



HAL
open science

Contributions à la commande prédictive pour la marche des quadrupèdes sur terrain 3D

Fanny Risbourg

► **To cite this version:**

Fanny Risbourg. Contributions à la commande prédictive pour la marche des quadrupèdes sur terrain 3D. Automatique / Robotique. INSA de Toulouse, 2023. Français. NNT : 2023ISAT0007. tel-04071107

HAL Id: tel-04071107

<https://laas.hal.science/tel-04071107>

Submitted on 17 Apr 2023

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.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue le *31/01/2023* par :

Fanny Risbourg

**Contributions à la commande prédictive pour la marche des
quadrupèdes sur terrain 3D**

JURY

OLIVIER STASSE	Directeur de Recherche	Président
OLIVIER LY	Professeur	Rapporteur
ADRIEN ESCANDE	Chercheur	Rapporteur
MARGOT VULLIEZ	Chargée de Recherche	Examinatrice
GENTIANE VENTURE	Professeure	Examinatrice
NICOLAS MANSARD	Directeur de Recherche	Directeur de thèse

École doctorale et spécialité :

EDSYS : Robotique 4200046

Unité de Recherche :

LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes (UPR 8001)

Directeur de Thèse :

Nicolas Mansard

Rapporteurs :

Olivier Ly et Adrien Escande

Remerciements

Cette thèse a été pleine de rebondissements et de nombreuses personnes ont contribué à son bon déroulement, tant d'un point de vue technique, qu'humain.

Je voudrais commencer par remercier mon directeur de thèse Nicolas Mansard. Sans ton soutien je n'aurais pas commencé, continué ou fini ma thèse. Tu m'as appris beaucoup sur la robotique et la recherche mais tu m'as surtout poussé à aller au bout de mes réflexions et de mes convictions. Je te remercie pour la confiance que tu m'as transmise, qui m'a permis de faire une soutenance dont je suis fière. Tes enseignements continueront de guider mes recherches, quelque soit la voie que je choisis.

Au cours de ces trois ans, j'ai travaillé dans plusieurs environnements différents. J'y ai toujours rencontré des personnes chaleureuses et intéressantes qui m'ont intégré à leur univers. Ma première année à Wandercraft m'a fait découvrir l'ambiance startup et je souhaite remercier mes collègues et encadrants qui ont rendu cette expérience très sympathique. Lors de mon séjour en Italie, j'ai eu la chance d'arriver dans un openspace où j'ai été formidablement bien accueillie. Je vous remercie pour les visites, les soirées, les weekends et la patience avec mon italien bredouillant. Finalement j'ai passé la majeure partie de ma thèse au LAAS, où j'ai découvert un environnement de recherche agréable au sein de l'équipe Gepetto. Merci à tous les membres de cette équipe, mais aussi aux pièces rapportées de RAP, RIS ou DISCO pour les soirées bridge et les après-midi tarot.

Enfin je tiens à remercier mes amis, les vieux comme les plus récents. Ceux que je vois rarement et qui habitent loin comme ceux qui partagent mon quotidien (à base de repas canaille et d'escapades en montagne). Ce serait vachement moins sympa sans vous.

Merci aussi à toute ma famille. Papa et Maman, vous m'avez apporté un soutien sans faille, et je sais que même si mes choix peuvent vous dérouter et vous sortir de votre zone de confort, vous serez toujours là à oeuvrer pour mon bonheur. Marion, je ne serai pas si résistante si tu n'avais pas été ma grande soeur, à m'initier aux joutes verbales ou physiques. J'ai enfin une pensée pour mes grands-parents, et particulièrement Jacques, qui m'a transmis sa rigueur intellectuelle et son amour des sciences.

Finalement, merci Corentin. Mais par contre je suis très caramel, bonbon et chocolat, surtout partagés avec toi.

Table des matières

Table des Abréviations et Acronymes	v
Introduction	1
Contexte	1
Contributions	5
Publications	7
1 Revue de Littérature	9
1.1 Contrôle des robots aveugles	10
1.2 Des robots qui voient	19
2 Les outils du roboticien	23
2.1 Modélisation	24
2.2 Contrôle Optimal	28
2.3 Le robot quadrupède Solo	34
3 Planification réactive et évitement d'obstacles	39
3.1 Planification réactive en environnements complexes	41
3.2 Adaptation du contrôleur de Solo	46
3.3 Étude expérimentale	48
3.4 Analyse quantitative	49
4 MPC avec optimisation de la durée des pas	53
4.1 Présentation de l'OCP	54
4.2 Résultats de marche en simulation	59
4.3 Apprentissage pour le démarrage à chaud de Crocoddyl	62

4.4	Simulation avec Crocodyl	68
5	Contrôle avec un MPC corps-complet	71
5.1	Calcul des consignes pour le contrôleur en impédance	73
5.2	Formulation du problème de contrôle optimal	74
5.3	Résultats	76
	Conclusion	83
	Contributions	83
	Perspectives pour le contrôle de la locomotion des robots marcheurs	84
	Perspectives pour la robotique	84
A	Contrôle par admittance appliqué à l'exosquelette Atalante	85
A.1	Contrôleur d'Atalante	87
A.2	Stabilisation par admittance	88
A.3	Compensation de flexibilités	90
A.4	Résultats expérimentaux	92
	Bibliographie	114

Table des Abréviations et Acronymes

CoM	Center of Mass
CoP	Center of Pressure
CPG	Central Pattern Generator
DoF	Degree of Freedom
DCM	Divergent Component of Motion
DDP	Differential Dynamic Programming
DP	Dynamic Programming
FDDP	Feasibility-driven Differential Dynamic Programming
HZD	Hybrid Zero Dynamics
IK	Inverse Kinematics
IMU	Inertial Measurement Unit
LIPM	Linear Inverted Pendulum Model
LQR	Linear Quadratic Regulator
MIP	Mixed Integer Program
MoCap	Motion Capture
MPC	Model Predictive Control
OCP	Optimal Control Problem
QP	Quadratic Problem
ZMP	Zero Moment Point

Introduction

Contexte

Il y a quelques millénaires, l'invention de la roue a révolutionné les transports et l'agriculture. Depuis elle équipe la plupart de nos véhicules et machines. Pourtant, les jambes permettent aux hommes et animaux de se mouvoir dans une bien plus grande variété d'environnements. En voyant un isard des Pyrénées sauter de rocher en rocher et un bus coincé par un tronc d'arbre tombé à terre, on comprend l'avantage de ce moyen de locomotion. Alors pourquoi un tel attachement de l'homme aux machines roulantes ?

Sans doute car les mystères de la locomotion n'ont pas encore été percés. Le cerveau parvient à orchestrer tous les muscles du corps pour lui permettre de se déplacer. Sans obstacle, on parle de marche réflexe : pas besoin de réfléchir où placer ses pieds, le cerveau le fait de manière inconsciente. En terrain plus irrégulier en revanche, nous devons regarder nos pieds pour choisir où les poser [Lau17]. Mais depuis des siècles, des scientifiques de différents domaines cherchent à comprendre comment s'effectue cette orchestration, sans y parvenir pleinement.

Les travaux ont commencé à la Renaissance quand Giovanni Alfonso Borelli décrit pour la première fois la mécanique du corps humain dans son livre *De motu animalium* (Mouvement des animaux). C'est la naissance de la biomécanique. Des évolutions technologiques ont par la suite permis de nouvelles analyses, comme l'invention de la chronophotographie qui permet à Eadweard Muybridge de décomposer le galop d'un cheval et la marche humaine [Lau16].

