β pour prendre en compte les dépassements d'accélération pendant l'exécution. En pratique, pour l'interaction Homme-robot, les contraintes cinématiques de la trajectoire étant petites par rapport aux capacités du système, il n'est pas utile de vérifier les dépassements d'accélération.

Dans cette section nous avons présenté, une méthode permettant de calculer une trajectoire à partir d'un chemin qui peut être défini soit dans l'espace articulaire, soit dans l'espace cartésien du robot.

5.6 Adaptation de la loi de mouvement pendant l'exécution

Pendant l'exécution de la trajectoire, l'humain situé dans l'espace de travail du robot peut bouger ; le robot peut donc le mettre en danger. Nous proposons d'utiliser les modèles géométriques du robot et de l'Homme actualisés à chaque itération pendant l'exécution pour garantir la sécurité et le confort de l'humain. Nous choisissons de prendre en compte la moyenne pondérée noté $cost$ des contraintes de sécurité et de visibilité introduites dans la section 5.1.1. La méthode pour modifier la loi de mouvement est identique à celle présentée dans la section précédente. Les coûts étant élevés lorsque la distance Homme-robot est faible ou lorsque le robot est en dehors du champ de vue de l'humain, le coût pris en compte est $cost_{inv} \in [0, 1]$ tel que :

$$cost_{inv}(k\Delta T) = 1 - cost(k\Delta T) \quad (5.20)$$

Le coût $cost_{inv}$ est ensuite lissé en-ligne par la fonction f_{smooth} présentée dans la section 3.3.5.2.

Dans le cas où la trajectoire $TR(\alpha(t))$ est planifiée dans l'espace cartésien, la loi $\alpha(t)$ est évaluée à chaque itération :

$$\alpha(k\Delta T) = \alpha((k-1)\Delta T) + f_{smooth}(\min(cost_{inv}(k\Delta T), \beta_{smooth}(k\Delta T)) \cdot \Delta T) \quad (5.21)$$

La section 6.5 présente une expérimentation relative à cette méthode.

5.7 Conclusion

Nous avons présenté un générateur de trajectoire permettant le calcul de trajectoire limitant la vitesse, l'accélération et le jerk pour un robot redondant sans considérer la non holonomie. Le générateur, intégré au niveau du planificateur de chemin, est d'abord présenté dans le cas d'une planification dans l'espace des configurations puis étendu à la planification dans l'espace cartésien. L'approche est générale et peut être appliquée sur des systèmes complexes à deux bras/deux mains. Nous avons proposé une méthode originale permettant de convertir une trajectoire cartésienne en une trajectoire articulaire. Nous avons vu que la planification dans l'espace cartésien peut introduire un dépassement des limites cinématiques du robot. Pour palier le problème du dépassement des vitesses articulaires, nous avons proposé une méthode de correction de la loi d'évolution de la trajectoire. Enfin, nous avons présenté une approche permettant de modifier en ligne la loi d'évolution temporelle sur la trajectoire et montré son utilité pour la prise compte de la présence de l'humain pendant l'exécution du mouvement. Dans le chapitre suivant, nous présentons l'intégration des méthodes proposées dans l'architecture globale d'un robot.

6

Expérimentations

Durant cette thèse, l'intégration des travaux et les expérimentations ont pris une part importante. L'ensemble des travaux présentés dans ce manuscrit sont intégrés sur la plateforme *évolutive* Jido (Figure 2.4, p. 10) et en cours d'intégration sur la plateforme PR2. Nous qualifions Jido de plateforme *évolutive*, car cette plateforme est constamment en évolution. Comme nous l'avons souligné dans le deuxième chapitre, au début de nos travaux Jido était équipé d'un bras PA10-6C Mitsubishi à six degrés de liberté muni d'une pince à deux doigts. L'intégration sur cette plateforme du planificateur de trajectoire dans l'espace cartésien est illustré par la première expérimentation présentée ci-après. Nous avons ensuite remplacé le bras initial par un bras Kuka LBR-IV à sept degrés de liberté et développé les composants logiciels de planification/contrôle de trajectoire s'intégrant à l'architecture globale de la plateforme. L'organe effecteur de Jido peut être une main Schunk SAHAND ou une pince à deux doigts. Le détail de la composition matérielle de Jido est présenté dans la section 2.1.

Dans la première partie de ce chapitre, nous présentons l'architecture logicielle de la plateforme Jido actuelle et les composants que nous avons été amené à développer. Dans la deuxième partie, nous présentons différentes applications des travaux présentés dans ce manuscrit.

6.1 Intégration de la génération de trajectoire sur la plateforme Jido

6.1.1 L'architecture logicielle de Jido

Dans la première version de la plateforme (équipée du bras PA10), le planificateur de trajectoire était dissocié du planificateur de chemin. Le générateur de trajectoire interpolait des suites de configurations fournies par le planificateur de chemin sans valider l'absence de collision due à la génération de la trajectoire. Lors du remplacement du PA10 par le bras Kuka, nous avons intégré le générateur de trajectoire au niveau du planificateur de chemin pour avoir une plus grande flexibilité. Notre préoccupation était de

rendre le générateur de trajectoire indépendant de l'architecture matérielle du robot. Le générateur peut ainsi être utilisé sur la plateforme PR2 que le LAAS vient d'acquérir.

L'architecture logicielle de Jido est basée sur l'architecture LAAS (Section 2.1.2) composée de modules GenOM [Fleury 97] accessibles par le gestionnaire de paquet *RobotPkg*¹ Les différents composants de l'architecture de Jido ont été développés et adaptés aux fils des évolutions de Jido. Les figures 6.1 et 6.2 illustrent respectivement la couche fonctionnelle de l'architecture et les principaux composants de la plateforme (pour la figure 6.2, la main est remplacée par la pince à mors parallèles).

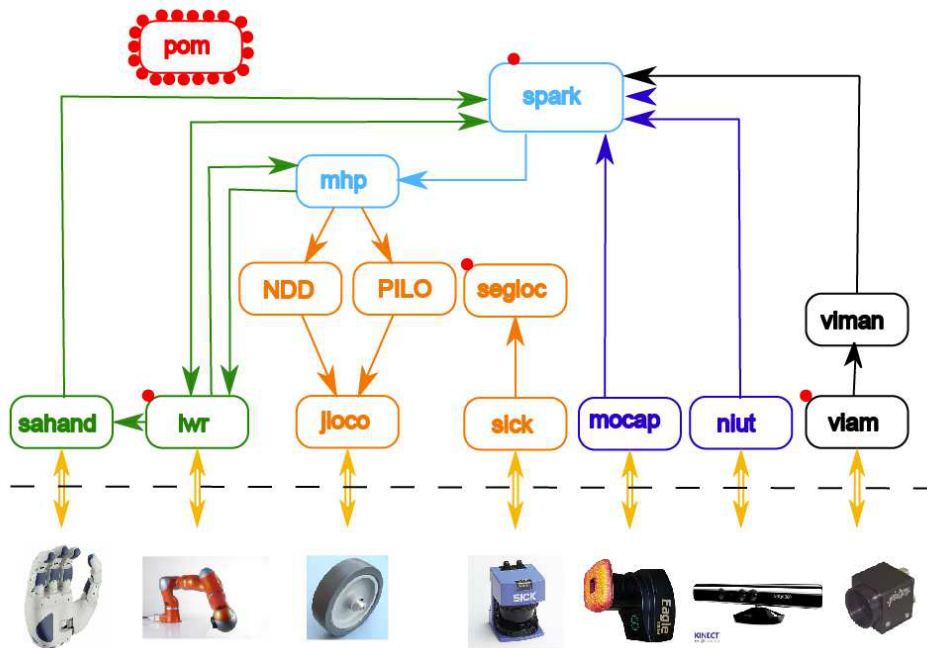


FIGURE 6.1 – Architecture logicielle de la plateforme Jido

La couche fonctionnelle est composée de modules dépendants du matériel et de modules génériques utilisés sur d'autres plateformes comme le PR2. Nous distinguons deux catégories de modules dépendants de l'architecture matérielle, les modules de perception et les modules de contrôle (les modules en gras et soulignés sont ceux que nous avons entièrement développés, les modules auxquels nous avons contribué sont en gras) :

- les modules de perception :
 - module **SICK** : réalise l'acquisition des données du télémètre laser utilisé pour la localisation du robot dans l'environnement ;
 - module **MOCAP** : réalise l'interface avec le système de capture de mouvement ;
 - module **NIUT** : réalise l'acquisition du capteur Kinect et fournit la position des parties du corps de l'Homme ;
 - module **VIAM** : réalise l'acquisition des caméras embarquées ;
 - module **VIMAN** : réalise l'identification et la localisation spatiale des objets. Nous avons développé ce module à partir de la bibliothèque ARTToolKitPlus [Wagner 07]. La fonctionnalité de base de cette bibliothèque est l'identification et la localisation des marqueurs en monoculaire. Chaque

1. <http://homepages.laas.fr/mallet/robotpkg>

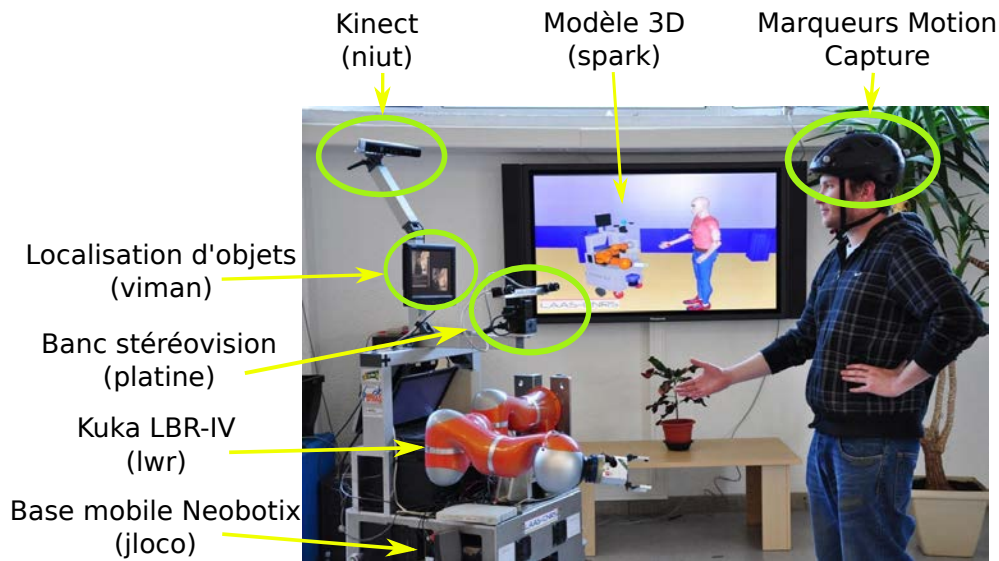
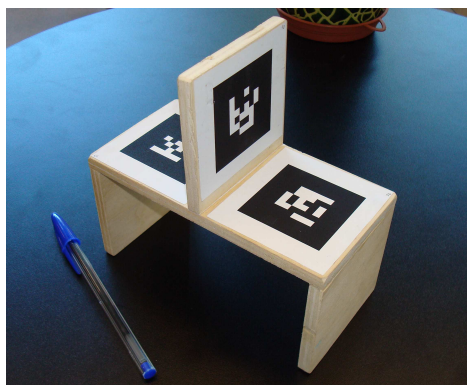
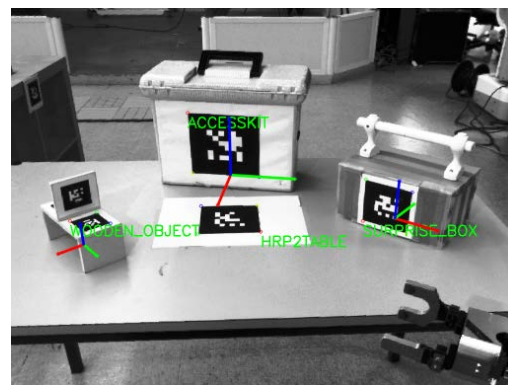


FIGURE 6.2 – Les principaux composants de Jido

marqueur possède un identifiant unique. Nous avons étendu cette librairie pour identifier et localiser des objets avec un banc de stéréovision. Dans l'extension, chaque objet peut contenir plusieurs marqueurs pour être localisé dans différentes orientations. La figure 6.3(a) illustre un objet et ses marqueurs et la figure 6.3(b), l'identification et localisation de plusieurs objets.



(a) Exemple d'objet localisé par le module viman



(b) Localisation multiple d'objets

FIGURE 6.3 – VIMAN, module d'identification et de localisation d'objets

- les modules de contrôle :
 - module **JLOCO** : permet de contrôler la plateforme du robot. Ce module n'est pas utilisé dans nos expérimentations ;
 - module **SAHAND** : permet de contrôler la main anthropomorphe Schunk SAH ;
 - module **LWR** : réalise le contrôle de trajectoire. Le module communique avec le bras Kuka par l'interface FRI (Fast Research Interface) fournie par Kuka. La communication est effectuée par le protocole réseau UDP. Le FRI propose plusieurs mode de contrôle (position, couple, impédance...). Le module LWR permet d'utiliser le mode de contrôle en position dans l'espace articulaire et le mode de contrôle en impédance dans l'espace cartésien. Bien que le robot Jido

ne possède qu'un seul bras manipulateur, le module LWR est conçu pour contrôler deux bras avec deux mains de manière synchronisée. Le LAAS possédant deux bras de ce type, nous avons validé cette fonctionnalité. Ce module observe les forces pour arrêter le mouvement dès qu'une force extérieure apparaît sur le bras. Cette détection permet de détecter le contact objet/table pour les tâches de pose d'objet. Le module intègre enfin des capacités de détection d'auto-collision basée sur le détecteur PQP [Larsen 99]. Cette extension est implémenté dans une librairie nommée *coldman-libs* (*COLL*ision *DE*tectioN *MAN*ager) et a été développé par Jean-Philippe SAUT.

Le module POM maintient les positions (repères) des différents composants matériels du robot.

Le module **SPARK** maintient le modèle 3D de l'environnement. L'ensemble des modèles CAO de chaque objet, du robot et de l'Homme sont connus et leurs positions sont actualisées à partir des informations fournies par les modules de perception et par les modules de contrôle.

Le module **MHP** (*Manipulation in Human Presence*) fournit les capacités de planification de mouvement (chemin et trajectoire). Ce module est basé sur la planificateur de mouvement Move3D [Siméon 01] dans lequel nous avons intégré nos travaux de planification de trajectoire. Ce planificateur propose des méthodes génériques de planification de mouvement capables de résoudre des problèmes impliquant des robots ayant des structures cinématiques complexes dans des environnements variés.

L'architecture de planification/contrôle que nous avons mise en place est illustrée sur la figure 6.4.

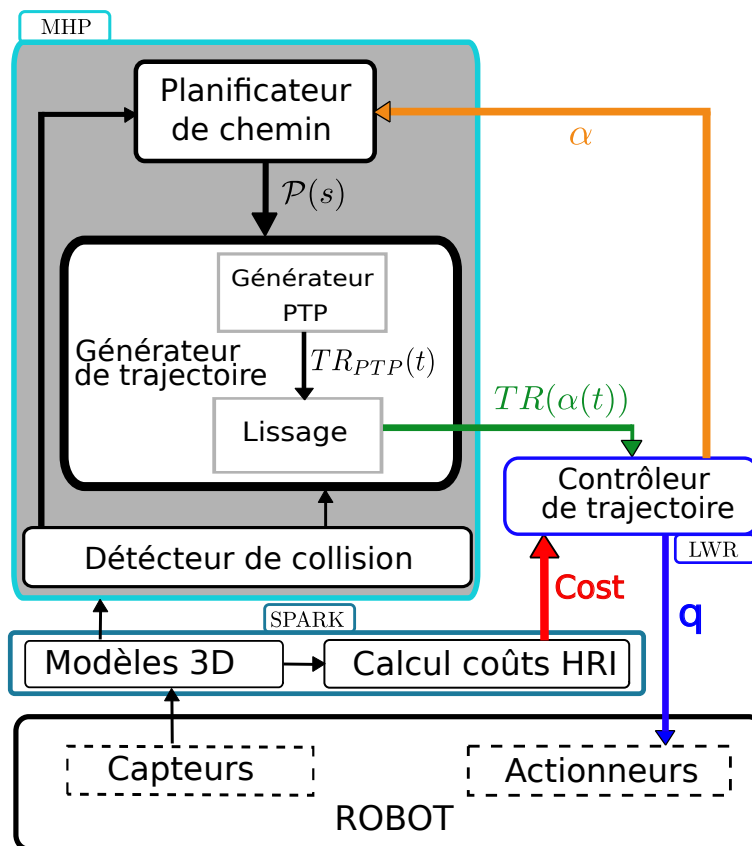


FIGURE 6.4 – Architecture planification/contrôle

Dans le cas de Jido, le module MHP produit des trajectoires décrivant les mouvements de chaque

articulation du bras et de la main. Le module LWR contrôle l'exécution temporelle des trajectoires reçues directement du module MHP. Il contrôle les mouvements de la main par l'intermédiaire du module SAHAND (LWR joue le rôle du décodeur de trajectoire et envoie les positions, vitesse et accélération à chaque composant matériel). La position temporelle α sur la trajectoire est constamment fournie au module MHP pour permettre de prédire les collisions due à l'environnement changeant et donc arrêter l'exécution de la trajectoire si nécessaire.