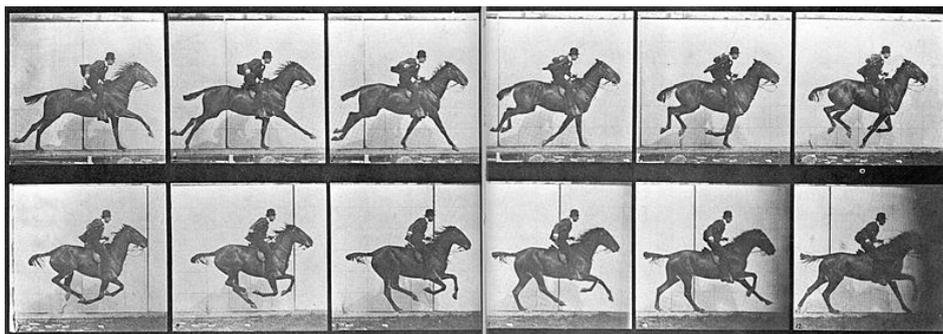


FIGURE 1 – Chronophotographie de cheval au galop par Eadweard Muybridge (1887)

Les technologies actuelles permettent d'étudier plus précisément l'activation des muscles et de certaines parties du cerveau, les forces de contact avec l'environnement et les trajectoires des différentes parties du corps. Reste maintenant à comprendre quels principes régissent l'orchestration lors de différentes tâches de locomotion.

Des robots à pattes...

En parallèle de cette étude de la locomotion et du développement de machines roulantes, des hommes ont rêvé de construire des robots à pattes, à leur image ou à celle d'autres animaux. Au 15^e siècle, parmi tous ses automates, Léonard de Vinci imagine ainsi l'automate chevalier. Habillé d'une armure il bouge les pieds et la tête grâce à un système de câbles [Ros94]. Des siècles plus tard, après la révolution industrielle, de nouvelles machines à pattes sont imaginées, comme le cheval mécanique de L.A. Rygg [Rai86] : Il dépose en 1893 un brevet pour la construction d'un cheval pouvant être monté et qui permettrait de générer un mouvement de marche à partir d'une source d'énergie par une série d'engrenages.

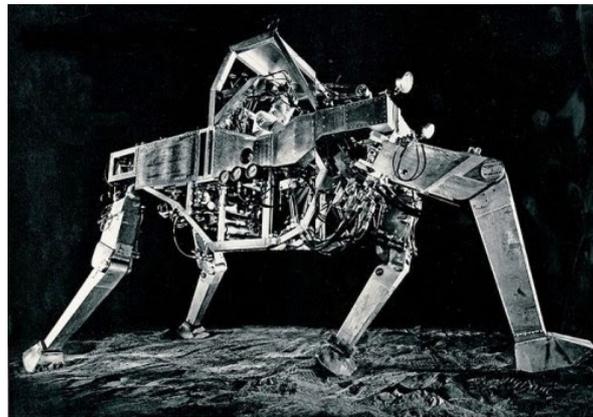


FIGURE 2 – Le camion qui marche de Ralph Mosher (1965) [Wal]

Dans les années 1950, il semble admis qu'une machine devra être contrôlée pour produire une locomotion fiable, et de premiers prototypes apparaissent. Ralph Mosher conçoit pour General Electric en 1965 la *Cybernetic Anthropomorphic Machine* (Machine anthropomorphe cybernétique) [McK91], aussi appelé *Camion qui marche*. C'est un quadrupède de 1 400 kg contrôlé par un opérateur humain qui depuis l'intérieur du véhicule appuie sur un ensemble des pédales. Des moteurs hydrauliques transmettent ensuite les mouvements désirés aux quatre pattes. C'est un premier succès qui est suivi par des essais d'exosquelettes pour augmenter la force d'un opérateur [Mos68].

Dans les années 1970, McGhee construit une série de robots hexapodes [Rai86] contrôlés par un ordinateur. Il parvient avec succès à marcher en équilibre statique au-dessus de trois jambes grâce à ses dix-huit moteurs. D'autres robots suivent et en 1984 Hirose [Hir84] parvient grâce à un astucieux assemblage de l'actionnement à faire monter un escalier à un robot quadrupède contrôlé par ordinateur. En parallèle de ce développement de robots qui marchent de façon statique, Roberge [Rob60] commence au MIT la recherche autour de l'équilibre d'un pendule inverse. Ces recherches permettront à Marc Raibert [Rai86] de construire dans les années 1980 les premiers robots hautement dynamiques, une série de monopodes qui parviennent en sautant à se déplacer à une vitesse de 2.2 m/s. Ses travaux s'intéressent au choix du point de contact des pieds, et sont transposables à des bipèdes ou quadrupèdes.

Plusieurs séries de robots humanoïdes japonais, notamment les robots ASIMO de Honda

[Hir07] et les robots du *Humanoid Robotic Program* (HRP) [Hir04] utilisent ces avancées pour permettre à des robots bipèdes de nouvelle génération de marcher et monter des escaliers. Depuis quelques années un grand nombre de quadrupèdes [Bel18a; Sem11; DC18a] et de bipèdes [Kui20; Sta17; Gon19a] démontrent de plus en plus de polyvalence et une grande robustesse sur une certaine variété de terrains.

...pour quoi faire ?

Les progrès en termes de design et contrôle des robots à pattes sont tels que quelques applications industrielles commencent à émerger. Après des décennies à faire avancer la technologie sans applications concrètes, peut-être est-il temps de se poser et de se demander à quoi de tels robots peuvent servir. Sans prétendre faire une revue exhaustive, quelques applications sont présentées si dessous accompagnées par les questions éthiques et sociétales qu'elles posent.

Un des premiers robots imaginé par l'homme est le robot Talos, dans la mythologie Grecque [May18]. Ce robot créé par Héphaïstos avait pour fonction de garder l'Île de Crète. La fonction militaire est une des plus répandues dans l'imaginaire collectif, véhiculée notamment à travers un vaste ensemble de livres et films, allant de l'œuvre d'Isaac Asimov à *Terminator*. Loin de la fiction, Boston Dynamics ne cache pas la fonction militaire de son robot BigDog [Rai08], partiellement financé par l'armée américaine. L'utilisation de robots militaires est présentée comme permettant de diminuer le coût humain de la guerre [Oba11]. Les implications réelles sont plus complexes. Par exemple, l'acceptabilité de la guerre peut impacter le nombre de conflits armés [Lin17]. De plus, l'autonomie décisionnelle des robots lève des questions quant à la responsabilité et à la prédictibilité des frappes, et rend difficile le respect du Droit international humanitaire [Sip]. Si l'utilisation de robots à pattes militaires en est à son balbutiement, des drones et robots roulants servent depuis plusieurs décennies dans des armées du monde entier [Spr13]. Plusieurs rapports [Liv; Hrw] ont montré que la politique américaine d'utilisation de drones dans la lutte contre le terrorisme était problématique, à la fois au regard de l'impact important sur les civils mais aussi car il favorisait le recrutement des organisations terroristes. Des fabricants prennent aujourd'hui position contre l'armement des robots [Ple], et des campagnes essaient d'influencer les politiques des états [Sto].

Une autre application des robots à pattes largement partagée dans la *Pop culture* est celle de robots sociaux, assistants personnels et compagnons pour lutter contre la solitude. La recherche sur ces applications [Gha21] porte principalement sur l'acceptabilité de ces robots [GA18]. De premiers robots ont été commercialisés à cet effet [Coo20], avec des capacités limitées par rapport aux attentes des consommateurs, résultant en plusieurs échecs commerciaux [Ack18; Tul19]. L'utilité même de ces robots par rapport à un animal ou humain est cependant peu étudiée [Bel18c; Hen21].

Dans le domaine médical, des exosquelettes [Høy22; Gur18a] voient le jour pour améliorer la mobilité de personnes handicapées ou permettre une meilleure rééducation après un accident. Ces exosquelettes sont aujourd'hui commercialisés pour une utilisation en hôpital dans un environnement contrôlé. Si des premières études tendent à montrer l'intérêt [Cal18]



FIGURE 3 – Robot quadrupède BigDog de Boston Dynamics

de tels outils pour la rééducation, les marches effectuées avec un tel dispositif diffèrent de l'allure naturelle du patient [Swa19]. Des versions personnelles pour une utilisation dans la vie de tous les jours sont en cours de développement [Rup16]. Les défis en termes de stabilité et robustesse ainsi que d'ergonomie sont cependant loin d'être résolus [FV20].

L'utilisation la plus répandue des robots à l'heure actuelle est industrielle, dans le cadre de l'automatisation des usines [Edw84]. Depuis la révolution industrielle, un nombre toujours plus grand de tâches peuvent-être effectuées par des robots [Bay18]. Les robots à pattes ont l'avantage de pouvoir être mobiles dans une usine sans nécessiter d'adaptation et ainsi de réaliser plusieurs tâches à moindre coût. Si des études montrent que l'automatisation et l'utilisation accrue de l'*intelligence artificielle* génèrent de l'emploi [Büh17], les compétences requises pour les emplois créés et supprimés ne sont pas les mêmes. Les dernières années ont ainsi montré une précarisation du milieu ouvrier [Mol22] qui fait craindre une augmentation des inégalités [Buh19] par une concentration accrue des moyens de production et une diminution de la répartition des richesses sans adaptation globale de la société. Peu d'études s'intéressent aux interactions homme-robots dans ce milieu [Cor22].

La dernière application à laquelle cette thèse s'intéressera est l'utilisation de robots à pattes pour l'exploration de milieux dangereux pour l'homme. Plusieurs exemples sont souvent cités dans ce domaine : l'exploration spatiale [Arm19], le déminage [DS07 ; Hab08] ou l'intervention en zone accidentée (par exemple suite à un incident nucléaire) [Zha18]. Les robots à roues sont aujourd'hui plus matures pour de telles tâches, mais les progrès de la locomotion à pattes pourrait voir arriver des robots suffisamment robustes pour de telles applications dans les années qui viennent [Dar].

D'autres domaines d'applications sont amenés à se développer au cours des années qui viennent, comme la robotique agricole qui utilise déjà des robots à roues [Ly15]. Cependant, le coût environnemental lié à la fabrication et l'utilisation des robots en termes de ressources et d'énergie est non négligeable et devrait être pris en compte dans l'évaluation de l'utilité des robots. Peu de ressources sont aujourd'hui disponibles à ce sujet, mais la recherche commence à se développer et plusieurs rapports sur l'impact du numérique ont déjà été publiés [Gre ;

[Ade](#)]. D'autres façons de penser la robotique émergent et le robot Yokobo [[Cap22](#)] illustre ce phénomène en revisitant complètement l'image traditionnelle du robot.

Contributions

Alors que les premiers robots marcheurs commencent à sortir des laboratoires, la locomotion des robots à pattes présente toujours de nombreux challenges à résoudre. Ces systèmes sont dits *sous-actionnés* car n'étant pas reliés au monde, la position et l'orientation de la base ne sont pas directement contrôlables. C'est par la position et la force des contacts avec l'environnement que ces six degrés de liberté peuvent être contrôlés. La force appliquée à ces points de contact est unilatérale et limitée par le cône de frottement. De plus, les points de contact avec l'environnement ne peuvent se faire que dans les limites cinématiques du robot. Ainsi les mouvements de la base sont limités et la gestion de l'équilibre en est affectée. Le contrôle de ces robots implique de prendre des décisions contraintes impliquant des modèles dynamiquement instables et des calculs coûteux en temps réel. Comme nous le verrons plus en détail dans le prochain chapitre, deux familles principales d'approches s'opposent pour répondre à ces problèmes.

Les approches fondées sur les modèles sont aujourd'hui les plus utilisées. Elles consistent à modéliser la dynamique du robot pour pouvoir prédire son évolution en fonction d'un contrôle appliqué. Le choix du contrôle est souvent fait en résolvant un problème de contrôle optimal (Optimal Control Problem - **OCP**) sur un horizon temporel fini. Cela permet d'optimiser de nombreux paramètres de la marche, comme la durée des pas, leur emplacement, et ce, avec des modèles de plus en plus riches de la dynamique des robots. La réactivité est essentielle pour la robustesse de la locomotion. Les problèmes d'optimisation sont donc généralement résolus en ligne à partir d'estimations de l'état courant dans un schéma de commande prédictive. Les approches fondées sur les données montrent des résultats impressionnants, mais souffrent d'un manque de garanties et de prédictibilité inhérent. Elles sont aussi plus difficiles à régler et à rendre polyvalentes. Malgré ces oppositions évidentes, ces deux approches reposent sur un point commun essentiel : la formulation d'un problème de contrôle optimal définissant un objectif numérique à optimiser ainsi qu'un cadre de simulation contraint.

Le choix de ce problème de contrôle optimal est essentiel. Un modèle réduit, qu'il comporte ou non des approximations pour linéariser ou rendre convexe la dynamique, permet une résolution plus rapide du problème de contrôle optimal en ligne, ou pour un horizon temporel plus long. Cependant, il peut être incapable de rendre compte de certains aspects cruciaux de la dynamique complète, éventuellement à cause des approximations réalisées. Les coûts et les contraintes utilisés pour guider la résolution vers une bonne solution sont aussi critiques et le réglage des pondérations peut changer radicalement le comportement. Finalement le choix des algorithmes de résolution numérique a un fort impact sur la qualité de la locomotion obtenue. Certains solveurs numériques permettent de résoudre rapidement des problèmes complexes, au risque de choisir des optimums locaux.

Le modèle centroïdal (que nous décrivons en détail dans la thèse) est largement utilisé pour la locomotion des robots quadrupèdes. Pour les robots quadrupèdes qui ont des pattes légères,

une approximation fréquente permet de modéliser l'évolution de la position et orientation de la base du robot en fonction des forces de contact avec l'environnement. Il dépend toutefois des positions des points de contact qui doivent être définis et ne peuvent être introduits par une relation linéaire. Des modèles plus riches sont étudiés pour palier ces limitations, en particulier le modèle complet de la dynamique (considérant le robot comme un système polyarticulé de corps rigides). Au cours de cette thèse, ce modèle a été utilisé dans des schémas de commande prédictive sur des robots quadripèdes et bipèdes contrôlés en couple, avec des résultats encourageants.

Cette thèse propose de contribuer à la locomotion des robots à pattes en questionnant le choix des modèles actuellement utilisés par les approches fondées sur les modèles. En particulier, nous proposons d'explorer plusieurs formulations de contrôle optimal pour la marche des quadripèdes et leur utilisation dans des cadres expérimentaux variés. L'objectif est d'aider à mieux cerner les limites des modèles de contrôle optimal existants, à proposer des ajouts à ces modèles, à évaluer la capacité de nos algorithmes de résolution à traiter ces ajouts et à comprendre leur conséquence sur le fonctionnement des robots réels par des approches expérimentales comparatives. Cela implique une approche transversale, partant de l'étude des modèles, la formulation des problèmes, leur résolution par des solveurs existants ou par le développement de méthode de résolution complémentaire et l'implémentation sur le robot, couronnement expérimental de l'étude.

Les deux premières parties de la thèse s'intéressent à la limite principale du modèle centroïdal : l'incapacité à prendre des décisions sur les contacts. Nous avons tout d'abord (Chapitre 3) étudié l'interaction entre un module de planification réactive et un schéma de contrôle par commande prédictive exploitant le modèle centroïdal, afin de permettre à un robot de raisonner sur son environnement lors du choix des contacts. Nous nous sommes ensuite (Chapitre 4) intéressés au choix de la durée des pas qui est intrinsèquement lié au choix des contacts. Nous avons proposé un schéma de contrôle par commande prédictive exploitant un modèle centroïdal augmenté d'une dynamique discrète permettant de choisir les positions et durées des contacts. La troisième partie de la thèse (Chapitre 5) nous a permis d'explorer les potentialités du modèle complet de la dynamique des robots en proposant un schéma de contrôle prédictif qui ne demande aucune information sur les positions des contacts et est adapté à la locomotion de robots contrôlés en impédance.