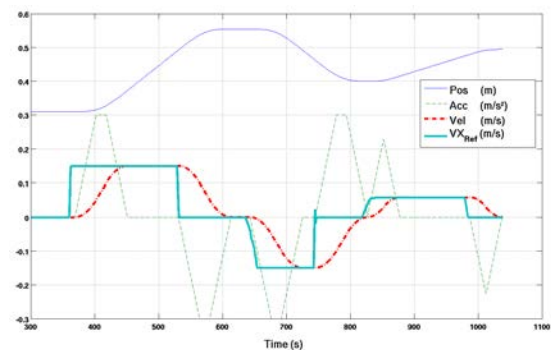
6.2 Les trajectoires dans la boucle de commande

La première expérimentation que nous présentons concerne une application directe du générateur de mouvement mono-dimensionnel de type V (chapitre 3). Cette expérimentation est effectuée avec la première version de Jido équipée du bras manipulateur PA10. L'architecture logicielle (non présentée dans ce manuscrit) comprenait un module GenoM XARM de contrôle dans l'espace cartésien du bras manipulateur communiquant avec un module de contrôle dans l'espace articulaire nommé QARM. Pour cette expérimentation, la commande du bras manipulateur est une commande en vitesse dans l'espace cartésien (position et orientation) [Broquère 08]. Nous avons développé un module JOYSTICK pour acquérir les données analogiques du périphérique Logitech Cordless Rumblepad II. Les consignes provenant du joystick sont traitées indépendamment pour chaque axe de l'espace cartésien par la méthode de suivi de consigne vitesse-accélération (voir Section 3.3.5.2). La figure 6.5(a), illustre l'expérimentation.

La figure 6.5(b) présente l'évolution de l'axe X calculée en-ligne à partir de la consigne correspondante.



(a) Expérimentation de commande en vitesse



(b) Évolution de l'accélération, de la vitesse et de la position de l'axe X pour une consigne en vitesse (V_{XRef})

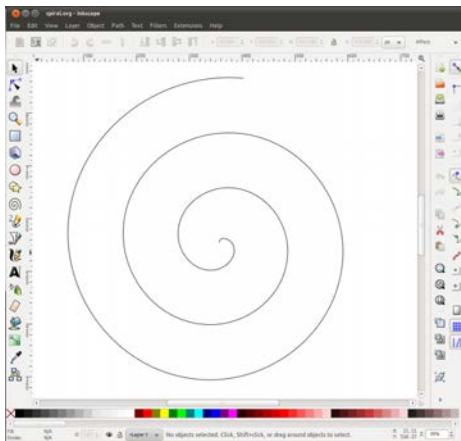
FIGURE 6.5 – Génération en ligne de trajectoire sous contraintes cinématiques

Cette première expérimentation valide l'utilisation en-ligne de l'algorithme de génération de trajectoire.

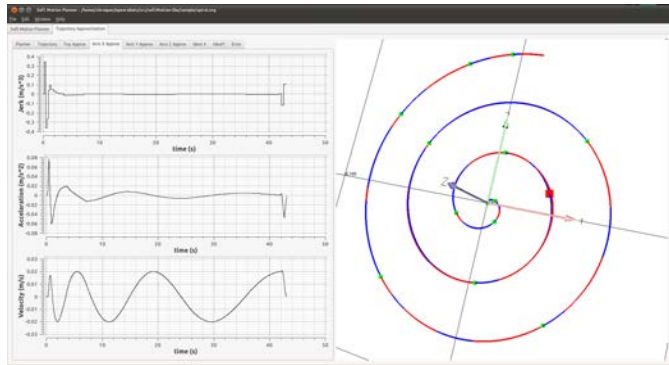
6.3 Approximation de trajectoires : “Écrire et dessiner”

Dans le cadre de la mise au point de l’algorithme 8, nous avons développé une interface graphique permettant de charger des dessins vectoriels au format libre svg² dans lesquels les chemins sont définis par des suites de courbes de Béziérs. De par leur définition, ces courbes peuvent ne pas être continues en vitesse et accélération. Les figures 6.6(a) et 6.6(b) présentent respectivement le dessin avec le logiciel inkscape et l’approximation d’une courbe en spirale. La loi de mouvement imposée est un mouvement point à point à sept segments avec les paramètres $V_{max} = 0.02m/s$, $A_{max} = 0.04m/s^2$ et $J_{max} = 0.12m/s^3$. L’erreur de trajectoire maximale désirée est fixée à 1mm (figure 6.6(c)).

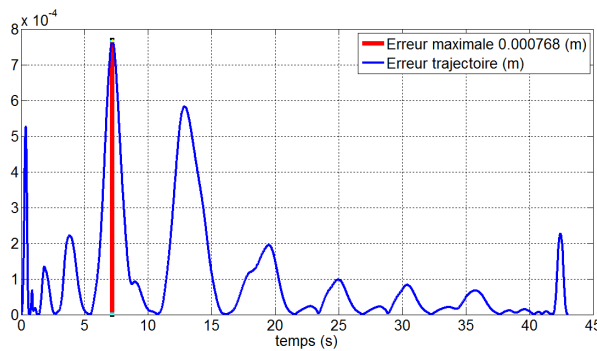
Remarque : la spirale d’entrée de notre logiciel est une courbe de Béziérs qui approxime une spirale d’Archimède (rayon $r = a + b\theta$).



(a) Spirale dessinée sous le logiciel inkscape



(b) Approximation de la courbe en spirale



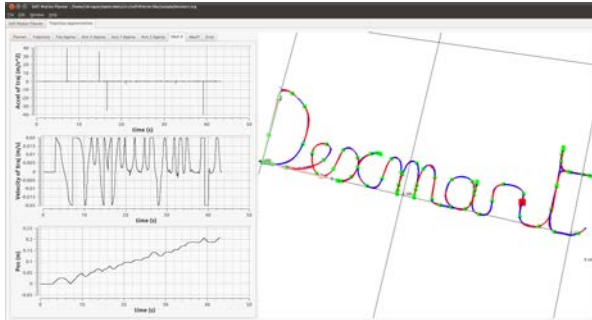
(c) Erreur de trajectoire

FIGURE 6.6 – Approximation d’une trajectoire à partir d’un fichier svg

A partir de cette interface, nous avons approximé une courbe non continue en vitesse représentant le mot « Dexmart » (figure 6.7(a)). La trajectoire approximée est composée de suites de segments de courbe cubique. Elle a été exécutée par le robot Kuka (figure 6.7(b)). Pour cela nous avons utilisé le contrôleur cartésien en impédance du robot. Le contrôleur a permis de lisser les discontinuités en vitesse et accélération. La courbe de départ représentant le mot « Dexmart » comporte des points anguleux et des points de rebroussements non lissés par notre méthode. Pour éviter les discontinuités en vitesse de

2. <http://www.w3.org/TR/SVG>

ces points, la méthode devrait comporter une phase d'analyse de la courbe à approximer pour générer des mouvements d'arrêts en ces points.



(a) Approximation d'une courbe non continue en vitesse et accélération



(b) Exécution sur le robot Kuka

FIGURE 6.7 – Approximation et exécution d'une courbe non continue en vitesse et accélération

6.4 Génération de trajectoire : “Tâche de saisie d’un objet”

L’expérimentation décrite ci-après illustre l’intégration du planificateur de trajectoire dans l’architecture globale du robot Jido. La tâche à réaliser est la saisie de la cassette grise en présence d’un obstacle (figure 6.8). La position de la table, de l’obstacle et de la cassette sont obtenues à partir du module VI-MAN. La séquence d’image sur la figure 6.9 illustre le mouvement planifié. La planification du chemin a été effectuée dans l’espace des configurations du robot.

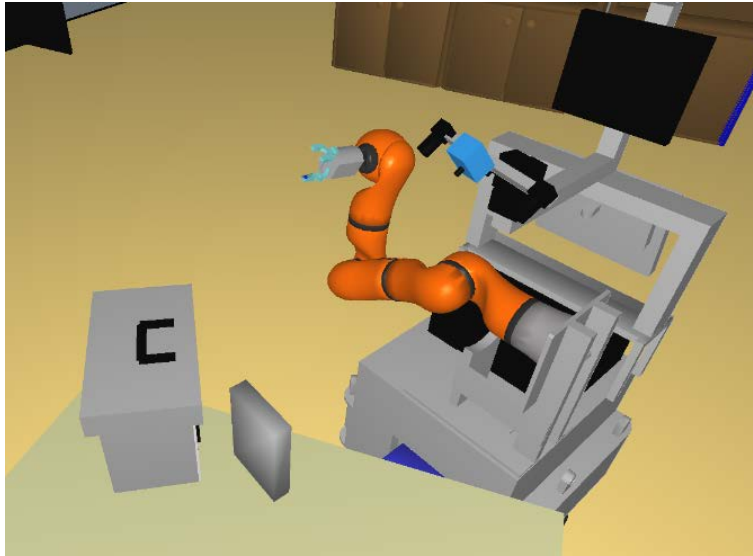


FIGURE 6.8 – Modèle de la scène pour la tâche de saisie mise en jour en ligne



FIGURE 6.9 – Séquence de mouvement illustrant la tâche de saisie de la cassette grise en présence d'un obstacle

6.5 Modification de la loi d'évolution de la trajectoire en présence de l'Homme

Cette expérimentation présente la modification de la loi de mouvement pendant l'exécution pour prendre en compte les contraintes de sécurité et de confort de l'Homme. Le champ de vue de l'Homme est calculé à partir de l'orientation du casque qui est localisé par le système de capture de mouvement. Parallèlement, le Kinect effectue l'acquisition de la posture de l'Homme. A partir de ces informations, le module SPARK calcule la visibilité de l'organe terminal du robot et la distance minimale Homme/robot (figure 6.10).

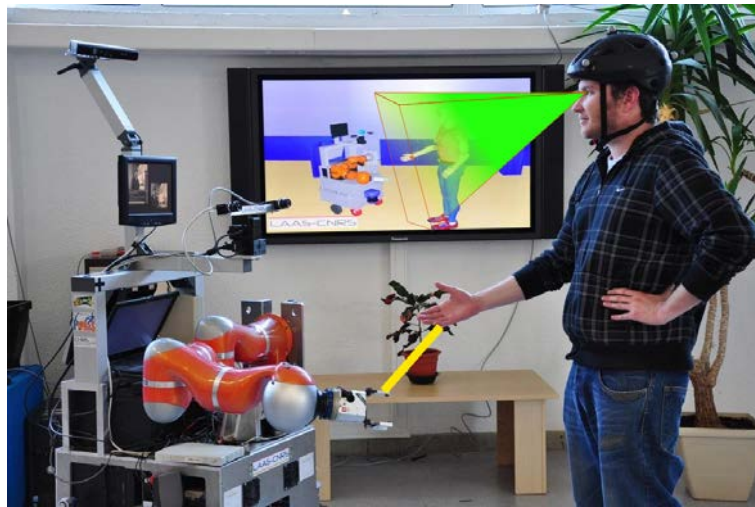


FIGURE 6.10 – Champ de vue de l'Homme et distance minimale Homme/robot

Les figures 6.11, 6.12 et 6.13 représentent trois exécutions d'une même trajectoire en présence d'un humain. La première figure représente la trajectoire initiale non adaptée de durée 3.2s. La figure 6.11(d) représente l'évolution des vitesses des axes du bras manipulateur durant l'exécution de la trajectoire.

La seconde figure (6.12) illustre l'influence du regard de l'Homme ; lorsque l'organe terminal du robot sort du champ de vue de l'humain, le robot s'arrête entre les instants 2.8s et 4.3s. La figure 6.13 illustre l'influence de la distance minimale entre le robot et l'Homme. Dans les figures 6.12(d) et 6.13(d), la courbe en pointillé représente la moyenne pondérée du coût de visibilité et du coût de distance. La vitesse du mouvement varie en fonction de l'inverse du coût jusqu'à s'arrêter à partir d'un certain seuil (10cm). Pendant l'enregistrement, l'humain a approché puis éloigné sa main plusieurs fois. Nous pouvons remarquer que les vitesses de chaque axe varient de manière synchronisée.

Le robot traite également les efforts extérieurs pour s'arrêter en cas de contact. Cette dernière fonction assure également un deuxième niveau de protection après le détecteur d'auto-collision en ligne COLDMAN pour limiter l'impact des collisions avec l'environnement.

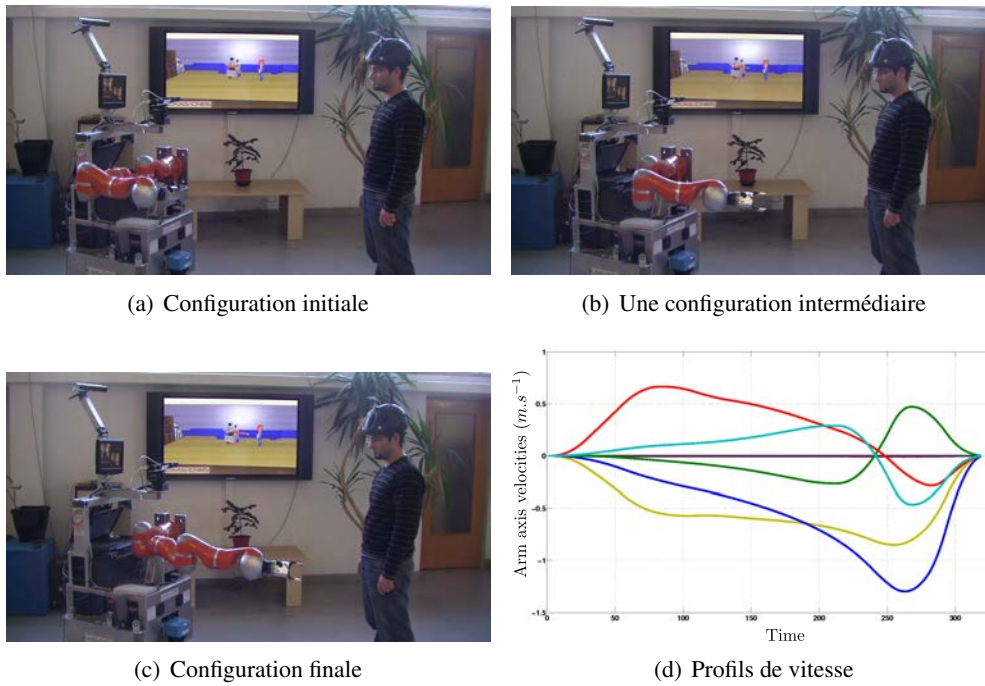


FIGURE 6.11 – Jido exécute une trajectoire en deux configurations dans l'espace cartésien sans prendre en compte l'humain

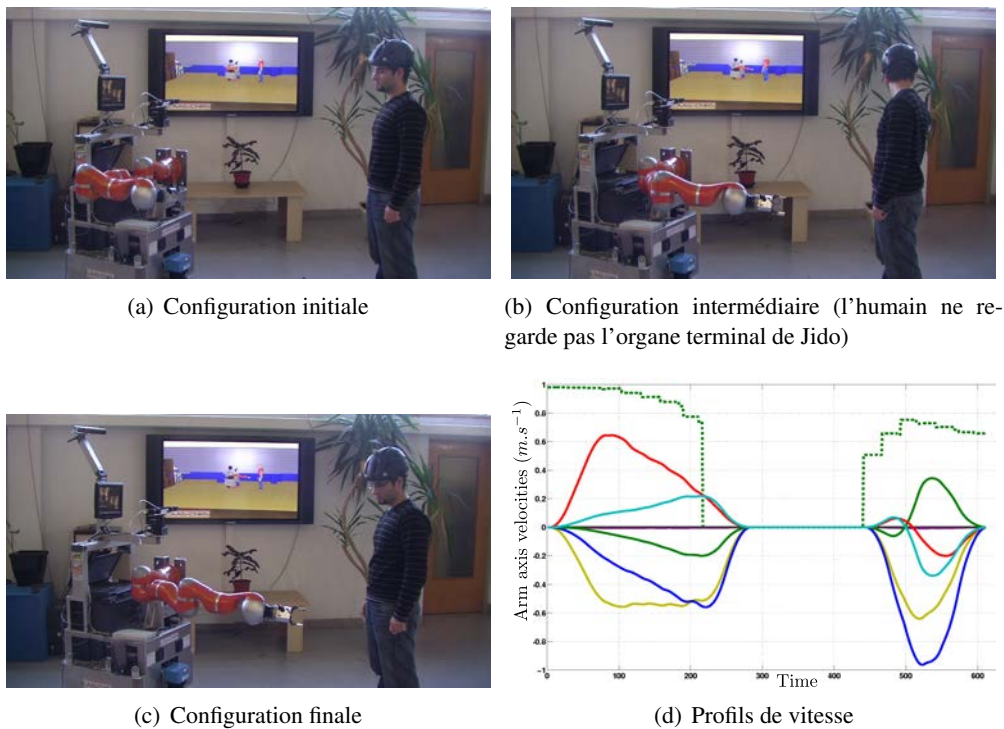


FIGURE 6.12 – Influence du champ de vue de l'humain sur la loi de mouvement

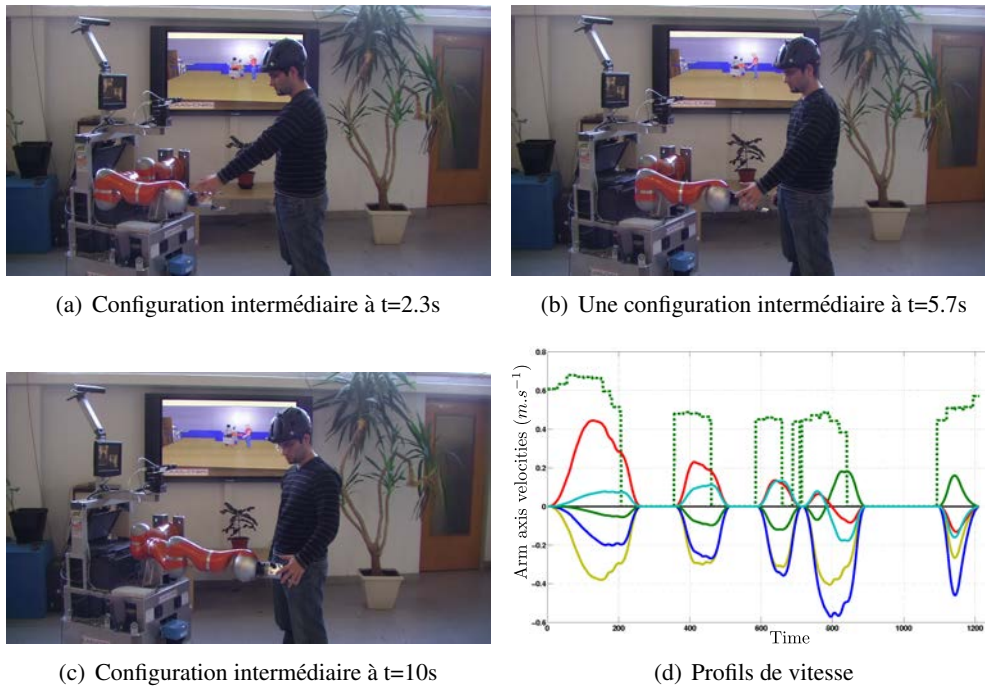


FIGURE 6.13 – Influence de la distance humain/robot sur la loi de mouvement

6.6 Conclusion

Dans ce chapitre nous avons d'abord détaillé l'architecture logicielle de Jido ainsi que les composants que nous avons développés. Nous avons présenté ensuite quatre expérimentations validant l'intégration de l'ensemble des travaux présentés dans ce manuscrit. L'intégration initiale du planificateur de trajectoire sur la plateforme Jido équipée du bras PA10 a été validée dans le cadre du projet européen COGNIRON³. Le planificateur est aujourd'hui utilisé quotidiennement et est intégré dans d'autres travaux notamment dans le cadre des projets européen DEXMART et CHRIS⁴. Il est (ou a été) aussi utilisé par plusieurs doctorants ou post-doctorant [Sisbot 08, Marin-Urias 09, Gharbi 10, Alili 11, Saut 10, Kumar Pandey 10, Mainprice 11]. L'adaptation de la loi de mouvement a également été validée par expérimentation pour le projet DEXMART en collaboration avec le laboratoire UNINA de Naples.

3. <http://www.cogniron.org/final/Home.php> accédé le 28/04/2011

4. <http://www.chrisfp7.eu> accédé le 28/04/2011

7

Conclusion et perspectives

Les travaux présentés dans ce manuscrit portent sur la génération de trajectoire pour la réalisation de tâches de manipulation d'objets et l'interaction Homme-robot. Le générateur proposé permet de réaliser un lien entre le planificateur de chemin et la commande du robot. Les méthodes proposées permettent de prendre en compte les contraintes liées à la présence de l'Homme, c'est à dire sa sécurité et son confort. La sécurité est traduite par la limitation de la vitesse et le confort par la limitation de l'accélération et du jerk.

Le générateur que nous avons développé est intégré au niveau du planificateur de chemin pour garantir l'absence de collision avec l'environnement lors du lissage de la trajectoire. Il permet de générer des trajectoires multidimensionnelles pour des robots holonomes. Nous avons présenté des applications de la génération de trajectoire dans l'espace articulaire et dans l'espace cartésien pour un robot manipulateur à sept degrés de liberté. De part sa généralité, le générateur de trajectoire peut être utilisé pour des systèmes complexes de grande dimension comme les robots équipés de deux bras et de deux mains ; la redondance du système est prise en compte au niveau du planificateur de chemin.

Un robot évoluant en présence de l'Homme doit modifier en ligne ses mouvements pour s'adapter aux mouvements des humains tout en respectant les contraintes cinématiques de limitation de la vitesse, de l'accélération et du jerk. Cependant les planificateurs de trajectoire actuels [Herrera-Aguilar 07, Haschke 08, Kröger 10b] sont limités au calcul des mouvements entre deux états avec conditions initiales quelconques en vitesse et accélération et des conditions finales en vitesse quelconques, mais nulle en accélération (mouvement de type IV [Kröger 10b]). Ils ne permettent donc pas de respecter les contraintes demandées. Ainsi, la première contribution de nos travaux est le développement d'un générateur de trajectoire limitant la vitesse, l'accélération et le jerk pour des mouvements avec des accélérations quelconques pour les états finaux. Nous avons validé l'utilisation en ligne de ce planificateur à travers des expérimentations et montré qu'il permettait de modifier la loi de mouvement de la

trajectoire multidimensionnelle du robot pendant l'exécution.

La deuxième contribution porte sur un algorithme d'approximation de trajectoire sous forme de suites de segments de courbes polynomiales cubiques. Nous avons montré que grouper ces segments par trois permet l'approximation efficace et rapide d'une trajectoire. Cette méthode est plus simple que la méthode classique d'approximation par une B-spline cubique et permet de fixer une erreur maximale d'approximation.

Nous avons proposé une application originale de cette méthode d'approximation où les trajectoires articulaires réalisées par le robot sont des approximations des trajectoires définies dans l'espace cartésien.

Enfin, l'ensemble des travaux présentés dans ce manuscrit ont été intégré et validé sur la plateforme Jido du LAAS-CNRS. Bien que le robot Jido ne soit équipé que d'un seul bras, les composants logiciels ont été développés pour prendre en compte deux bras. Ainsi, le déploiement de l'architecture planification/contrôle de trajectoire sur la future plateforme à deux bras que le LAAS va recevoir sera immédiat. Le planificateur mis en place est utilisé quotidiennement et constitue la base d'autres travaux de recherche.

Améliorations et extensions

Dans cette partie, nous proposons d'abord des améliorations et perspectives spécifiques à chaque méthode présentée dans ce manuscrit. Nous proposerons ensuite des perspectives plus générales de recherche.

Générateur de trajectoire de type V et plus

- nous avons proposé une décomposition de l'algorithme de génération de trajectoire élémentaire en 32 cas différents qui s'appuient sur la division du plan accélération-vitesse en sept zones. Il peut exister une meilleure décomposition permettant de réduire le nombre de cas et donc d'envisager de travailler à la réduction de la complexité de l'algorithme ;
- pour établir l'algorithme, nous avons représenté les mouvements dans le plan accélération-vitesse. Ceci nous a permis d'aborder les calculs sur une base géométrique d'intersection de paraboles et de s'affranchir de la représentation en fonction du temps. Pour développer un algorithme de type IX qui limite la dérivé du jerk (le snap), il n'existe pas de plan pour représenter tout le mouvement et simplifier les calculs. Prenons, par exemple, le cas du mouvement point à point, il est composé de 15 segments de polynôme de degrés quatre que nous pouvons représenter dans l'espace jerk-accélération-vitesse (figure 7.1). Sa projection dans le plan accélération-vitesse (figure 7.2) montre la disparition des points anguleux. Même si sur le plan pratique la limitation du snap ne semble pas indispensable, quelle méthode faudrait-il mettre en œuvre pour mettre au point un tel algorithme ?

Algorithme d'approximation de trajectoire

- pour approximer une trajectoire, nous avons divisé la trajectoire en intervalles reliant deux points de la trajectoire initiale. À chaque borne de ces intervalles, l'erreur de trajectoire est nulle. Lorsque l'erreur sur l'intervalle est supérieure à l'erreur maximale désirée, l'intervalle est divisé en deux

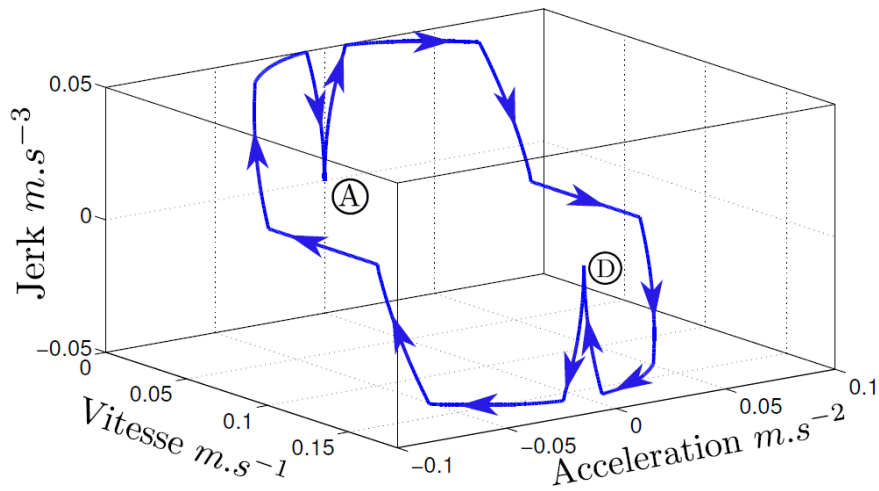


FIGURE 7.1 – Mouvement point à point à snap limité représenté dans l'espace jerk-accélération-vitesse (par analogie au plan accélération-vitesse, le point A correspond à l'arrêt et le point D au segment à vitesse constante)

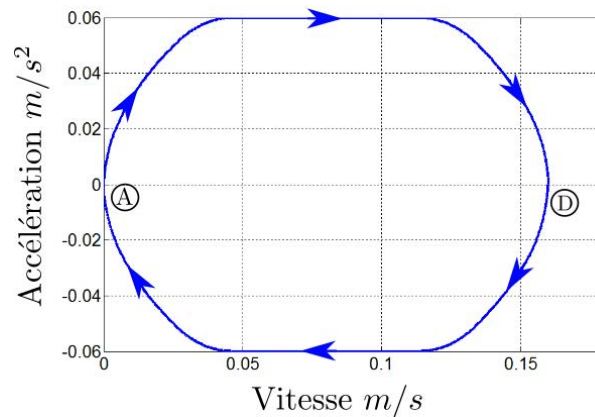


FIGURE 7.2 – Mouvement point à point à snap limité représenté dans l'espace accélération-vitesse

sous intervalles. Lors de cette division nous avons pris un point appartenant à la trajectoire initiale. Une amélioration serait de choisir un point au voisinage de la trajectoire afin de réduire les erreurs des deux nouvelles sous trajectoires comme illustré sur la figure 7.3 ;

- l'algorithme d'approximation suppose que les trajectoires initiales respectent des limites en jerk, accélération et vitesse. Pour l'illustrer nous avons construit des trajectoires avec des lois d'évolution en double S dépendant uniquement de la longueur de la courbe et des paramètres curviligne de vitesse, accélération et jerk imposés. Une extension possible serait de définir la loi de mouvement en prenant en compte la courbure du chemin. Des trajectoires bornées en jerk, accélération et vitesse seraient alors obtenues directement notamment pour des chemins comportant des points anguleux ou des points de rebroussement.
- afin d'éviter tout calcul itératif lors de l'approximation d'une trajectoire, il serait intéressant de connaître la longueur maximale d'une portion de la trajectoire permettant de ne pas dépasser une erreur donnée. Plus précisément, ceci revient à majorer l'erreur d'approximation en fonction des

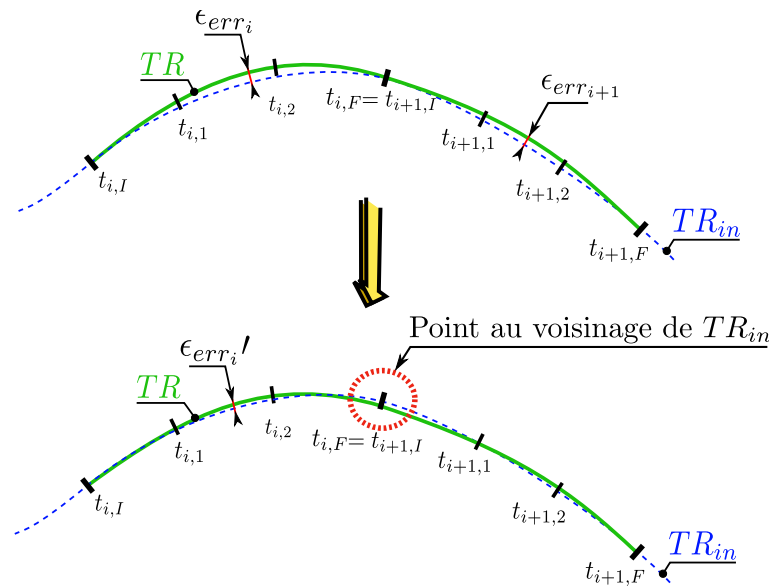


FIGURE 7.3 – Choix d’un point au voisinage de la trajectoire initiale pour améliorer l’approximation et obtenir $\epsilon_{err_i}' \leq \max\{\epsilon_{err_i}, \epsilon_{err_{i+1}}\}$

caractéristiques de jerk, d’accélération et de vitesse de la portion de trajectoire considérée.

Générateur de trajectoire appliqué aux robots manipulateurs

- les contraintes cinématiques que nous avons choisies pourraient être adaptées aux contraintes dynamiques du système. Par exemple, les paramètres d’accélération d’un robot varient avec la charge et la position du bras. Il nous paraît envisageable de considérer des limites variables par morceaux.
- l’utilisation directe du planificateur local temporel par le planificateur de chemin (trajectoire) permettrait de choisir le chemin le plus court en temps et non plus en distance.
- nous avons pris en compte la position de l’Homme pendant l’exécution pour adapter la vitesse. Serait-il utile d’introduire une carte de coût représentant les vitesses admissibles autour de l’Homme dans l’étape de la planification de trajectoire ?
- l’introduction du paramètre temps au travers des trajectoires doit permettre de prendre en compte la mobilité des éléments de l’environnement lorsque, comme en milieu industriel, ils sont connus à l’avance. Des mouvements synchronisés ou coordonnés sont envisageable dans ce cas.

Perspectives générales

Ce travail a montré l’intérêt de l’utilisation des trajectoires en robotique aussi bien au niveau de la planification que du contrôle des mouvements. Nous avons aussi vu que pour la surveillance des collisions avec les humains, le raisonnement au niveau des trajectoires permet d’anticiper les événements. La prise en compte explicite du temps au moyen des trajectoires donne la possibilité de réaliser des mouvements coordonnés entre des éléments d’un même robot ou entre plusieurs robots.

Bien sûr, l’étude de la dynamique des robots est indispensable, par exemple, pour permettre la marche. Mais compte tenu des temps de calcul des simulateurs dynamiques, il est inenvisageable pour

longtemps encore de les utiliser pour planifier des mouvements pour l'interaction Homme-robot. La notion de trajectoire bornée en jerk, accélération et vitesse permet de générer des trajectoires qui seront probablement dynamiquement acceptables et réalisables par le robot.

La planification est souvent décomposée en deux parties : la planification des tâches et la planification des déplacements. Ici encore, la notion de trajectoire qui donne à la fois des informations temporelles sur la durée d'une action et géométrique sur les lieux occupés à un instant donné, peut être utilisée pour raisonner et échanger les informations.

Enfin les robots deviennent de plus en plus complexes, une plate-forme mobile équipée de deux bras et deux mains, soit plus de quarante degrés de liberté, est courant dans les laboratoires. A l'avenir ces robots devront coopérer entre eux et avec d'autres systèmes pour réaliser un ensemble de tâches. Ces systèmes ne peuvent pas être contrôlés par une seule unité de calcul, ce qui nécessite que les différents calculateurs communiquent des informations spatio-temporelles. Les trajectoires constituées de séries de fonctions cubiques que nous avons étudiés dans ce mémoire pourraient être utilisées comme support pour ces échanges d'information.



Calcul de l'état d'une trajectoire à un instant donné

Cette partie détaille le calcul de l'état d'une trajectoire mono-dimensionnelle ; c'est à dire de l'accélération $A_k(t)$, de la vitesse $V_k(t)$ et de la position $X_k(t)$ à un instant t avec $\sum_{i=1}^{k-1} T_i < t \leq \sum_{i=1}^k T_i + T_k$ sur une trajectoire composée d'une suite de N segments de courbes polynomiales cubiques (fig. A.1). Nous considérons l'instant initial t_I nul ($t_I = 0$). Les conditions initiales de la trajectoire sont A_I pour l'accélération, V_I pour la vitesse et X_I pour la position.