Cette thèse a ainsi apporté les contributions scientifiques suivantes dans le domaine :

- | | | |
|------------|---|--|
| Chapitre 3 | { | <p>La planification réactive de mouvements</p> <ul style="list-style-type: none"> — L'implémentation en accès-libre d'un schéma de contrôle complet pour le contrôle en temps réel du robot Solo. — L'analyse quantitative de ce schéma de contrôle pour en valider l'architecture. |
| Chapitre 4 | { | <p>L'optimisation de la durée des pas</p> <ul style="list-style-type: none"> — La formulation d'un OCP centroïdal qui optimise les positions et durées des pas. — L'entraînement d'un réseau de neurones pour le démarrage à chaud du solveur numérique. |

- Chapitre 5 { **Le contrôle optimal corps-complet**
- Le calcul des consignes pour le contrôleur bas niveau d’un robot contrôlé en impédance.
 - La proposition d’un coût original dans l’OCP pour s’affranchir de l’utilisation de trajectoire précalculées pour les pieds de vol.

La première année de cette thèse a été effectuée au sein de l’entreprise Wandercraft [Wan]. Wandercraft conçoit et commercialise des exosquelettes médicaux de membres inférieurs. Dans ce cadre, nous avons étudié la stabilisation de la marche d’un exosquelette médical de membres inférieurs autoportant, c’est-à-dire sans utilisation de béquille. En exploitant les travaux mis au point sur les robots bipèdes [Car19a], nous avons développé le premier stabilisateur permettant une marche autonome sur l’exosquelette Atalante [Gur18b]. Le stabilisateur a aussi été validé avec des pilotes paraplégiques et a permis d’étendre largement la gamme de mouvements atteignables avec l’exosquelette. Nous présentons en annexe A un bref résumé de ces travaux qui ne s’inscrivent pas dans la problématique étudiée dans la suite de la thèse. Ce travail a donné lieu à la soumission d’un brevet portant sur le contrôle par admittance d’un exosquelette de membres inférieur avec flexibilités. Ce brevet est en cours d’examen.

Publications

Cette thèse a donné lieu à la publication des articles suivants :

Conférence avec actes

- Publié**
- ❖ Fanny RISBOURG, Thomas CORBÈRES, Pierre-Alexandre LÉZIART, Thomas FLAYOLS, Nicolas MANSARD et Steve TONNEAU. « Real time footstep planning and control of the Solo quadruped robot in 3D environments ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022
- Soumis**
- ↻ Alessandro ASSIRELLI, Fanny RISBOURG, Gianni LUNARDI, Thomas FLAYOLS et Nicolas MANSARD. « Whole-Body MPC without Foot References for the Locomotion of an Impedance-Controlled Robot ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2023 Soumis à *IEEE ICRA*, actuellement en cours d’examen.

Brevet

- Dépot**
- ↻ « Procédé de mise en mouvement de l’exosquelette ». Brev. FR2021/052335. Fanny RISBOURG, Gabriele BUONDONNO, Roch MOLLERO, Stanislas BROSSETTE et Guilhem BOÉRIS. 2021 Actuellement en cours d’examen.

Revue de Littérature

Contents

1.1	Contrôle des robots aveugles	10
1.1.1	Générateurs de Modèles Centraux	10
1.1.2	De l'utilité des modèles	11
1.1.3	Modèles réduits	12
1.1.4	Modèles complets	15
1.2	Des robots qui voient	19
1.2.1	Planification	20
1.2.2	Systèmes complets	21

Introduction

Ce premier chapitre présente les principales méthodes de contrôle qui ont été explorées et les résultats expérimentaux associés. Compte tenu de la grande variété des méthodes, la revue n'est pas exhaustive, mais se concentre sur les approches qui ont été appliquées au contrôle de robots quadrupèdes.

Le contrôle de robots *aveugles* sera d'abord abordé dans la Section 1.1. Nous y présenterons une première famille de contrôleurs inspirés de la biologie, les générateurs de modèles centraux (*Central Pattern Generator*). Les parties suivantes nous permettront d'explorer la diversité des approches fondées sur les modèles. Guidé par un fil chronologique, nous nous intéresserons aux modèles réduits, puis nous verrons comment l'augmentation des capacités de calcul des ordinateurs et le développement d'algorithmes de résolution numérique adaptés, ont permis aux contrôleurs de gagner en complexité et stabilité grâce à l'utilisation de modèles plus riches. La deuxième section 1.2 nous permettra de s'intéresser au contrôle des robots qui ont des informations sur la topologie du terrain qui les entoure Section 1.2. Après avoir brièvement présenté les avancées en termes de planification de contacts, nous verrons comment ces dernières années ont vu émerger des robots dotés d'une architecture complète de locomotion prenant en compte des informations visuelles pour calculer les mouvements de locomotion.

1.1 Contrôle des robots aveugles

La recherche en locomotion des quadrupèdes s'est d'abord intéressée à la génération de mouvements sur sol plat. La plupart des robots n'ont donc pour l'instant pas de capteurs ou d'algorithmes leur permettant d'obtenir des informations sur l'environnement (relief, présence d'obstacles, etc.). Des capteurs peuvent en revanche renseigner sur l'état du robot : vitesse et orientation de la base, contacts avec l'environnement. . . . Utiliser les mesures de ces capteurs pour adapter la commande permet d'obtenir des contrôleurs robustes à la fois aux erreurs de modèle (modèle du robot et de son environnement) et aux perturbations. Plusieurs approches ont permis d'obtenir de bons résultats en simulation et sur différents robots. Les contrôleurs les plus performants permettent aujourd'hui aux robots de se déplacer sur des terrains fortement accidentés, de monter et descendre des escaliers et résister à de fortes perturbations.

1.1.1 Générateurs de Modèles Centraux

Quand on voit un robot à pattes, on pense instinctivement à sa contrepartie biologique : l'homme pour un humanoïde, le chien ou le chat pour un quadrupède. . . . Pourtant, pour la conception des robots et de leurs contrôleurs, il n'y a que peu d'approches qui s'inspirent effectivement de la biologie [Ly11]. Ces approches sont inspirantes pour la robotique en général, mais cherchent aussi à valider artificiellement notre compréhension de la structuration de la locomotion naturelle, tant au niveau de la mécanique que du contrôle.

Les générateurs de modèles centraux (Central Pattern Generator - CPG), sont des réseaux de neurones qui peuvent produire de manière quasi-autonome des signaux périodiques. Ces signaux peuvent servir à générer des mouvements répétitifs comme la respiration, la nage ou la marche. Le Réseau Locomoteur Spinal est un CPG situé dans la moelle épinière et associé à la marche. Des signaux du cerveau et des retours sensoriels permettent de modifier les signaux [Fri06] émis. Ce sont ces réseaux qui expliquent par exemple pourquoi une poule peut continuer de courir pendant quelques secondes une fois sa tête coupée.

Les travaux de Taga [Tag94; Tag95] dans les années 1990 montrent que l'interaction entre des CPGs et un modèle musculo-squelettique humain permet de générer des mouvements de locomotion. D'abord testés sur une jambe robotique [Lew00], des premiers contrôleurs fondés sur des CPG sont rapidement mis en place sur des robots quadrupèdes [Tsu01]. Ils reposent sur l'interaction entre plusieurs oscillateurs non linéaires et permettent la génération de mouvements de marche avec adaptation de l'allure. Des signaux de capteurs de contact sont utilisés pour obtenir des informations sur l'état du robot et de l'environnement. Lewis montre en 2002 [Lew02] que des CPGs couplés à des réflexes basiques comme le contrôle de la posture et de l'extension des jambes en contact peuvent permettre d'apprendre en quelques minutes à un robot à trotter.

En utilisant des oscillateurs qui permettent un contrôle indépendant de la phase de contact et de la phase de vol, Righetti [Rig08] propose un CPG intégrant un retour sensoriel dont les propriétés peuvent être changées grâce à des paramètres simples pour obtenir différentes allures. D'abord développés dans l'espace articulaire les CPGs sont ensuite utilisés dans

l'espace cartésien [Liu11]. L'utilisation d'une cinématique inverse permet ensuite le calcul des commandes articulaires.

Pour améliorer la robustesse Barasuol [Bar13] propose l'ajout d'un stabilisateur au schéma de contrôle. Focchi [Foc13] enchaîne avec la proposition d'un nouveau réflexe. En parallèle, Ajallooeian [Aja13] reprend le concept de modèle virtuel, en anglais *Virtual Model Control* [Pra01], pour permettre de contrôler la direction du robot.

Plus récemment, Massi [Mas19] propose une architecture avec apprentissage, inspirée de la biologie. Un CPG est associé à un contrôleur *cérébral* qui contient un modèle de la dynamique inverse du robot et vient corriger les consignes de locomotion données par le CPG. Des algorithmes évolutifs sont utilisés pour apprendre les paramètres optimaux pour la locomotion.

Ces contrôleurs inspirés par l'observation de la nature peuvent aussi être combinés avec des méthodes d'optimisation. Ainsi Arena [Are21] utilise de la commande prédictive pour choisir un paramètre d'orientation qui sera transmis au CPG et contrôler ainsi la direction de la marche d'un quadrupède.

Le fonctionnement de la locomotion réflexe de l'homme et des animaux inspire donc aujourd'hui encore la conception de nombreux contrôleurs. Reproduisant le CPG à l'aide d'oscillateurs non linéaires, ils parviennent à générer différentes allures parfois très dynamiques [Zha21] pour des bipèdes et quadrupèdes. Des réflexes et stabilisateurs peuvent être ajoutés pour améliorer la robustesse et des méthodes d'apprentissage permettent d'optimiser les paramètres de la marche. Lors de tâches plus complexes où un mouvement réflexe ne suffit pas, des méthodes exploitant un modèle de la dynamique des robots permettent d'améliorer les performances.

Une direction voisine de ces travaux consiste à caractériser le mouvement naturel, en particulier humain, avec les mêmes modèles (mécaniques et de contrôle) que ceux utilisés dans cette thèse, comme le fait une partie de l'équipe Gepetto [Gep]. Ces travaux sont d'autant plus inspirants que le but initial de cette thèse était de participer à la restauration de la locomotion humaine chez des sujets handicapés [Ris]. Nous les citons donc pour rendre compte de cet intérêt.

1.1.2 De l'utilité des modèles

La force des CPGs est leur capacité à fonctionner sans impliquer de modèle ou de calculatoire compliquée. Mais cela nécessite un réglage soigneux du contrôleur et empêche de garantir d'une quelconque manière le fonctionnement du système, en particulier lorsqu'on le sort des conditions environnementales pour lesquelles il a été réglé. Au contraire, l'utilisation d'un modèle du fonctionnement du robot dans son environnement permet de choisir algorithmiquement la bonne réaction du système, en fonction de l'évolution qu'on sera alors capable de prédire. On appelle commande prédictive (Model Predictive Control - MPC) [Cam13] ce type d'approche, consistant à choisir le contrôle à chaque cycle comme l'optimum d'un problème de contrôle optimal sur un horizon partant de l'état estimé le plus probable à partir des dernières mesures des capteurs. La commande prédictive implique une optimisation d'un

problème qui sera d'autant plus grand que la complexité de la modélisation (noté N) et la durée de l'horizon prédictif (noté T) sont grands, suivant une loi en $O(N^3T)$.

Pour prédire le comportement futur du robot en fonction des commandes appliquées, il faut modéliser la dynamique du robot. Ils permettent notamment de relier l'évolution de la position du Centre de Masse - **CoM** - et du moment cinétique en fonction des forces de contact avec l'environnement, et découlent des équations du mouvement dites de Newton et Euler, que l'on détaillera Section 2.1. Deux contraintes limitent toutefois les mouvements possibles :

- Les forces sont unilatérales (le robot ne peut que *pousser* sur le sol) et contraintes par les limites du frottement
- Les contacts sont binaires (il y a contact ou il n'y a pas de contact) et des impacts peuvent se produire à la création d'un contact.

Il existe donc des états du robots pour lesquels il sera impossible d'appliquer des forces permettant d'éviter une chute. Ces états sont dits non *viables*. Un état est dit *capturable* s'il existe un contrôle qui permet d'arriver à un équilibre stable depuis cet état. L'étude des modèles a souvent été faite sous l'angle de la *capturabilité* et *viabilité* [Wie02 ; Wie16 ; Pra06b].

1.1.3 Modèles réduits

Des modèles réduits ont été étudiés pour réduire la complexité N . Ils permettent de modéliser la dynamique du centre de masse ou d'autre grandeurs cinématiques caractéristiques du mouvement. Des approximations peuvent être utilisées pour linéariser les équations.

Une première approche consistant à décomposer le contrôle sera d'abord présentée : une première partie choisit des consignes cinématiques offrant des garanties de stabilité et une deuxième partie calcule les commandes articulaires permettant de réaliser ces consignes cinématiques. Exploitant des modèles souvent linéaires de la dynamique, elle a permis d'obtenir de premières marches stables sur des terrains irréguliers. Une deuxième partie montrera comment un contrôle plus direct des forces de contact permet d'obtenir des comportements plus dynamiques et robustes.

1.1.3.1 Suivi de consignes cinématiques

Lors d'une marche sur sol plat, les forces peuvent être réduites au point de moment zéro (Zero Moment Point - **ZMP**) [Vuk04] ou Centre de pression (Center of Pressure - **CoP**). L'unilatéralité des forces de contact a pour conséquence d'impliquer l'appartenance du **ZMP** au polygone de support [Wie02]. Cette contrainte est alors utilisée pour caractériser une marche dynamiquement stable [Kaj03a] (par opposition à des caractérisations plus réductrices comme les marches dites "statiques", où on impose de manière conservatrice au **CoM** de rester dans le polygone de support, permettant d'obtenir la contrainte précédente en négligeant la vitesse [DP16]). Contrôler le **ZMP** a donc été une stratégie populaire pour arriver à une marche stable. Un modèle linéaire de la dynamique, le *pendule inverse linéarisé* (Linear Inverted Pendulum Model - **LIPM**) permet de calculer facilement le **ZMP** à partir du Centre de masse (Center of Mass -**CoM**).

Kajita [Kaj03a] propose en 2003 un générateur de mouvements pour le robot humanoïde HRP-2 reposant sur le contrôle du ZMP et le LIPM. Wieber [Wie06a] montre que le contrôleur de Kajita peut être vu comme régulant le *jerk* (dérivée de l'accélération) du CoM. Il augmente ce contrôleur en utilisant de la commande prédictive (Model Predictive Control - MPC) [Cam13; Her10a]. Le MPC consiste à résoudre successivement des problèmes de contrôle optimal (Optimal Control Problem - OCP) à temps fini sur un horizon fuyant. Les problèmes de contrôle optimal sont initialisés à partir des dernières estimations d'état et le contrôle est déduit de la politique optimale du dernier OCP calculé. Ce schéma permet notamment de choisir des coûts à minimiser et éventuellement des contraintes à respecter. Les MPC ont par la suite été largement utilisés avec des modèles réduits [Dim09]. D'autres variables peuvent être optimisées par l'OCP, comme la position des pieds par exemple [Die08; Her10a; Her10b]. Le développement de solveurs numériques adaptés a ensuite permis de résoudre des OCP non linéaires, par exemple prenant en compte des contraintes d'évitement d'obstacles [Nav17].