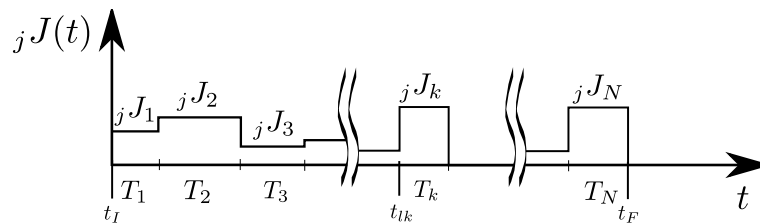


FIGURE A.1 – Profil du jerk d'une suite de segments de courbes polynomiales cubiques associé à une trajectoire ${}_j\text{TR}(t)$ (j est l'axe considéré)

Calcul de l'accélération

L'accélération à un instant t appartenant au premier segment ($t \in [0, T_1]$) est :

$$\begin{aligned}
 A_1(t) &= \int_0^t J_1 dt + A_I \\
 A_1(t) &= J_1 t + A_I
 \end{aligned}
 \tag{A.1}$$

De même pour les segments 2 ($t \in [T_1, T_1 + T_2]$) et 3 ($t \in [T_1 + T_2, T_1 + T_2 + T_3]$) :

$$\begin{aligned} A_2(t) &= \int_{T_1}^t J_2 dt + J_1 T_1 + A_I \\ A_2(t) &= J_2(t - T_1) + J_1 T_1 + A_I \end{aligned} \quad (\text{A.2})$$

$$A_3(t) = J_3(t - T_1 - T_2) + J_1 T_1 + J_2 T_2 + A_I \quad (\text{A.3})$$

En généralisant, on obtient :

$$\begin{aligned} A_k(t) &= \int_{\sum_{i=1}^{k-1} T_i}^t J_k dt + \sum_{i=1}^{k-1} J_i T_i + A_I \\ A_k(t) &= J_k(t - \sum_{i=1}^{k-1} T_i) + \sum_{i=1}^{k-1} J_i T_i + A_I \end{aligned} \quad (\text{A.4})$$

Démonstration par récurrence de l'équation A.4 :

L'équation est vérifiée pour $k = 1$ (eq. A.1). Nous supposons que l'équation A.4 est vraie pour k .

Calcul pour $k + 1$:

$$\begin{aligned} A_{k+1}(t) &= \int_{\sum_{i=1}^k T_i}^t J_{k+1} dt + A_k(\sum_{i=1}^k T_i) \\ A_{k+1}(t) &= J_{k+1}(t - \sum_{i=1}^k T_i) + J_k(\sum_{i=1}^k T_i - \sum_{i=1}^{k-1} T_i) + \sum_{i=1}^{k-1} J_i T_i + A_I \\ A_{k+1}(t) &= J_{k+1}(t - \sum_{i=1}^{(k+1)-1} T_i) + \sum_{i=1}^{(k+1)-1} J_i T_i + A_I \end{aligned} \quad (\text{A.5})$$

Donc l'équation A.4 est vérifiée.

Calcul de la vitesse

La vitesse à un instant t appartenant au premier segment est :

$$\begin{aligned} V_1(t) &= \int_0^t A_1(t) dt + V_I \\ V_1(t) &= \int_0^t \int_0^t J_1 dt dt + \int_0^t A_I(t) dt + V_I \\ V_1(t) &= \frac{J_1}{2} t^2 + A_I t + V_I \end{aligned} \quad (\text{A.6})$$

De même pour les segments 2 et 3 :

$$\begin{aligned}
 V_2(t) &= \int_{T_1}^t A_2(t) dt + V_1(T_1) \\
 &= \int_{T_1}^t \left(\int_{T_1}^t J_2 dt + J_1 T_1 + A_I \right) dt + \frac{J_1 T_1^2}{2} + A_I T_1 + V_I \\
 &= \frac{J_2}{2} (t - T_1)^2 + J_1 T_1 (t - T_1) + \frac{J_1 T_1^2}{2} + A_I t + V_I
 \end{aligned} \tag{A.7}$$

$$\begin{aligned}
 V_3(t) &= \int_{T_1+T_2}^t A_3(t) dt + V_2(T_1 + T_2) \\
 &= \frac{J_3}{2} (t - (T_1 + T_2))^2 + J_1 T_1 (t - T_1) + J_2 T_2 (t - (T_1 + T_2)) \\
 &\quad + \frac{J_2 T_2^2}{2} + \frac{J_1 T_1^2}{2} + A_I t + V_I
 \end{aligned} \tag{A.8}$$

En généralisant, on obtient :

$$V_k(t) = \frac{J_k}{2} \left(t - \sum_{i=1}^{k-1} T_i \right)^2 + \sum_{i=1}^{k-1} J_i T_i \left(t - \sum_{j=1}^i T_j \right) + \sum_{i=1}^{k-1} \frac{J_i T_i^2}{2} + A_I t + V_I \tag{A.9}$$

Démonstration par récurrence de l'équation A.9 :

L'équation est vérifiée pour $k = 1$ (eq. A.12). Nous supposons que l'équation A.9 est vraie pour k .

Calcul pour $k + 1$:

$$\begin{aligned}
 V_{k+1}(t) &= \frac{J_{k+1}}{2} \left(t - \sum_{i=1}^k T_i \right)^2 + A_k \left(\sum_{i=1}^k T_i \right) \left(t - \sum_{i=1}^k T_i \right) + V_k \left(\sum_{i=1}^k T_i \right) \\
 V_{k+1}(t) &= \frac{J_{k+1}}{2} \left(t - \sum_{i=1}^k T_i \right)^2 + \left(J_k \left(\sum_{i=1}^k T_i - \sum_{i=1}^{k-1} T_i \right) + \sum_{i=1}^{k-1} J_i T_i + A_I \right) \left(t - \sum_{i=1}^k T_i \right) \\
 &\quad + \frac{J_k}{2} \left(\sum_{i=1}^k T_i - \sum_{i=1}^{k-1} T_i \right)^2 + \sum_{i=1}^{k-1} J_i T_i \left(\sum_{i=1}^k T_i - \sum_{j=1}^i T_j \right) + \sum_{i=1}^{k-1} \frac{J_i T_i^2}{2} \\
 &\quad + A_I \sum_{i=1}^k T_i + V_I
 \end{aligned} \tag{A.10}$$

En développant et après simplification on obtient :

$$V_{k+1}(t) = \frac{J_{k+1}}{2} \left(t - \sum_{i=1}^{(k+1)-1} T_i \right)^2 + \sum_{i=1}^{(k+1)-1} J_i T_i \left(t - \sum_{j=1}^i T_j \right) + \sum_{i=1}^{(k+1)-1} \frac{J_i T_i^2}{2} + A_I t + V_I \tag{A.11}$$

Donc l'équation A.9 est vérifiée.

Calcul de la position

La vitesse à un instant t appartenant au premier segment est :

$$\begin{aligned} X_1(t) &= \int_0^t V_1(t) dt + X_I \\ &= \frac{J_1}{6}t^3 + \frac{A_I}{2}t^2 + V_I t + X_I \end{aligned} \quad (\text{A.12})$$

De même pour les segments 2 et 3 :

$$\begin{aligned} X_2(t) &= \int_{T_1}^t V_2(t) dt + X_1(T_1) \\ &= \frac{J_2}{6}(t - T_1)^3 + \frac{J_1 T_1}{2}(t - T_1)^2 + \frac{J_1 T_1^2}{2}(t - T_1) + \frac{J_1 T_1^3}{6} + \frac{A_I}{2}t^2 + V_I t + X_I \\ X_3(t) &= \int_{T_1+T_2}^t V_3(t) dt + X_2(T_1 + T_2) \\ &= \frac{J_3}{6}(t - (T_1 + T_2))^3 + \frac{J_1 T_1}{2}(t - T_1)^2 + \frac{J_2 T_2}{2}(t - (T_1 + T_2))^2 \\ &\quad + \frac{J_1 T_1^2}{2}(t - T_1) + \frac{J_2 T_2^2}{2}(t - (T_1 + T_2)) + \frac{J_1 T_1^3}{6} + \frac{J_2 T_2^3}{6} + \frac{A_I}{2}t^2 + V_I t + X_I \end{aligned} \quad (\text{A.13})$$

En généralisant, on obtient :

$$\begin{aligned} X_k(t) &= \frac{J_k}{6} \left(t - \sum_{i=1}^{k-1} T_i \right)^3 + \sum_{i=1}^{k-1} \frac{J_i T_i}{2} \left(t - \sum_{j=1}^i T_j \right)^2 + \sum_{i=1}^{k-1} \frac{J_i T_i^2}{2} \left(\sum_{j=1}^i T_j \right) + \sum_{i=1}^{k-1} \frac{J_i T_i^3}{6} \\ &\quad + \frac{A_I}{2}t^2 + V_I t + X_I \end{aligned} \quad (\text{A.14})$$

Comme pour les équations A.4 et A.9, l'équation A.14 se démontre par récurrence.

B

Calcul de la distance critique

Dans cette annexe, nous présentons les différents cas du calcul de la distance critique. Nous noterons A_i l'accélération, V_i la vitesse et X_i le position à l'instant T_i avec $i \in [0..7]$.

Cas de la zone 1

Condition : $V_{fmm} \geq V_{smp}$ Suivant le positionnement des points P_1 et P_2 on obtient deux types de mouvements critiques. Cette différence provient du croisement ou non de la parabole à Jerk négatif du point initial avec celle à jerk positif du point final. Ainsi, en plus des équations des mouvements élémentaires, pour pouvoir résoudre le système d'équations, il faut calculer le point d'intersection des deux paraboles en utilisant l'équation des courbes à jerk minimal et maximal (eq. 3.14) définie précédemment. Nous noterons ce point (A_1, V_1) puisqu'il correspond à l'instant T_1 .

– Premier type (Fig. B.1(a)) $V_{s0p} < V_{f0m}$

Le mouvement atteint l'accélération maximale et est $P_1 - T_1 - T_2 - P_2$

Equations :

$$J_1 = J_{max}$$

$$A_1 = A_I + J_{max} \cdot T_{jpa}$$

$$V_1 = V_I + A_I \cdot T_{jpa} + \frac{1}{2} \cdot J_{max} \cdot T_{jpa}^2$$

$$X_1 = X_I + V_I \cdot T_{jpa} + \frac{1}{2} \cdot A_I \cdot T_{jpa}^2 + \frac{1}{6} \cdot J_{max} \cdot T_{jpa}^3$$

$$A_2 = A_1$$

$$V_2 = V_1 + A_1 \cdot T_{aca}$$

$$X_2 = X_1 + V_1 \cdot T_{aca} + \frac{1}{2} \cdot A_1 \cdot T_{aca}^2$$

$$A_F = A_2 - J_{max} \cdot T_{jna}$$

$$V_F = V_2 + A_2 \cdot T_{jna} - \frac{1}{2} \cdot J_{max} \cdot T_{jna}^2$$

$$dc = X_2 + V_2 \cdot T_{jna} + \frac{1}{2} \cdot A_2 \cdot T_{jna}^2 - \frac{1}{6} \cdot J_{max} \cdot T_{jna}^3 - X_I$$

Solution :

$$A_2 = A_1 = A_{max}$$

$$V_1 = V_{smp}$$

$$V_2 = V_{fmm}$$

$$T_{jpa} = \frac{A_{max} - A_I}{J_{max}}$$

$$T_{aca} = \frac{V_{fmm} - V_{smp}}{A_{max}}$$

$$T_{jna} = \frac{A_{max} - A_F}{J_{max}}$$

$$X_1 = X_I + V_I \cdot T_{jpa} + \frac{1}{2} \cdot A_I \cdot T_{jpa}^2 + \frac{1}{6} \cdot J_{max} \cdot T_{jpa}^3$$

$$X_2 = X_1 + V_{smp} \cdot T_{aca} + \frac{1}{2} \cdot A_{max} \cdot T_{aca}^2$$

$$dc = X_2 + V_{fmm} \cdot T_{jna} + \frac{1}{2} \cdot A_{max} \cdot T_{jna}^2 - \frac{1}{6} \cdot J_{max} \cdot T_{jna}^3 - X_I$$

– Deuxième type (Fig. B.1(b)) $V_{s0p} \geq V_{f0m}$

Le mouvement est $P_1 - T_1 - P_2$

Equations :

$$V_1 = V_I - \frac{1}{2 \cdot J_{max}} \cdot (A_1^2 - A_I^2)$$

$$V_1 = V_F + \frac{1}{2 \cdot J_{max}} \cdot (A_1^2 - A_F^2)$$

$$J_1 = -J_{max}$$

$$A_1 = A_I - J_{max} \cdot T_{jna}$$

$$V_1 = V_I + A_I \cdot T_{jna} - \frac{1}{2} \cdot J_{max} \cdot T_{jna}^2$$

$$X_1 = X_I + V_I \cdot T_{jna} + \frac{1}{2} \cdot A_I \cdot T_{jna}^2 - \frac{1}{6} \cdot J_{max} \cdot T_{jna}^3$$

$$A_F = A_1 + J_{max} \cdot T_{jpa}$$

$$V_F = V_1 + A_1 \cdot T_{jpa} + \frac{1}{2} \cdot J_{max} \cdot T_{jpa}^2$$

$$dc = X_1 + V_1 \cdot T_{jpa} + \frac{1}{2} \cdot A_1 \cdot T_{jpa}^2 + \frac{1}{6} \cdot J_{max} \cdot T_{jpa}^3 - X_I$$

Solution :

$$A_1 = -\sqrt{J_{max} \cdot (V_I - V_F) + \frac{1}{2} \cdot (A_F^2 + A_I^2)}$$

$$V_1 = V_I - \frac{1}{2 \cdot J_{max}} \cdot (A_1^2 - A_I^2)$$

$$T_{jna} = \frac{A_I - A_1}{J_{max}}$$

$$T_{jpa} = \frac{A_F - A_1}{J_{max}}$$

$$X_1 = X_I + V_I \cdot T_{jna} + \frac{1}{2} \cdot A_I \cdot T_{jna}^2 - \frac{1}{6} \cdot J_{max} \cdot T_{jna}^3$$

$$dc = X_1 + V_1 \cdot T_{jpa} + \frac{1}{2} \cdot A_1 \cdot T_{jpa}^2 + \frac{1}{6} \cdot J_{max} \cdot T_{jpa}^3 - X_I$$

Cas de la zone 2

Condition : $V_{fmp} \leq V_{smp}$

Comme pour la zone 1, il y a deux types de mouvement.

– Premier type (Fig. B.2(a)) $V_{s0m} \geq V_{f0p}$

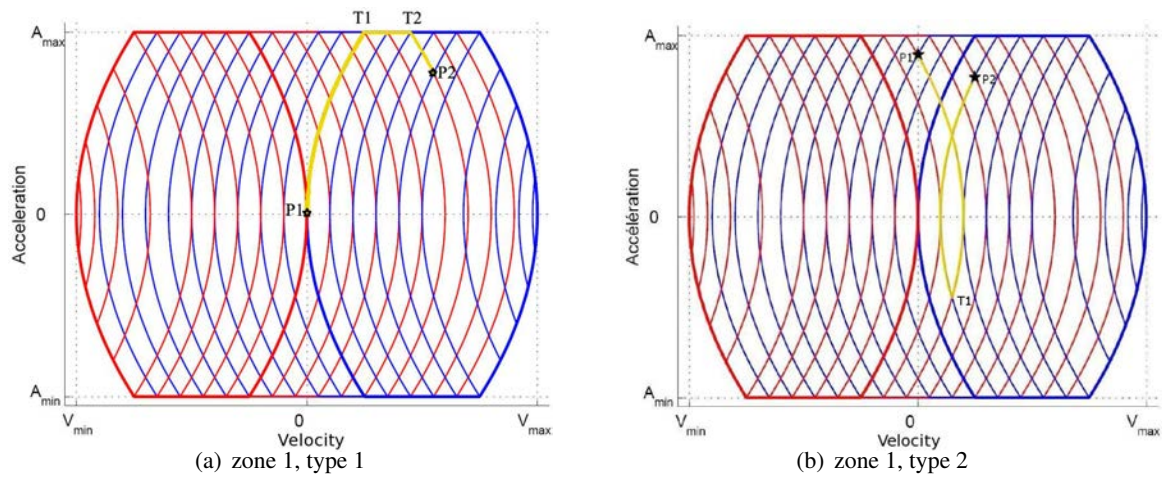


FIGURE B.1 – Distances critiques zone 1

Motion is $P_1 - T_1 - T_2 - P_2$

Equations :

$$J_1 = -J_{max}$$

$$A_1 = A_I - J_{max} \cdot T_{jna}$$

$$V_1 = V_I + A_I \cdot T_{jna} - \frac{1}{2} \cdot J_{max} \cdot T_{jna}^2$$

$$X_1 = X_I + V_I \cdot T_{jna} + \frac{1}{2} \cdot A_I \cdot T_{jna}^2 - \frac{1}{6} \cdot J_{max} \cdot T_{jna}^3$$

$$A_2 = A_1$$

$$V_2 = V_1 + A_1 \cdot T_{aca}$$

$$X_2 = X_1 + V_1 \cdot T_{aca} + \frac{1}{2} \cdot A_1 \cdot T_{aca}^2$$

$$A_F = A_2 + J_{max} \cdot T_{jpa}$$

$$V_F = V_2 + A_2 \cdot T_{jpa} + \frac{1}{2} \cdot J_{max} \cdot T_{jpa}^2$$

$$dc = X_2 + V_2 \cdot T_{jpa} + \frac{1}{2} \cdot A_2 \cdot T_{jpa}^2 + \frac{1}{6} \cdot J_{max} \cdot T_{jpa}^3 - X_I$$

Solution :

$$A_2 = A_1 = -A_{max}$$

$$V_1 = V_{s0m}$$

$$V_2 = V_{f0p}$$

$$T_{jna} = \frac{A_{max} + A_I}{J_{jpa}}$$

$$T_{aca} = -\frac{V_{f0p} - V_{s0m}}{A_{max}}$$

$$T_{jpa} = \frac{A_{max} + A_F}{J_{jpa}}$$

$$X_1 = X_I + V_I \cdot T_{jna} + \frac{1}{2} \cdot A_I \cdot T_{jna}^2 - \frac{1}{6} \cdot J_{max} \cdot T_{jna}^3$$

$$X_2 = X_1 + V_{s0m} \cdot T_{aca} - \frac{1}{2} \cdot A_{max} \cdot T_{aca}^2$$

$$dc = X_2 + V_{f0p} \cdot T_{jpa} - \frac{1}{2} \cdot A_{max} \cdot T_{jpa}^2 + \frac{1}{6} \cdot J_{max} \cdot T_{jpa}^3 - X_I$$

– Deuxième type (Fig. B.2(b)) ($V_{s0m} < V_{f0p}$)

Le mouvement est $P_1 - T_1 - P_2$

Equations :

$$V_1 = V_I + \frac{1}{2 \cdot J_{max}} \cdot (A_1^2 - A_I^2)$$

$$V_1 = V_F - \frac{1}{2 \cdot J_{max}} \cdot (A_1^2 - A_F^2)$$

$$J_1 = J_{max}$$

$$A_1 = A_I + J_{max} \cdot T_{jpa}$$

$$V_1 = V_I + A_I \cdot T_{jpa} + \frac{1}{2} \cdot J_{max} \cdot T_{jpa}^2$$

$$X_1 = X_I + V_I \cdot T_{jpa} + \frac{1}{2} \cdot A_I \cdot T_{jpa}^2 + \frac{1}{6} \cdot J_{max} \cdot T_{jpa}^3$$

$$A_F = A_1 - J_{max} \cdot T_{jna}$$

$$V_F = V_1 + A_1 \cdot T_{jna} - \frac{1}{2} \cdot J_{max} \cdot T_{jna}^2$$

$$dc = X_1 + V_1 \cdot T_{jna} + \frac{1}{2} \cdot A_1 \cdot T_{jna}^2 - \frac{1}{6} \cdot J_{max} \cdot T_{jna}^3 - X_I$$

Solution :

$$A_1 = \sqrt{J_{max} \cdot (V_F - V_I) + \frac{1}{2} \cdot (A_F^2 + A_I^2)}$$

$$V_1 = V_I + \frac{1}{2 \cdot J_{max}} \cdot (A_1^2 - A_I^2)$$

$$T_{jpa} = \frac{A_1 - A_I}{J_{max}}$$

$$T_{jna} = \frac{A_1 - A_F}{J_{max}}$$

$$X_1 = X_I + V_I \cdot T_{jpa} + \frac{1}{2} \cdot A_I \cdot T_{jpa}^2 + \frac{1}{6} \cdot J_{max} \cdot T_{jpa}^3$$

$$dc = X_1 + V_1 \cdot T_{jna} + \frac{1}{2} \cdot A_1 \cdot T_{jna}^2 - \frac{1}{6} \cdot J_{max} \cdot T_{jna}^3 - X_I$$

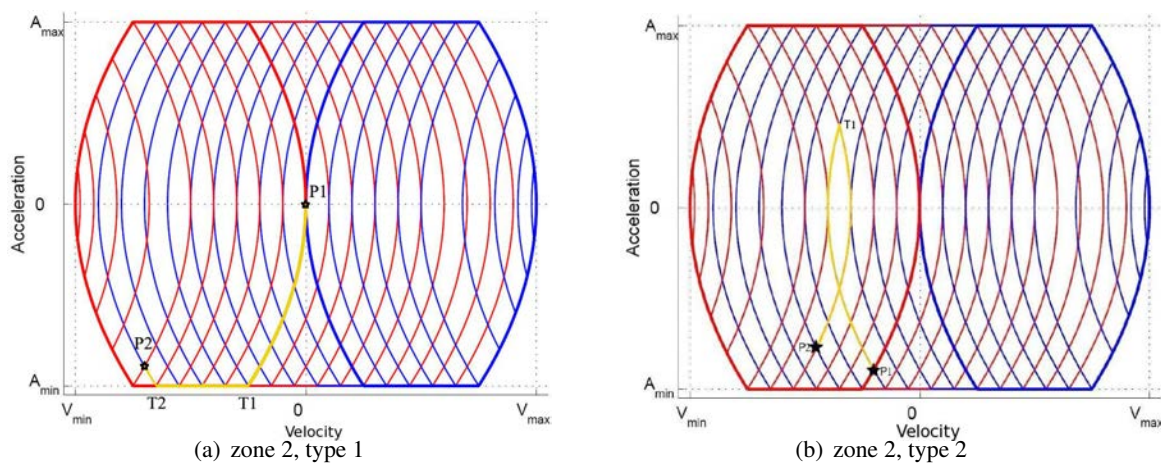


FIGURE B.2 – Distances critiques zone 2

Cas de la zone 3

Condition :

$$[(V_{fmp} > V_{smp}) \cap (V_{fmm} > V_{smm})] \cup [(A_F \leq 0) \cap (V_{fmm} > V_{smm}) \cap (V_{fmm} < V_{smp})] \cup [(V_{fmp} > V_{smp}) \cap (A_F \geq 0)]$$

Il y a deux types de mouvement.

- Premier type (Fig. B.3(a)) $((V_{s0p} > V_{f0m}) \cap (A_F > 0) \cap (A_I > 0))$

Le mouvement est $P_1 - T_1 - P_2$

Les équations sont les mêmes que dans la cas *zone 1, type 2*.

– Deuxième type (Fig. B.3(b)) $non((V_{s0p} > V_{s0m}) \cap (A_F > 0) \cap (A_I > 0))$

Le mouvement est $P_1 - T_1 - P_2$

Les équations sont les mêmes que pour le cas *zone 2, type 2*.

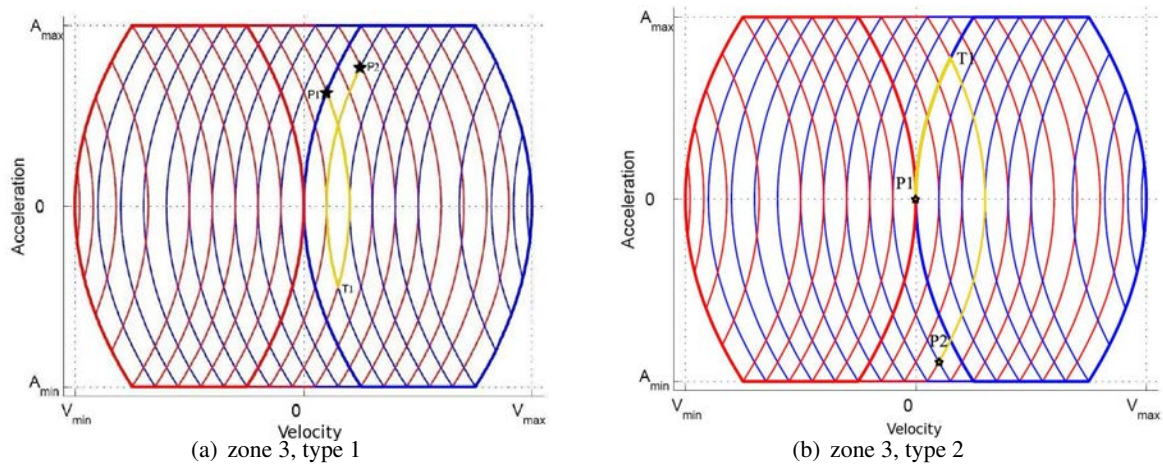


FIGURE B.3 – Distances critiques zone 3

Cas de la zone 4

Condition : $[(V_{fmm} \leq V_{smm}) \cap (V_{fmp} > V_{smm})] \cup [(A_F > 0) \cap (V_{fmm} > V_{smm}) \cap (V_{fmp} < V_{smp})]$

Il y a deux types de mouvement.

– Premier type (Fig. B.4(a)) $((V_{s0m} < V_{f0p}) \cap (A_F < 0) \cap (A_I < 0))$

Le mouvement est $P_1 - T_1 - P_2$

Les équations sont les mêmes que pour le cas *zone 2, type 2*.

– Deuxième type (Fig. B.4(b)) $non((V_{s0m} < V_{f0p}) \cap (A_F < 0) \cap (A_I < 0))$

Le mouvement est $P_1 - T_1 - P_2$

Les équations sont les mêmes que pour le cas *zone 1, type 2*.

Cas de la zone 5

Condition : $[(A_F > A_I) \cap (V_{s0m} == V_{f0m})]$

– Premier type (Fig. B.5(a)) $(A_F > 0) \cap (A_I \geq 0)$

Le mouvement est $P_1 - T_1 - P_2$

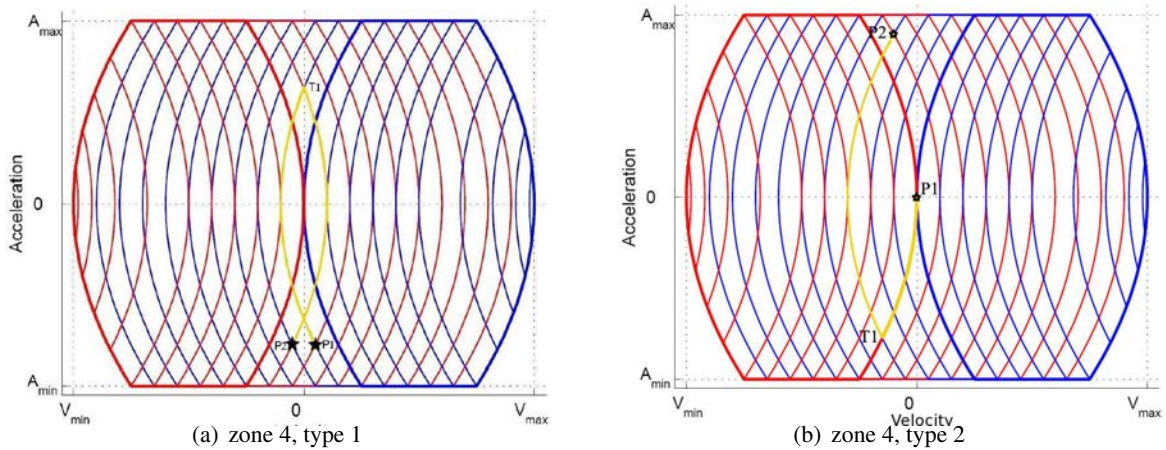


FIGURE B.4 – Distances critiques zone 4

Les équations sont les mêmes que pour le cas *zone 1, type 2*.

– Deuxième type (Fig. B.5(b)) ($A_F < 0$) \cap ($A_I \leq 0$)

Le mouvement est $P_1 - T_1 - P_2$

Les équations sont les mêmes que pour le cas *zone 2, type 2*.

– Troisième type (Fig. B.5(c)) ($A_F \geq 0$) \cap ($A_I \leq 0$)

Le mouvement est $P_1 - P_2$

Equations :

$$J_1 = J_{max}$$

$$A_F = A_I + J_{max} \cdot T_{jpa}$$

$$V_F = V_I + A_I \cdot T_{jpa} + \frac{1}{2} \cdot J_{max} \cdot T_{jpa}^2$$

$$dc = V_I \cdot T_{jpa} + \frac{1}{2} \cdot A_I \cdot T_{jpa}^2 + \frac{1}{6} \cdot J_{max} \cdot T_{jpa}^3$$

Solution :

$$T_{jpa} = \frac{(A_F - A_I)}{J_{max}}$$

$$dc = V_I \cdot T_{jpa} + \frac{1}{2} \cdot A_I \cdot T_{jpa}^2 + \frac{1}{6} \cdot J_{max} \cdot T_{jpa}^3$$

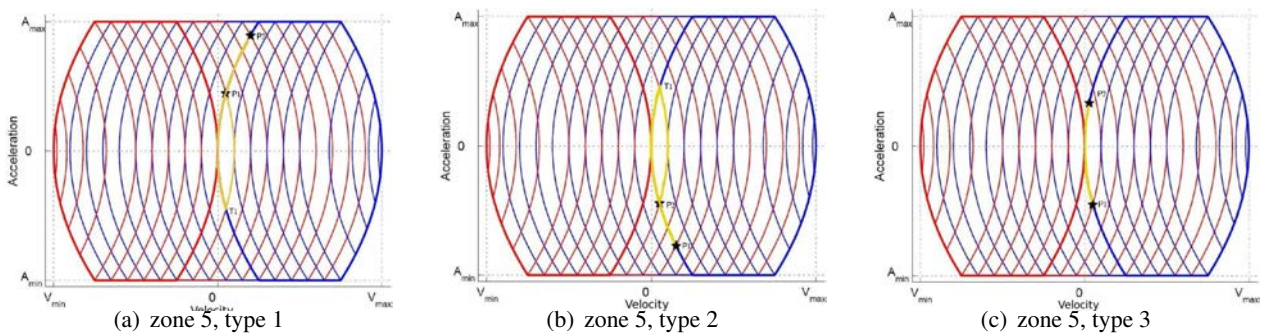


FIGURE B.5 – Distances critiques zone 5

Cas de la zone 6

Condition : $[(A_F < A_I) \cap (V_{s0p} == V_{f0p})]$

– Premier type (Fig. B.6(a)) $(A_F < 0) \cap (A_I \leq 0)$

Motion is $P_1 - T_1 - P_2$

Les équations sont les mêmes que pour le cas zone 2, type 2.

– Deuxième type (Fig. B.6(b)) $(A_F \geq 0) \cap (A_I > 0)$

Le mouvement est $P_1 - T_1 - P_2$

Les équations sont les mêmes que pour le cas zone 1, type 2.