Ces MPC génèrent des trajectoires pour le CoM et éventuellement les positions des contacts. Une cinématique inverse (Inverse Kinematics - IK) est ensuite utilisée pour obtenir des commandes articulaires permettant de suivre ces trajectoires [Kaj03b]. Les trajectoires des pieds de vol sont en général définies par des polynômes ou courbes de Bézier bien choisis. Un problème quadratique (Quadratic Problem - QP) peut-être utilisé pour profiter de la redondance des robots et ajouter des contraintes sur la dynamique, les forces ou couples [Fuj98]. La résolution successive de plusieurs QPs peut permettre de hiérarchiser les tâches à effectuer [Esc10; Esc14].

Outre le CoM et le ZMP, d'autres critères de stabilité ont été étudiés. Un autre critère provenant du modèle du pendule inverse linéarisé a permis d'obtenir des marches stables pour bipèdes sur des escaliers [Car19a] : l'analyse du LIPM permet d'isoler la partie divergente de la dynamique, que l'on appelle partie divergente du mouvement (Divergent Component of Motion - DCM) ou point de capture (capture point) [Koo12]. Un contrôle linéaire en boucle fermée de ce DCM peut être utilisé pour contrôler l'équilibre de la marche [Eng11]. Le contrôle de DCM permet de remplacer le contrôle de la position et la vitesse du CoM dans la cinématique inverse, ce qui permet un réglage plus aisé des gains [Car19a].

Cette approche visant à calculer des consignes cinématiques puis des consignes articulaires permettant de les réaliser est appelée *artificial synergy synthesis* [Wie16]. Elle a permis le contrôle de plusieurs quadrupèdes sur des terrains variés [Bel18b; Mas17; DC18b]. Pour palier aux limites du LIPM, qui implique de nombreuses hypothèses réductrices le rendant en particulier difficile à utiliser ailleurs que sur sol plat, d'autres modèles et critères de stabilité sont étudiés, comme le cône de frottement 6D (Contact Wrench Cone) [Liu12; Car15a; Dai16] qui généralise le ZMP sur sol non plat. Le modèle centroïdal et le modèle inertiel rigide (présentés Section 2.1) s'affranchissent ainsi de certaines hypothèses de LIPM et ont été largement utilisés pour le contrôle des quadrupèdes.

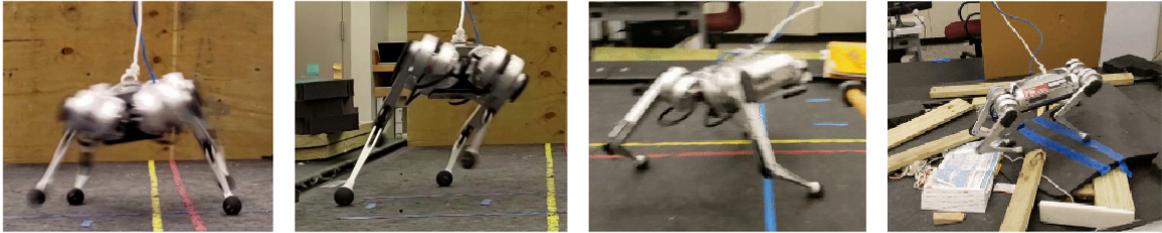


FIGURE 4 – MPC centroïdal pour le contrôle du robot quadrupède Mini-Cheetah [Ble20]

1.1.3.2 Contrôle des forces

Le rôle prépondérant des forces de contact avec l’environnement pour contrôler le CoM est mis en évidence par les équations de Newton et Euler. Des stratégies ont donc été développées pour jouer plus directement sur ces forces. Des forces de référence peuvent être calculées à partir du modèle (LIPM, inertiel rigide, centroïdal. . .) et différentes approches permettent de les suivre.

La première consiste à utiliser un stabilisateur en complément d’un contrôle du ZMP ou du DCM . Le générateur de marche génère des consignes de force ou de ZMP , qui ne peuvent être suivies directement sur des robots contrôlés en position. On utilise alors de l’admittance pour remédier à ce problème. Cela consiste à adapter les consignes de position envoyées au robot à partir d’une erreur en force. Les contrôleurs de la littérature utilisent de l’admittance au niveau des points de contact [Kaj10 ; Kim09 ; Bou18] ou au niveau du CoM [Sug02 ; Eng12]. Ces deux stratégies peuvent aussi être utilisées de manière simultanée [Car19a].

Une seconde approche consiste à adapter le contrôleur corps-complet pour prendre en compte les forces à suivre. Dans un MPC on peut calculer les trajectoires du CoM ainsi que des forces de contact. Un contrôleur corps-complet qui ne suit plus seulement les trajectoires du CoM , mais les forces de contact choisies par le MPC est utilisé pour obtenir les consignes articulaires [Kim19 ; Léz21]. Ces premiers travaux ont utilisé le modèle inertiel rigide, qui rend la dynamique centroïdale linéaire grâce à quelques hypothèses, permettant son utilisation dans un MPC. Les résultats expérimentaux sur les robots quadrupèdes Mini-Cheetah (voir Fig. 4) et Solo montrent une agilité permettant d’évoluer sur des terrains variés et de suivre une vitesse de référence. Le modèle inertiel rigide, qui réduit la dynamique à celle de la base du robot, est en effet particulièrement adapté au contrôle des robots quadrupèdes, qui ont des pattes légères et dont les mouvements de rotation de la base ont besoin d’être contrôlés.

Plus récemment, le développement de nouveaux solveurs numériques adaptés à des problèmes non-linéaires ont permis d’utiliser un MPC avec la dynamique centroïdale complète (sans les hypothèses de linéarisation) [Cor21]. Une autre technique a été utilisée pour s’affranchir de cette linéarisation : l’utilisation d’heuristiques a permis de faciliter la résolution en régularisant les trajectoires autour de comportements nominaux [Ble20].

Ces modèles réduits ne permettent toutefois pas de contrôler les positions des pieds ou d’un corps particulier du robot, qui doit être faite en dehors du MPC ou calculée à la main.

Pour optimiser les positions des pieds en contact des méthodes ont été proposées, qui utilisent des variables d'état supplémentaires dans l'OCP [Ble19; Cor21]. Pas de réelle amélioration dans la qualité de mouvements obtenus [Léz22] est toutefois observée.

Le contrôle des forces a ainsi permis d'obtenir des marches robustes sur une grande variété de terrains sans connaissance de l'environnement. Mais seules les forces de contact et l'état du robot défini par le modèle peuvent être optimisés directement. Les travaux effectués pendant cette thèse s'inspirent de ces modèles qui font aujourd'hui autorité pour le contrôle des quadrupèdes, mais vont tenter de palier à leurs limitations pour améliorer la robustesse du contrôle, notamment en optimisant les positions des contacts ou les durées des pas.

1.1.4 Modèles complets

Utiliser la dynamique complète du robot dans un OCP permet la génération de mouvements plus riches et versatiles. Cette dynamique non linéaire et hybride est cependant coûteuse à calculer et plusieurs stratégies sont aujourd'hui utilisées, pour optimiser des trajectoires ou des politiques :

- Un calcul hors-ligne d'un ensemble de trajectoires optimales qui sont ensuite obtenues en-ligne grâce à une *librairie de trajectoires*
- L'utilisation en ligne d'un MPC corps-complet
- L'utilisation d'apprentissage pour calculer une politique de contrôle optimale

Ces approches sont présentées dans cette section ainsi que des résultats expérimentaux associés.