– Troisième type (Fig. B.6(c)) $(A_F \leq 0) \cap (A_I \geq 0)$

Le mouvement est $P_1 - P_2$

Equations :

$$J_1 = -J_{max}$$

$$A_F = A_I - J_{max} \cdot T_{jna}$$

$$V_F = V_I + A_I \cdot T_{jna} - \frac{1}{2} \cdot J_{max} \cdot T_{jna}^2$$

$$dc = V_I \cdot T_{jna} + \frac{1}{2} \cdot A_I \cdot T_{jna}^2 - \frac{1}{6} \cdot J_{max} \cdot T_{jna}^3$$

Solution :

$$T_{jna} = \frac{(A_I - A_F)}{J_{max}}$$

$$dc = V_I \cdot T_{jna} + \frac{1}{2} \cdot A_I \cdot T_{jna}^2 - \frac{1}{6} \cdot J_{max} \cdot T_{jna}^3$$

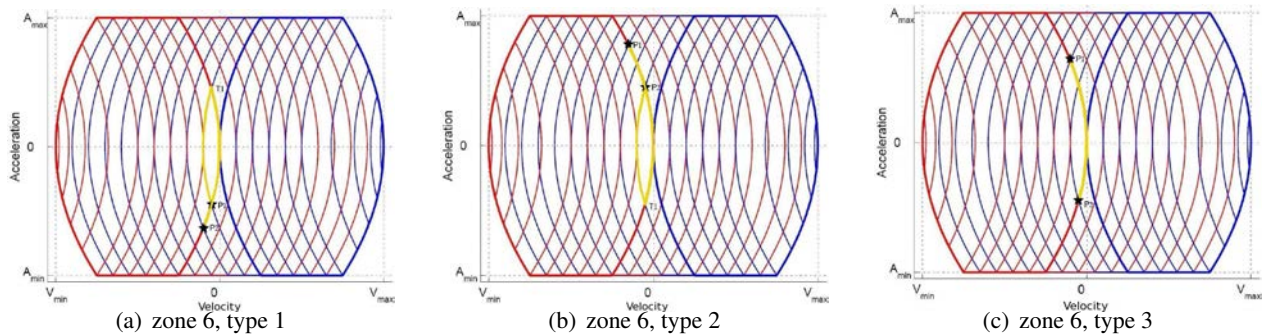
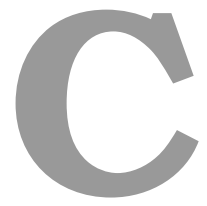


FIGURE B.6 – Distances critiques zone 6



Algorithmes de génération de trajectoires mono-dimensionnelles (suite)

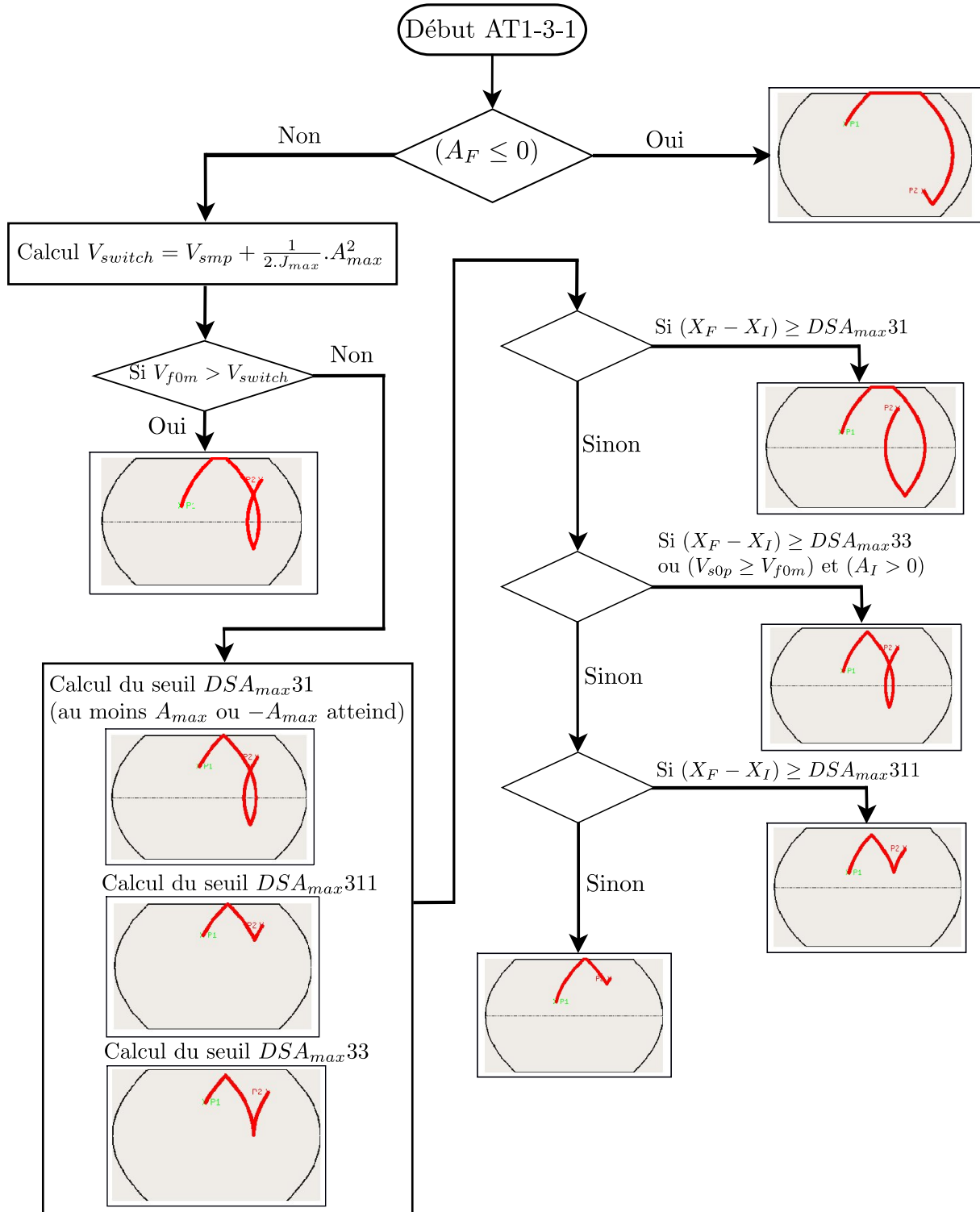


FIGURE C.1 – AT1-3-1 : Algorithme pour les trajectoires n’atteignant pas A_{max} et $-A_{max}$ de la zone 1

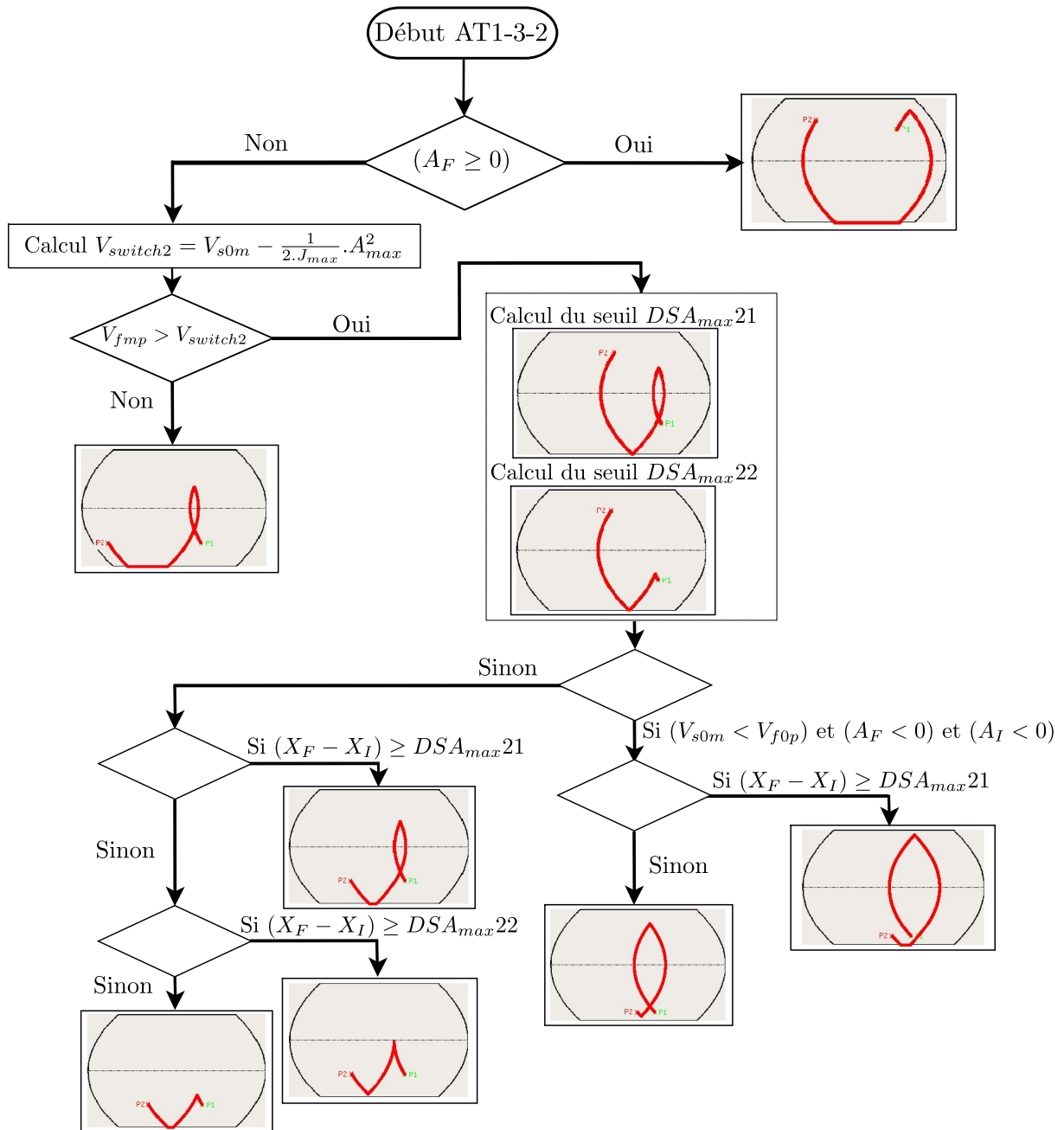


FIGURE C.2 – AT1-3-2 : Algorithme pour les trajectoires n’atteignant pas A_{max} et $-A_{max}$ de la zone 2

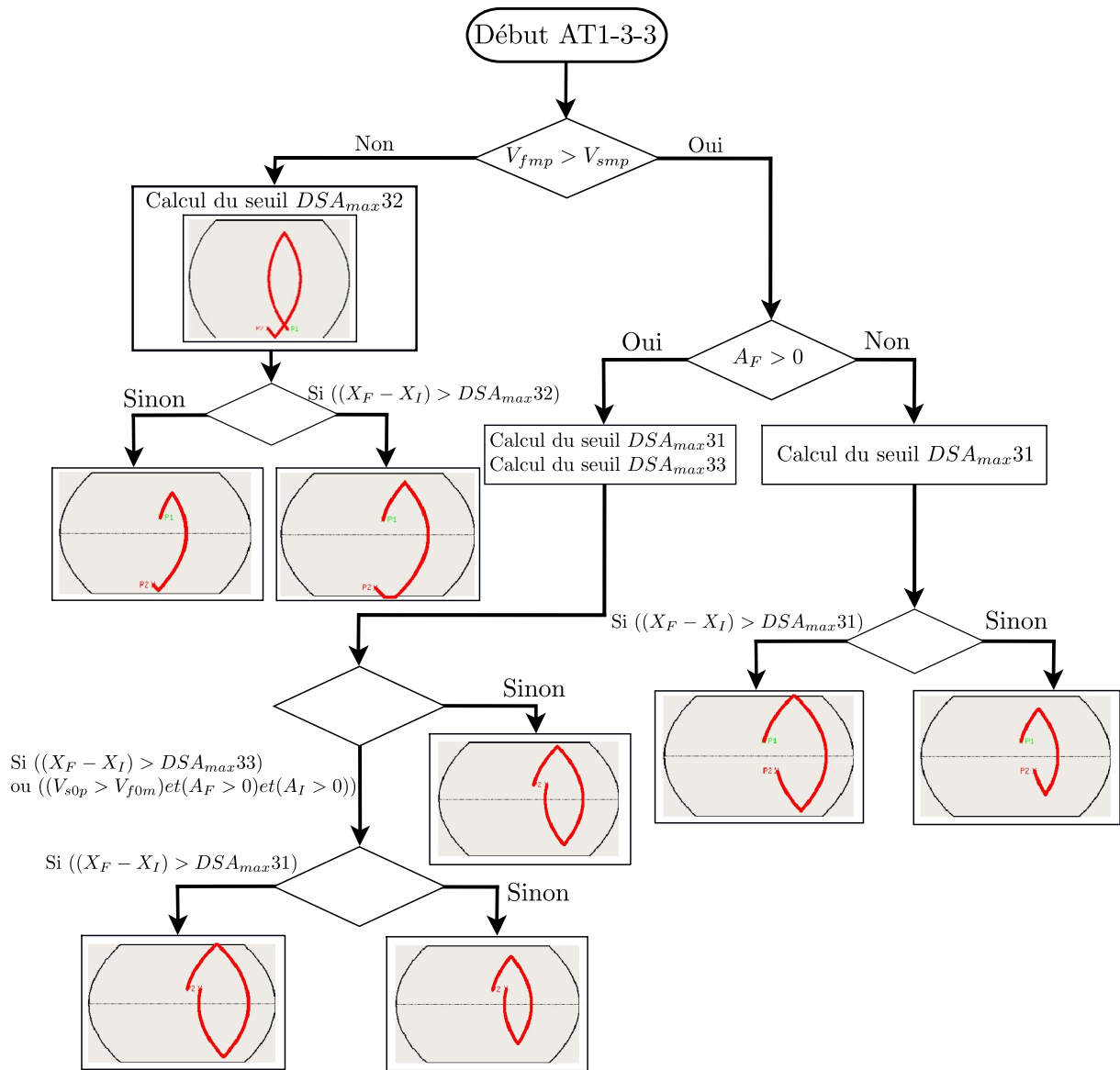
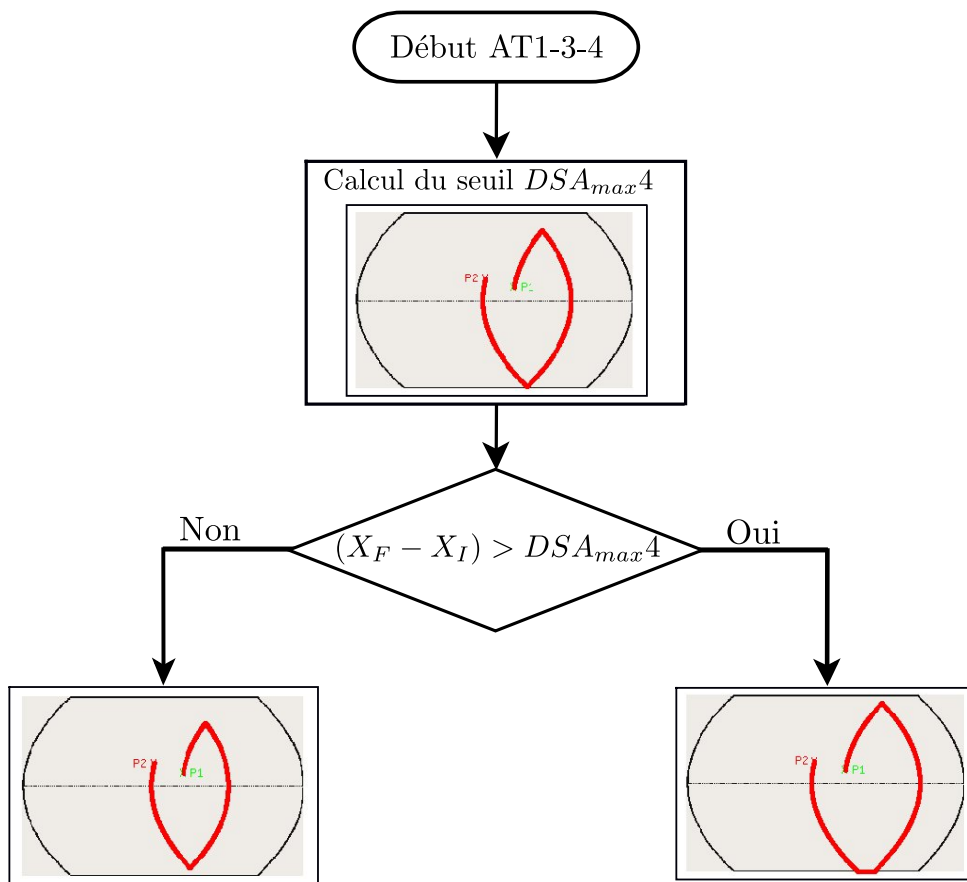
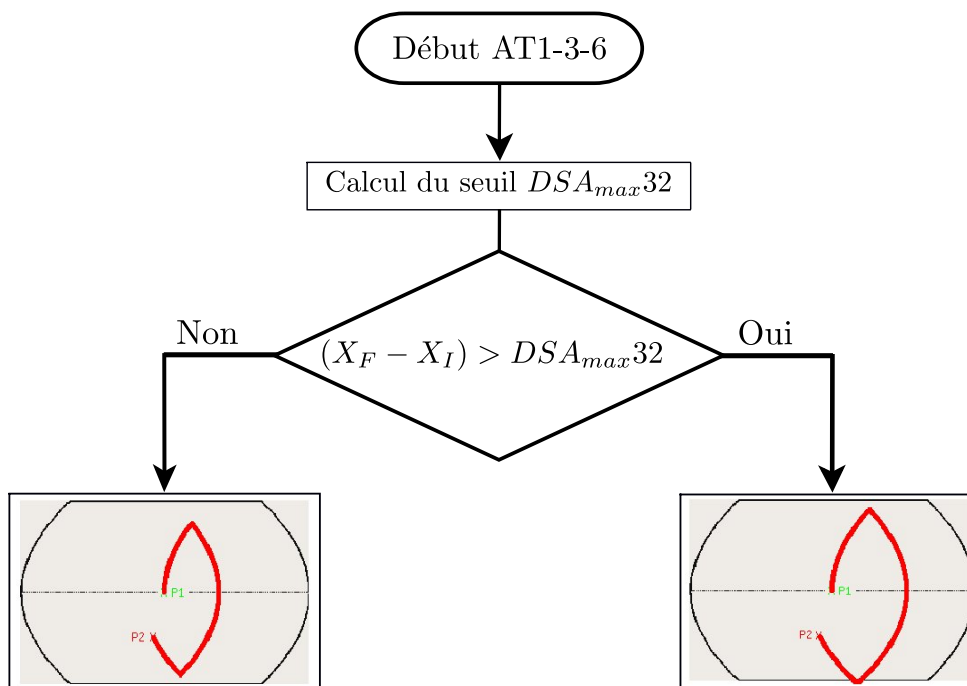


FIGURE C.3 – AT1-3-3 : Algorithme pour les trajectoires n’atteignant pas A_{max} et $-A_{max}$ de la zone 3

FIGURE C.4 – AT1-3-4 : Algorithme pour les trajectoires n'atteignant pas A_{max} et $-A_{max}$ de la zone 4FIGURE C.5 – AT1-3-6 : Algorithme pour les trajectoires n'atteignant pas A_{max} et $-A_{max}$ de la zone 6

D

Calcul de la durée d'un mouvement point à point

Cette annexe détaille les calculs de l'équation 3.23. Cette équation permet de calculer la durée d'un mouvement point à point atteignant l'accélération et la vitesse maximale.

Dans la section 3.2, nous avons obtenu les équations 3.16 :

$$\begin{aligned}T_j &= \frac{A_{max}}{J_{max}} \\T_a &= \frac{V_{max}}{A_{max}} - \frac{A_{max}}{J_{max}}\end{aligned}$$

La durée totale du mouvement étant (eq. 3.19) :

$$T = 4T_j + 2T_a + T_{vc}$$

Le système d'équation d'un mouvement point à point atteignant A_{max} et V_{max} s'écrit :

$$\begin{aligned}
 A_1 &= J_{max}T_j \\
 V_1 &= \frac{1}{2}J_{max}T_j^2 \\
 X_1 &= \frac{1}{6}J_{max}T_j^3 \\
 A_2 &= A_{max} \\
 V_2 &= A_{max}T_a + V_1 \\
 X_2 &= \frac{1}{2}A_{max}T_a^2 + V_1T_a + X_1 \\
 A_3 &= -J_{max}T_j + A_{max} \\
 V_3 &= -\frac{1}{2}J_{max}T_j^2 + A_{max}T_j + V_2 \\
 X_3 &= -\frac{1}{6}J_{max}T_j^3 + \frac{1}{2}A_{max}T_j^2 + V_2T_j + X_2 \\
 A_4 &= 0 \\
 V_4 &= V_{max} \\
 X_4 &= V_{max}T_{vc} + X_3 \\
 A_5 &= -J_{max}T_j + A_4 \\
 V_5 &= -\frac{1}{2}J_{max}T_j^2 + A_4T_j + V_4 \\
 X_5 &= -\frac{1}{6}J_{max}T_j^3 + \frac{1}{2}A_4T_j^2 + V_4T_j + X_4 \\
 A_6 &= -A_{max} \\
 V_6 &= -A_{max}T_a + V_5 \\
 X_6 &= -\frac{1}{2}A_{max}T_a^2 + V_5T_a + X_5 \\
 A_F &= J_{max}T_j + A_6 \\
 V_F &= \frac{1}{2}J_{max}T_j^2 + A_6T_j + V_6 \\
 D &= \frac{1}{6}J_{max}T_j^3 + \frac{1}{2}A_6T_j^2 + V_6T_j + X_6
 \end{aligned} \tag{D.1}$$

A partir des equations D.1, 3.16 et 3.19 on obtient :

$$D = \frac{V_{max}}{A_{max}J_{max}} (-V_{max}J_{max} - A_{max}^2 + TA_{max}J_{max}) \tag{D.2}$$

Et donc on retrouve la relation :

$$T = \frac{D}{V_{max}} + \frac{V_{max}}{A_{max}} + \frac{A_{max}}{J_{max}}$$

Références bibliographiques

- [Alami 98] R. Alami, R. Chatila, S. Fleury, M. Ghallab & F. Ingrand. *An Architecture for Autonomy*. International Journal of Robotics Research, vol. 17, pages 315–337, 1998. [10](#)
- [Albu-Schäffer 07] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck & G. Hirzinger. *The DLR lightweight robot : design and control concepts for robots in human environments*. Industrial Robot : An International Journal, vol. 34, no. 5, pages 376–385, 2007. [8](#)
- [Alili 11] S. Alili. *Interaction Décisionnelle Homme-Robot : Planification de tâche pour un robot interactif en environnement humain*. PhD thesis, 2011. [102](#)
- [Amato 98] N.M. Amato, O.B. Bayazit, L.K. Dale, C. Jones & D. Vallejo. *OBPRM : an obstacle-based PRM for 3D workspaces*. In International Workshop on Algorithmic Foundations of Robotics, pages 155–168. A. K. Peters, Ltd., Natick, MA, USA, 1998. [22](#)
- [Andersson 89] Russell L. Andersson. *Aggressive Trajectory Generator for a Robot Ping-Pong Player*. IEEE Control Systems Magazine, vol. 9, pages 15–20, February 1989. [27](#)
- [Bazaz 97] S.A. Bazaz & B. Tondu. *Optimal on-line robot trajectory generation in cartesian space*. In SPIE Proc. Architectures, Networks and Intelligent Systems for Manufacturing Integration, volume 3203, pages 2–10, 1997. [27](#)
- [Bazaz 98] S. A. Bazaz & B. Tondu. *3-cubic spline for on-line cartesian space trajectory planning of an industrial manipulator*. In Int. Workshop on Advanced Motion Control, pages 493–498, Coimbra, 1998. [27](#)
- [Begeman 99] P.C. Begeman, K. Pratima & P. Prasad. *Bending strength of the human cadaveric forearm due to lateral loads*. In Proceedings 43rd Stapp Car Crash Conference, SAE 99SC24, pages 343–350. Society of Automotive Engineers, 1999. [13](#)
- [Bestaoui 89] Y. Bestaoui. *On line motion generation with velocity and acceleration constraints*. Robotics and Autonomous Systems, vol. 5, pages 279–288, 1989. [27](#)
- [Biagiotti 08] L. Biagiotti & C. Melchiorri. *Trajectory planning for automatic machines and robots*. Springer, 2008. [24](#), [25](#), [56](#), [73](#)
- [Bicchi 04] A. Bicchi & G. Tonietti. *Dealing with the Safety-Performance Trade-Off in Robot Arms Design and Control*. IEEE Robotics and Automation Magazine, vol. (in press), 2004. [13](#)
- [Biggs 03] G. Biggs & B. Macdonald. *A Survey of Robot Programming Systems*. In in Proceedings of the Australasian Conference on Robotics and Automation, CSIRO. Citeseer, 2003. [7](#)

- [Bobrow 85] J.E. Bobrow, S. Dubowsky & J.S. Gibson. *Time optimal control of robotic manipulators along specified paths*. International Journal of Robotics Research, vol. 4, pages 3–17, 1985. [26](#)
- [Bobrow 88] J.E. Bobrow. *Optimal robot path planning using the minimum-time criterion*. IEEE Journal of Robotics and Automation, vol. 4, pages 443–450, August 1988. [26](#)
- [Borst 09] C. Borst, T. Wimbock, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P.R. Giordano, R. Konietzschke, W. Sepp, S. Fuchs *et al.* *Rollin'Justin-Mobile platform with variable base*. In IEEE International Conference on Robotics and Automation, pages 1597–1598. IEEE, 2009. [8](#)
- [Brady 82] M. Brady, J. Hollerbach, T. Johnson & T. Lozano-Perez. *Robot motion, planning and control*. The MIT Press, Cambridge, Massachusetts, 1982. [25](#)
- [Broquère 08] X. Broquère, S. Sidobre & I. Herrera-Aguilar. *Soft motion trajectory planner for service manipulator robot*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2808–2813, Sept. 2008. [5](#), [31](#), [95](#)
- [Broquère 10] X. Broquère, D. Sidobre & K. Nguyen. *From Motion Planning to Trajectory Control with Bounded Jerk for Service Manipulator Robots*. In IEEE International Conference on Robotics and Automation, pages 4505–4510, May 2010. [4](#), [63](#)
- [Buck 59] C.A. Buck, FB Dameron, MJ Dow & HV Skowlund. *Study of normal range of motion in the neck utilizing a bubble goniometer*. Archives of physical medicine and rehabilitation, vol. 40, page 390, 1959. [13](#)
- [Canny 88] J.F. Canny. *The complexity of robot motion planning*. MIT Press, Cambridge, MA, USA, 1988. [19](#), [21](#)
- [Castain 84] R.H. Castain & R.P. Paul. *An on-line dynamic trajectory generator*. International Journal of Robotics Research, vol. 3, pages 68–72, 1984. [25](#), [27](#)
- [Chand 85] S. Chand & K. L. Doty. *On-line polynomial trajectories for robot manipulators*. International Journal of Robotics Research, vol. 4, pages 38–48, Summer 1985. [26](#)
- [Chatila 02] R. Chatila, R. Alami, T. Simeon, J. Pettre & L. Jaillet. *Safe, reliable and friendly interaction between humans and humanoids*. In IARP International Workshop on Humanoid and Human Friendly Robotics, pages 83–87, Tsukuba, Japan, December 2002. [14](#)
- [Choset 05] H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki & S. Thrun. *Principles of robot motion : Theory, algorithms, and implementations (intelligent robotics and autonomous agents)*. The MIT Press, June 2005. [19](#)
- [Clodic 07] Aurelie Clodic. *Supervision pour un robot interactif : Action et interaction pour un robot autonome en environnement humain*. PhD thesis, Université Paul Sabatier, June 2007. [10](#)
- [Clodic 09] Aurélie Clodic, Hung Cao, Samir Alili, Vincent Montreuil, Rachid Alami & R. Chatila. *SHARY : A Supervision System Adapted to Human-Robot Interaction*. In Oussama Khatib, Vijay Kumar & George Pappas, éditeurs, Experimental Robotics, volume 54 of *Springer Tracts in Advanced Robotics*, pages 229–238. Springer Berlin / Heidelberg, 2009. [10](#)

- [Cortés 04] J. Cortés & T. Siméon. *Sampling-based motion planning under kinematic loop-closure constraints*. In International Workshop on Algorithmic Foundations of Robotics, Utrecht, Netherland, 2004. [86](#)
- [Dam 98] E.B. Dam, M. Koch & M. Lillholm. *Quaternions, interpolation and animation*. 1998. [17](#)
- [De Santis 08] A. De Santis, B. Siciliano, A. De Luca & A. Bicchi. *An atlas of physical human-robot interaction*. Mechanism and Machine Theory, vol. 43, no. 3, pages 253–270, March 2008. [13](#)
- [Devore 99] J.L. Devore & N.R. Farnum. *Applied statistics for engineers and scientists*. Duxbury Press, 1999. [13](#)
- [Dombre 07] E. Dombre & W. Khalil. *Robot Manipulators : Modeling, Performance Analysis and Control*. 2007. [20](#), [24](#)
- [Flash 84] T. Flash & N. Hogan. *The Coordination of Arm Movements : An Experimentally Confirmed Mathematical Model*. Journal of neuroscience, vol. 5, pages 1688–1703, 1984. [14](#), [29](#)
- [Fleury 97] S. Fleury, M. Herrb & R. Chatila. *GenoM : A Tool for the Specification and the Implementation of Operating Modules in a Distributed Robot Architecture*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 842–848, 1997. [9](#), [92](#)
- [Fong 03] T. Fong, I. Nourbakhsh & K. Dautenhahn. *A Survey of Socially Interactive Robots*. Robotics and Autonomous Systems, vol. 42, no. 3-4, 2003. [14](#)
- [Fuchs 09] M. Fuchs, C. Borst, P.R. Giordano, A. Baumann, E. Kraemer, J. Langwald, R. Gruber, N. Seitz, G. Plank, K. Kunze *et al.* *Rollin'Justin-Design considerations and realization of a mobile platform for a humanoid upper body*. In IEEE International Conference on Robotics and Automation, pages 4131–4137. IEEE, 2009. [8](#)
- [Geering 86] H. Geering, L. Guzzella, S. Hepner & C. Onder. *Time-optimal motions of robots in assembly tasks*. IEEE transactions on automatic control, vol. 31, no. 6, pages 512–518, 1986. [26](#)
- [Gerelli 10] O. Gerelli & C.G.L. Bianco. *A discrete-time filter for the on-line generation of trajectories with bounded velocity, acceleration, and jerk*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 3989–3994, may 2010. [26](#)
- [Gharbi 10] M. Gharbi. *Planification de mouvements et manipulation d'objets par des torses humanoïdes*. PhD thesis, 2010. [102](#)
- [Goodrich 07] M.A. Goodrich & A.C. Schultz. *Human-robot interaction : a survey*. Foundations and Trends in Human-Computer Interaction, vol. 1, no. 3, pages 203–275, 2007. [13](#)
- [Haddadin 08a] Sami Haddadin, Alin Albu-Schaffer, Mirko Frommberger & Gerd Hirzinger. *The role of the robot mass and velocity in physical human-robot interaction - Part II : Constrained blunt impacts*. In IEEE International Conference on Robotics and Automation, pages 1339–1345, Pasadena, USA, May 2008. [12](#)
- [Haddadin 08b] Sami Haddadin, Alin Albu-Schaffer & Gerd Hirzinger. *The role of the robot mass and velocity in physical human-robot interaction - Part I : Non-constrained blunt impacts*. In

- IEEE International Conference on Robotics and Automation, pages 1331–1338, Pasadena, USA, May 2008. [12](#)
- [Haddadin 09] Sami Haddadin, Alin Albu-Schäffer & Gerd Hirzinger. *Requirements for Safe Robots : Measurements, Analysis and New Insights*. Int. J. Rob. Res., vol. 28, pages 1507–1527, November 2009. [12](#)
- [Hall 63] E.T. Hall. *A System for the Notation of Proxemic Behavior I*. American anthropologist, vol. 65, no. 5, pages 1003–1026, 1963. [78](#)
- [Han 01] L. Han & N.M. Amato. *A Kinematics-Based Probabilistic Roadmap Method for Closed Chain Systems*. In B. R. Donald, K. M. Lynch & D. Rus, éditeurs, International Workshop on Algorithmic Foundations of Robotics, pages 233–246. A K Peters, Wellesley, MA, 2001. [86](#)
- [Haschke 08] R. Haschke, E. Weitnauer & H. Ritter. *On-Line Planning of Time-Optimal, Jerk-Limited Trajectories*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3248–3253. IEEE, 2008. [28](#), [29](#), [31](#), [103](#)
- [Herrera-Aguilar 05] I. Herrera-Aguilar & D. Sidobre. *On-line trajectory planning of robot manipulator's end effector in Cartesian Space using quaternions*. In 15th Int. Symposium on Measurement and Control in Robotics, 2005. [28](#)
- [Herrera-Aguilar 07] I. Herrera-Aguilar. *Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service*. PhD thesis, Université de Toulouse, Toulouse, France, September 2007. [28](#), [31](#), [37](#), [40](#), [103](#)
- [Hirzinger 02] G. Hirzinger, N. Sporer, A. Albu-Schäffer, M. Hähnle, R. Krenn, A. Pascucci & M. Schedl. *DLR's Torque-Controlled Light Weight Robot III - Are We Reaching the Technological Limits Now ?* In IEEE International Conference on Robotics and Automation, pages 1710–1716. IEEE, 2002. [8](#)
- [Hogan 84] N. Hogan. *An organizing principle for a class of voluntary movements*. Journal of neuroscience, vol. 4, pages 2745–2754, 1984. [14](#), [29](#)
- [Hollerbach 83] J.M. Hollerbach. *Dynamic Scaling of Manipulator Trajectories*. 1983. [26](#)
- [Hsu 03] D. Hsu, T. Jiang, J. Reif & Z. Sun. *The bridge test for sampling narrow passages with probabilistic roadmap planners*. In IEEE International Conference on Robotics and Automation, volume 3, pages 4420–4426, sept. 2003. [22](#)
- [Ikuta 03] K. Ikuta, H. Ishii & M. Nokata. *Safety Evaluation Method of Design and Control for Human-Care Robots*. International Journal of Robotics Research, vol. 22, no. 5, pages 281–297, 2003. [12](#), [13](#), [23](#)
- [ISO-10218 06] ISO-10218. *Robot for industrial environments - Safety requirements - Part 1 :Robot*. 2006. [12](#)
- [Jaillet 05] L. Jaillet, A. Yershova, S.M. LaValle & T. Siméon. *Adaptive tuning of the sampling domain for dynamic-domain RRTs*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2851–2856, August 2005. [22](#)