1.1.4.1 Librairie de trajectoires

La génération hors-ligne de trajectoires permet de générer des trajectoires corps-complet respectant des critères en termes de cyclicité et viabilité, sans être limité par le temps de calcul. Mais suivre sur le robot une trajectoire de référence calculée hors ligne n'est pas toujours possible à cause d'erreurs de modèle ou de perturbations extérieures. Une solution consiste à générer une base de données comprenant de nombreuses trajectoires de référence, puis à choisir une trajectoire en ligne en fonction de l'état du robot [Wie06b]. Chaque trajectoire de la base de donnée est la solution d'un OCP. Plusieurs méthodes existent pour obtenir la trajectoire de référence en fonction de l'état courant du robot : interpolation [Wie06b; Guo21], plus proche voisin [Atk92; Liu13]...

La méthode la plus largement utilisée pour la génération de telles trajectoires est appelée dynamique zéro hybride (Hybrid Zero Dynamics - HZD). L'observation de marcheurs passifs [McG90] (voir Fig. 5) a montré l'existence de cycles asymptotiques périodiques de marche et a inspiré Chevallereau et Grizzle [Wes18a] à rechercher des outils permettant de générer des trajectoires économes en énergie et asymptotiquement stables pour des robots sous-actionnés (comme le sont les bipèdes ou quadrupèdes avec des pieds ponctuels) [Wes18b]. La HZD définit ainsi une trajectoire à travers un ensemble de *contraintes virtuelles* articulaires imposées grâce à un contrôle à retour d'état. Les contraintes virtuelles sont choisies pour que la dynamique

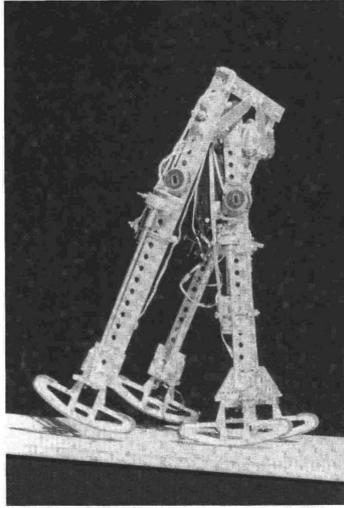


FIGURE 5 – Marcheur passif de Tad McGeer

du système réduit qui respecte ces contraintes converge asymptotiquement vers une orbite périodique respectant certaines propriétés.

La librairie de trajectoire générée dépend de la tâche à accomplir. Dans le cas d'une récupération de chute, Liu propose [Liu09] de générer une grille de trajectoires optimales correspondant à des vitesses initiales de la base. Pour une marche avec un contrôle de la vitesse, Da [Da16] génère des trajectoires HZD pour différentes vitesses et les interpole pour avoir la vitesse de référence souhaitée. La même méthode appliquée au robot Cassie donne des résultats expérimentaux impressionnants [Gon19b].

Les trajectoires optimales ne sont pas toutes utilisées de la même façon dans le contrôleur. Nguyen [Ngu20] utilise un QP en ligne pour adapter la trajectoire de référence provenant de la librairie. Il peut ainsi imposer le respect de contraintes, sur le placement des pieds par exemple. La même approche est utilisée par [Guo21] qui propose un synthétiseur de trajectoire qui peut remplacer en ligne un MPC : la trajectoire optimale obtenue par interpolation est modifiée pour respecter des contraintes cinématiques puis suivie jusqu'au prochain appel à la librairie de trajectoires. La trajectoire de référence peut aussi servir à démarrer à chaud le solveur numérique qui résout en ligne un OCP [Liu09 ; Liu13]. Elle facilite la convergence et permet d'avoir des trajectoires optimales.

Cette méthode, d'abord développée sur des bipèdes, a ensuite été testée avec succès pour le contrôle de quadrupèdes [Ma19 ; Liu15]. Elle est toutefois limitée par la combinatoire associée à la génération de la librairie de trajectoire : la dimension de l'état initial du robot et des consignes est trop grand pour en faire l'exploration.

1.1.4.2 MPC Corps-complet

L'augmentation des capacités de calcul des ordinateurs récents ainsi que l'utilisation de solveurs numériques plus adaptés à la dynamique non linéaire et hybride des robots ouvrent

des nouvelles portes et permettent de s'affranchir des limites liées à la combinatoire de la génération d'une librairie de trajectoires. En effet, de premiers travaux parviennent à utiliser en ligne un MPC corps-complet : L'OCP utilise la dynamique complète du robot pour trouver à partir de l'état courant une trajectoire optimale utilisant tout le corps du robot.

L'intérêt de ce type de contrôleur était déjà pressenti : il permet de calculer de manière réactive une trajectoire optimale utilisant la dynamique complète du robot pour réaliser une ou plusieurs tâches. De premières études en simulation présentent une variété de mouvements réalisables sur un robot bipède [Ere15] avec ce type de contrôle.

En 2015, Koenemann [Koe15] propose une première implémentation sur un robot réel : l'humanoïde HRP-2. Ce contrôleur est testé pour des tâches du haut du corps (suivi de cible avec une main, contact avec une table. . .) associée à un maintien de l'équilibre. Le retard induit par la durée d'optimisation est pris en compte, mais sur le robot, il crée des discontinuités dans le contrôle telles qu'un retour d'état n'est pas possible. C'est donc l'état de référence qui est utilisé pour initialiser l'OCP, ce qui limite grandement la robustesse du contrôleur. Neunert [Neu18] propose un premier MPC corps-complet utilisé avec succès pour de la marche. Il réalise des expériences avec deux robots quadrupèdes, HyQ et ANYmal. L'OCP utilise un modèle complet des contacts qui lui permet d'optimiser la position et durée des contacts. Un modèle virtuel est utilisé en plus du MPC pour améliorer les performances.

Ces différents travaux résolvent l'OCP non linéaire en construisant et résolvant itérativement des problèmes LQR. C'est le principe de la programmation différentielle dynamique (Differential Dynamic Programming - DDP) avec une approximation de Gauss-Newton. L'algorithme du DDP est adapté dans la librairie Crocoddyl pour permettre la génération de mouvements plus dynamiques [Mas20] sans avoir besoin de fournir une supposition initiale (*initial guess*) faisable. Cette nouvelle formulation est utilisée sur le robot humanoïde Talos pour des tâches de suivi de cible et de locomotion [Dan21 ; Dan22b].

Plusieurs OCP sont parfois résolus en cascade : Budhiraja [Bud19] puis Meduri [Med22] utilisent d'abord un OCP centroïdal pour trouver les trajectoires des degrés de liberté sous-actionnés du robot, puis un OCP corps-complet prend cette trajectoire comme référence pour générer les consignes articulaires. Des expériences sur le robot quadrupède Solo [Gri19b] montrent un comportement très dynamique lors de tâches de locomotion ou de sauts. Katayama propose d'utiliser un deuxième OCP pour optimiser les instants des changements de contact [Kat22]. Il effectue des expériences de sauts sur le robot quadrupède Unitree A1.

Ces différents contrôleurs ont été testés principalement sur des robots contrôlés en couple [Neu18 ; Dan21 ; Kat22]. Les délais induits par le temps de calcul de la trajectoire optimale sont en général de l'ordre de 10 ms alors que le contrôle des moteurs se fait à une fréquence proche de 1 kHz. Les gains de Ricatti ou *feedback gains* du solveur DDP sont donc utilisés pour obtenir les consignes de couple à partir de l'état courant et la dernière politique optimale [Koe15 ; Dan22a].