- [Jaillet 08] L. Jaillet, J. Cortés & T. Siméon. *Transition-based RRT for path planning in continuous cost spaces*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2145–2150, sept. 2008. [22](#), [24](#), [78](#)
- [Kavraki 96] L.E. Kavraki, P. Svestka, J.-C. Latombe & M.H. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pages 566–580, aug 1996. [22](#)
- [Kavraki 08] L.E. Kavraki & S.M. LaValle. Motion Planning In Springer Handbook of Robotics, by B. Siciliano and O. Khatib. 2008. [21](#)
- [Khalil 99] W. Khalil & E Dombre. Modélisation identification et commande des robots, volume 56. Hermes, 1999. [20](#), [25](#)
- [Khalil 03] W. Khalil & E Dombre. Modeling, identification and control of robots, volume 56. ASME, 2003. [24](#)
- [Khatib 86] O Khatib. *Real-time obstacle avoidance for manipulators and mobile robots*. International Journal of Robotics Research, vol. 5, no. 1, pages 90–98, 1986. [21](#), [24](#)
- [Koay 07] K.L. Koay, EA Sisbot, DS Syrdal, ML Walters, K. Dautenhahn & R. Alami. *Exploratory study of a robot approaching a person in the context of handing over an object*. Proc. of AAAI-SS on Multi-disciplinary Collaboration for Socially Assistive Robotics, pages 18–24, 2007. [78](#)
- [Kröger 10a] T. Kröger. On-line trajectory generation in robotic systems, volume 58 of *Springer Tracts in Advanced Robotics*. Springer, Berlin, Heidelberg, Germany, first edition, jan 2010. [25](#), [26](#)
- [Kröger 10b] T. Kröger & F.M. Wahl. *Online Trajectory Generation : Basic Concepts for Instantaneous Reactions to Unforeseen Events*. Robotics, IEEE Transactions on, vol. 26, no. 1, pages 94–111, 2010. [29](#), [31](#), [103](#)
- [Kuffner 00] Jr. Kuffner J.J. & S.M. LaValle. *RRT-connect : An efficient approach to single-query path planning*. In IEEE International Conference on Robotics and Automation, volume 2, pages 995–1001, 2000. [22](#), [78](#)
- [Kulić 05] Dana Kulić & Elizabeth Croft. *Safe planning for human-robot interaction*. Journal of Robotic Systems, vol. 22, no. 7, pages 383–396, 2005. [24](#)
- [Kumar Pandey 10] A. Kumar Pandey & R. Alami. *Mightability : A Multi-State Visuo-Spatial Reasoning for Human-Robot Interaction*. In Experimental Robotics (ISER), 12th International Symposium on, 2010. [102](#)
- [Lambrechts 05] P. Lambrechts, M. Boerlage & M. Steinbuch. *Trajectory planning and feedforward design for electromechanical motion systems*. Control Engineering Practice, vol. 13, no. 2, pages 145–157, 2005. [28](#), [29](#)
- [Larsen 99] E. Larsen, S. Gottschalk, M.C. Lin & D. Manocha. *Fast proximity queries with swept sphere volumes*. 1999. [94](#)
- [Latombe 91] J.-C. Latombe. Robot motion planning. Kluwer Academic Publishers, Boston, MA, 1991. [19](#)

- [LaValle 98] S.M. LaValle. *Rapidly-exploring random trees : A new tool for path planning*. Rapport technique 98–11, Computer Science Departement, Iowa State University, October 1998. [22](#)
- [LaValle 01] Steven M. LaValle & J. Kuffner. *Rapidly-Exploring Random Trees : Progress and Prospects*. In *Algorithmic and Computational Robotics : New Directions*, pages 293–308, Wellesley, MA, 2001. [22](#)
- [LaValle 06] S.M. LaValle. Planning algorithms. [Online], 2006. [19](#), [21](#)
- [Lin 83] C. S. Lin, Po Rong Chang & J. Y. S. Luh. *Formulation and optimization of Cubic Polynomial joint trajectories for Industrial Robots*. IEEE Transactions on Automatic Control, vol. AC-28, pages 1066–1074, December 1983. [26](#)
- [Liu 02] Steven Liu. *An on-line reference-trajectory generator for smooth motion of Impulse-Controlled Industrial Manipulators*. In *7th International Workshop on Advanced Motion Control*, pages 365–370, 2002. [27](#), [37](#)
- [Llibre 95] M. Llibre. *Définition de trajectoires d'attitude continues en accélération*. 1995. [26](#)
- [Lloyd 93] John Lloyd & Vincent Hayward. *Trajectory Generation for Sensor-Driven and Time-Varying Tasks*. International Journal of Robotics Research, vol. 12, pages 380–393, August 1993. [27](#), [81](#), [82](#)
- [Lozano-Pérez 83] T. Lozano-Pérez. *Spatial Planning : A Configuration Space Approach*. IEEE Transactions on Computers, vol. C–32, no. 2, pages 108–120, Feb. 1983. [19](#)
- [Macfarlane 03] Sonja Macfarlane & Elizabeth Croft. *Jerk-Bounded Manipulator Trajectory Planning : Design for Real-Time Applications*. IEEE Transactions on Robotics and Automation, vol. 19, pages 42–52, February 2003. [28](#), [82](#), [83](#)
- [Mainprice 11] J. Mainprice, E.A. Sisbot, L. Jaillet, J Cortés, T. Siméon & R. Alami. *Planning Human-aware motions using a sampling-based costmap planner*. In *IEEE International Conference on Robotics and Automation*, 2011. [24](#), [77](#), [78](#), [102](#)
- [Maldonado 10] Alexis Maldonado, Ulrich Klank & Michael Beetz. *Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information*. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 18-22 2010. [8](#)
- [Mallet 10] A. Mallet, C. Pasteur, M. Herrb, S. Lemaignan & F. Ingrand. *GenoM3 : Building middleware-independent robotic components*. In *IEEE International Conference on Robotics and Automation*, pages 4627–4632. IEEE, 2010. [9](#)
- [Marin-Urias 09] L.F. Marin-Urias. *Planification et contrôle de mouvements en interaction avec l'homme. Reasoning about space for human-robot interaction*. PhD thesis, 2009. [102](#)
- [Marler 05] R.T. Marler, S. Rahmatalla, M. Shanahan & K. Abdel-Malek. *A new discomfort function for optimization-based posture prediction*. 2005. [79](#)
- [Melvin 73] J.W. Melvin, R.G. Snyder, L.W. Travis & N.R. Olson. *Response of Human Larynx to Blunt Loading*. 1973. [13](#)

- [Melvin 80] J. Melvin. *Human tolerance to impact conditions as related to motor vehicle design*. Society of Automotive Engineers J885, 1980. 13
- [Mertz 71] H.J. Mertz & L.M. Patrick. *Strength and response of the human neck*. 1971. 13
- [Miller 89] M.A. Miller. *The biomechanical response of the lower abdomen to belt restraint loading*. The Journal of trauma, vol. 29, no. 11, page 1571, 1989. 13
- [Moon 09] S. Moon & G.S. Virk. *Survey on ISO standards for industrial and service robots*. In ICROS-SICE International Joint Conference, pages 1878–1881. IEEE, 2009. 12
- [Nilsson 69] N.J. Nilsson. *A mobius automation : an application of artificial intelligence techniques*. In International Joint Conference on Artificial Intelligence, volume 1, pages 509–520, San Francisco, CA, USA, 1969. Morgan Kaufmann Publishers Inc. 21
- [Nokata 02] M. Nokata, K. Ikuta & H. Ishii. *Safety-optimizing method of human-care robot design and control*. In IEEE International Conference on Robotics and Automation, ICRA, volume 2, pages 1991–1996, Washington DC, USA, May 2002. 23
- [Ott 06] C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schaffer, B. Brunner, H. Hirschmuller, S. Kielhofer, R. Konietschke, M. Suppa, T. Wimbock, F. Zacharias & G. Hirzinger. *A Humanoid Two-Arm System for Dexterous Manipulation*. In IEEE/RAS International Conference on Humanoid Robots, pages 276–283, December 2006. 8
- [Pacchierotti 06] Elena Pacchierotti, Henrik Christensen & Patric Jensfelt. *Embodied Social Interaction for Service Robots in Hallway Environments*. Field and Service Robotics, pages 293–304, 2006. 23
- [Park 09] J.-J. Park & J.-B. Song. *Collision analysis and evaluation of collision safety for service robots working in human environments*. In Advanced Robotics, 2009. ICAR 2009. International Conference on, pages 1–6, 2009. 12, 13
- [Paul 79] R.P. Paul. *Manipulator Cartesian Path Control*. IEEE Trans. Syst., Man Cybern., vol. 9, pages 702–711, May 1979. 25
- [Paul 81] R.P. Paul. *Robot manipulators. mathematics, programming and control*. The MIT Press, The Massachusetts Institute of Technology, 1981. 25
- [Pfeiffer 87] Friedrich Pfeiffer & Rainer Johanni. *A concept for manipulator trajectory planning*. IEEE Journal of Robotics and Automation, vol. RA-3, pages 115–123, April 1987. 26
- [Phriends 09] Phriends. *Physical Human Robot Interaction : depENDability and Safety*. In European FP6 project, <http://www.phriends.eu/>, October 2006 - November 2009. 12
- [Piazzi 97] A. Piazzi & A. Visioli. *An interval algorithm for minimum-jerk trajectory planning of robot manipulators*. In IEEE Conf. on Decision and Control, pages 1924–1927, December 1997. 26
- [Piegl 87] L.A. Piegl & W. Tiller. *Curve and surface constructions using rational B-splines*. Comput. Aided Des., vol. 19, pages 485–498, November 1987. 18
- [Piegl 97] L.A. Piegl & W. Tiller. *The NURBS book*. Springer Verlag, 1997. 18

- [Press 07] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling *et al.* Numerical recipes, volume 3. Cambridge university press Cambridge, 2007. [48](#)
- [Rodriguez 06] S. Rodriguez, X. Tang, J.-M. Lien & N.M. Amato. *An obstacle-based rapidly-exploring random tree*. In IEEE International Conference on Robotics and Automation, pages 895–900, may 2006. [22](#)
- [Sahar 86] G. Sahar & J.M. Hollerbach. *Planning of minimum-time trajectories for robot arms*. International Journal of Robotics Research, vol. 5, no. 3, page 90, 1986. [26](#)
- [Saut 10] J.P. Saut, M. Gharbi, J. Cortés, D. Sidobre & T. Siméon. *Planning Pick-and-Place tasks with two-hand regrasping*. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 4528–4533. IEEE, 2010. [102](#)
- [Schwartz 83] J. T. Schwartz & M. Sharir. *On the Piano Movers' Problem : III. Coordinating the Motion of Several Independent Bodies*. International Journal of Robotics Research, vol. 2, pages 46–75, September 1983. [21](#)
- [Shikin 95] E.V. Shikin & A.I. Plis. Handbook on Splines for the User. CRC, 1995. [18](#)
- [Shin 85] K.G. Shin & N.D. McKay. *Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints*. IEEE Transactions on Automatic Control, vol. AC-30, pages 531–541, June 1985. [26](#)
- [Shoemake 85] Ken Shoemake. *Animating rotation with Quaternion curves*. Proc. on Computer graphics and interactive techniques, vol. 19, pages 245–254, July 1985. [17](#)
- [Siciliano 08] B. Siciliano & O. Khatib. Springer Handbook of Robotics. 2008. [11](#), [16](#)
- [Sidobre 10] D. Sidobre, J.-P. Saut, X. Broquère & J. Mainprice. *Reactive Grasping for Human-Robot Interaction*. In Robotics Science and Systems, Workshop Representations for object grasping and manipulation in single and dual arm tasks, 2010. [5](#)
- [Siméon 00] T. Siméon, J.-P. Laumond & C. Nissoux. *Visibility-Based Probabilistic Roadmaps for Motion Planning*. Advanced Robotics Journal, vol. 14(6), pages 477–494, 2000. [22](#)
- [Siméon 01] T. Siméon, J.-P. Laumond & F. Lamiroux. *Move3D : a generic platform for path planning*. In International Symposium on Assembly and Task Planning, volume 4, pages 25–30, 2001. [24](#), [94](#)
- [Sisbot 07] L.F. Sisbot E.A.and Marin-Urias, R. Alami & T. Simeon. *Human Aware Mobile Robot Motion Planner*. IEEE Transactions on Robotics, vol. 23, pages 874–883, 2007. [14](#), [23](#), [77](#), [78](#)
- [Sisbot 08] E.A. Sisbot. *Towards Human-Aware robot Motions*. PhD thesis, LAAS/CNRS, Université Paul Sabatier, October 2008. [102](#)
- [Sisbot 10] E.A. Sisbot, L.F. Marin-Urias, X. Broquère, D. Sidobre & R. Alami. *Synthesizing Robot Motions Adapted to Human Presence*. International Journal of Social Robotics, vol. 2, no. 3, pages 329–343, 2010. [4](#)
- [Slotine 89] J. J. E. Slotine & H. S. Yang. *Improving the efficiency of time-optimal path-following algorithms*. IEEE Trans. on Robotics and Automation, vol. 5, pages 118–124, February 1989. [26](#)

- [SMERobot™ 09] SMERobot™. *SME : Small and Medium sized Enterprise*. In European FP6 project, <http://www.smerobot.org/>, March 2005 - May 2009. 12
- [Takemura 07] H. Takemura, K. Ito & H. Mizoguchi. *Person following mobile robot under varying illumination based on distance and color information*. In IEEE International Conference on Robotics and Biomimetics, ROBIO, pages 1500–1505, Bangkok, Thailand, December 2007. 23
- [Taylor 79] R.H. Taylor. *Planning and Execution of Straight Line Manipulator Trajectories*. IBM Journal of Research and Development, vol. 23, pages 424–436, 1979. 25
- [Toudu 99] B. Toudu & S. A. Bazaz. *The three-cubic method : An optimal online robot joint trajectory generator under velocity, acceleration and wandering constraints*. Journal of Robotics Systems, vol. 18(9), pages 893–901, September 1999. 27
- [Švestka 97] P. Švestka. *Robot Motion Planning using Probabilistic Roadmaps*. PhD thesis, Universiteit Utrecht, 1997. 22
- [Wagner 07] D. Wagner & D. Schmalstieg. *Artoolkitplus for pose tracking on mobile devices*. In Proceedings of 12th Computer Vision Winter Workshop (CVWW'07), pages 139–146. Cite-seer, 2007. 92
- [Yoshimi 06] T. Yoshimi, M. Nishiyama, T. Sonoura, H. Nakamoto, S. Tokura, H. Sato, F. Ozaki, N. Matsuhira & H. Mizoguchi. *Development of a Person Following Robot with Vision Based Target Detection*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pages 5286–5291, Beijing, China, October 2006. 23
- [Zinn 04] M. Zinn, O. Khatib, B. Roth & J. K. Salisbury. *Playing it safe [human-friendly robots]*. vol. 11, no. 2, pages 12–21, 2004. 13
- [Zollo 05] L. Zollo, B. Siciliano, A. De Luca, E. Guglielmelli & P. Dario. *Compliance Control for an Anthropomorphic Robot with Elastic Joints : Theory and Experiments*. Journal of Dynamic Systems, Measurement, and Control, vol. 127, no. 3, pages 321–328, September 2005. 13

Abstract

The context of service robotics is characterized by the presence of humans in the vicinity of the robot. The movements of these robots should not disturb the safety of humans or their comfort. From the motion planning point of view, the planner must both avoid hitting humans or colliding with the environment and also adapt the robot's kinematic limits depending on the proximity of humans. At each level of the system (Planning and execution / Control), the robot must ensure the safety and the comfort of humans.

We propose an approach of motion planning and motion control based on polynomial trajectories. In the first part, we present a trajectory generator which limits the speed, the acceleration and the jerk (derivative of the acceleration). The motion planner generates trajectories consisting of series of segments of cubic polynomial curves. The mono-dimensional case is first introduced and then extended to the multi-dimensional one.

In the second part, we propose to approximate the trajectories by sequences of triplets of segments of cubic curves. This method allows to find trajectories that respect a given maximum error. These trajectory generators are integrated into the path planner and produce directly executable motion. An original application of the trajectory approximation is the approximation of a trajectory defined in Cartesian space by a trajectory defined in the joint space. This approach simplifies the structure of the robot controller.

The presence of humans in the workspace of the robot requires also an adaptation of the trajectories during the execution. We propose a method to adapt the motion law of the multidimensional path at runtime.

This work, conducted as part of the European project DEXMART and the ANR project ASSIST, has been integrated and validated on the Jido and PR2 platforms of LAAS-CNRS.

Keywords : Robotics, Trajectory planning, Human-Robot Interaction, Manipulation

Résumé

Le contexte de la robotique de service est caractérisé par la présence de l'homme dans l'espace de travail du robot. Les mouvements de ces robots ne doivent perturber ni la sécurité de l'homme ni son confort. D'un point de vue planification de mouvement, le planificateur doit d'une part éviter de heurter l'homme ou l'environnement et d'autre part adapter les limites cinématiques du robot en fonction de la proximité de l'homme. A chaque niveau du système (planification et exécution/contrôle), le robot doit garantir la sécurité et le confort de l'homme.

Nous proposons une approche de la planification et du contrôle de mouvement basée sur des trajectoires polynomiales. Dans une première partie, nous présentons un générateur de trajectoires qui limite la vitesse, l'accélération et le jerk. Il génère des trajectoires composées de suites de segments de courbes cubiques. Le cas mono-dimensionnel est d'abord présenté puis étendu au cas multi-dimensionnel.

Dans une deuxième partie, nous proposons d'approximer les trajectoires par des suites de triplets de segments de courbes cubiques. Cette méthode permet de calculer des trajectoires respectant une erreur maximale donnée. Ces générateurs de trajectoire sont intégrés au planificateur de chemin et produisent des trajectoires directement exécutables. Une application originale de l'approximation permet d'approximer une trajectoire définie dans l'espace cartésien par une trajectoire définie dans l'espace articulaire. Cette approche simplifie la structure du contrôleur du robot.

La présence de l'homme dans l'espace de travail du robot nécessite une adaptation des trajectoires pendant l'exécution. Nous proposons une méthode pour adapter la loi de mouvement de la trajectoire multidimensionnelle pendant l'exécution.

Ces travaux, menés dans le cadre du projet européen DEXMART et du projet ANR ASSIST, ont été intégrés et validés sur les plateformes Jido et PR2 du LAAS-CNRS.

Mots clés : Robotique, Planification de trajectoire, Interaction Homme-Robot, Manipulation