L'amélioration des temps de calcul permet aujourd'hui d'optimiser en ligne des mouvements du corps complet du robot. Cette approche permet de générer des mouvements

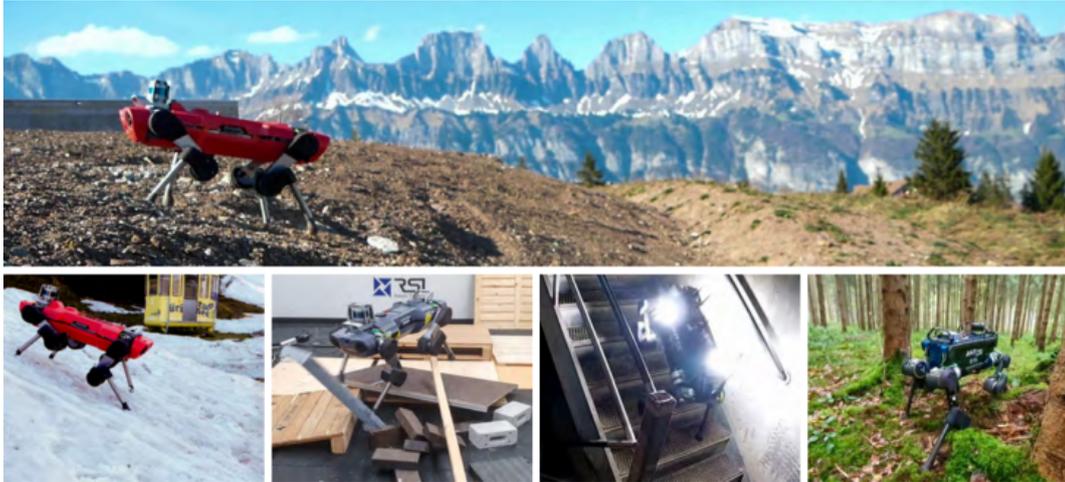


FIGURE 6 – Contrôle par apprentissage par renforcement du robot ANYmal [Lee20]

très dynamiques pour des tâches variées (suivi de cible, marche, sauts...). La différence de fréquence entre le contrôle et la génération des trajectoires optimales, ainsi que les retards engendrés, posent toutefois des problèmes dont l'étude est aujourd'hui réduite à un empirisme expert mal documenté. Cette thèse étudie certaines de ces limitations pour proposer des solutions permettant l'utilisation de modèles complets dans un MPC sur tout type de robot, remédiant ainsi aux principales limites des modèles réduits.

1.1.4.3 Optimisation de politique par apprentissage

L'apprentissage est aujourd'hui utilisé dans de nombreux domaines et la robotique ne fait pas exception. En complément de modèles physiques ou non, des méthodes fondées sur les données ont montré de bons résultats pour le contrôle de robots à pattes. Si ces approches connaissent un fort engouement aujourd'hui, elles ne sont toutefois pas encore matures et restent limitées par le manque de compréhension du fonctionnement des politiques obtenues, ainsi que par le manque de garanties qui en découle. Cette thèse ne s'est donc pas intéressée à ces approches, mais compte tenu des bons résultats expérimentaux récents, un rapide aperçu des différentes méthodes est présenté dans cette section.

Le principal problème à résoudre pour utiliser ce type d'approche, est de générer une quantité de données suffisante. S'il est possible d'utiliser des données générées lors d'expériences avec le robot [Yan22], le processus demande une intervention humaine fréquente et un temps conséquent. Les simulateurs sont donc en général préférés. Ils permettent de générer beaucoup de données en un temps limité, mais les contrôleurs appris avec des données de simulation ne sont pas forcément robustes sur un robot réel [Iba21]. Ce problème de transfert, communément appelé *sim-to-real*, est largement étudié [Pen18; Tan18]. Pour remédier aux différences entre simulation et modèle, des modélisations poussées des actionneurs sont généralement introduites dans les simulations [Geh16].

L'apprentissage a d'abord été utilisé en complément d'une approche fondée sur les modèles.

Mansard [Man18] *démarre à chaud* un MPC non linéaire grâce à un modèle appris, pour en faciliter la convergence. Ponton apprend une politique pour le contrôle en impédance d’un robot quadrupède [Pon14]. Le contrôleur est utilisé avec une cinématique inverse qui suit des trajectoires de pieds. L’apprentissage est utilisé pour améliorer la robustesse et l’efficacité énergétique du contrôleur, mais n’est testé qu’en simulation. De la même manière, les paramètres d’un contrôleur en boucle fermée sont optimisés dans un processus d’apprentissage par la pratique dans [Geh16]. Cette fois-ci, l’apprentissage se fait dans une simulation comprenant un modèle des contacts et des actionneurs suffisamment fidèles pour que l’application sur le robot réel puisse se faire directement. Dans une autre veine, Xie [Xie21] utilise de l’apprentissage par renforcement pour contrôler les accélérations d’un modèle centroïdal, dont on déduit les forces de contact à appliquer. Ce contrôleur est testé dans différents scénarios avec le robot Unitree A1.

D’autres travaux se passent complètement d’un modèle physique du robot. Utilisant des positions pour les pieds prédéfinies, des contrôleurs corps-complet entraînés par apprentissage par renforcement en simulation [Tso20 ; Gan21] permettent au robot ANYmal de marcher sur des terrains irréguliers. D’autres algorithmes n’utilisent pas d’information sur les contacts, qui sont uniquement déterminés par le mouvement corps-complet du robot [Sch17 ; Fuj18 ; Haa18 ; Lee20]. Ces contrôleurs peuvent nécessiter un temps d’apprentissage de plusieurs jours, mais sont robustes dans une grande variété d’environnements (voir Fig. 6).

Les méthodes fondées sur de l’apprentissage ont ainsi permis d’obtenir des marches dynamiques et stables dans une grande diversité d’environnements avec des robots bipèdes ou quadrupèdes. Mais les temps d’apprentissage sont très longs et nécessitent des simulateurs performants avec un modèle fidèle de la dynamique et des actionneurs.

1.2 Des robots qui voient

Si les contrôleurs *aveugles* permettent à des robots quadrupèdes de se déplacer sur une grande variété de terrains, ils sont incapables de s’adapter à des environnements avec des obstacles infranchissables ou des trous par exemple. C’est en général un opérateur humain qui choisit l’itinéraire emprunté par le robot pour éviter de tels écueils [Léz21]. C’est en utilisant des informations sur le terrain qu’un robot va pouvoir s’y déplacer en autonomie en choisissant de lui-même quel itinéraire lui permet d’atteindre un objectif. Plusieurs étapes sont généralement utilisées pour mettre en place ce type de contrôle :

- Plusieurs types de capteurs peuvent être utilisés pour obtenir des informations sur le monde qui nécessitent des traitements pour les rendre utilisables dans un contrôleur.
- La **cartographie** consiste à faire des cartes de l’environnement à partir des données de ces capteurs [Dav07 ; Doh17].
- La **localisation** permet de situer le robot dans cet environnement [Del99 ; Las21]
- La **navigation** [Zhe21] et la **planification** s’intéressent au choix de l’itinéraire et des emplacements des pas.

Chacune de ces différentes tâches est un domaine de recherche à part entière. Cette thèse aborde uniquement le domaine de la planification, aussi nous nous intéresserons uniquement

à la planification dans ce chapitre, avant de présenter les contrôleurs mettant en place un schéma complet de locomotion avec vision.

1.2.1 Planification

La planification consiste à choisir la séquence des futurs points de contact avec l’environnement. Les points de contacts choisis doivent être réalisables par le robot et lui permettre d’évoluer dans l’environnement sans tomber. La question de faisabilité est cruciale pour que le contrôleur corps-complet qui utilise ce plan de contact puisse le réaliser.

Pour prouver la faisabilité d’un plan de contact, trouver une trajectoire corps-complet qui le réalise est une méthode couteuse. Une méthode plus efficace consiste à utiliser un modèle simplifié du robot pour trouver une trajectoire en équilibre quasi-statique qui relie les contacts [DP16]. Ce critère de faisabilité est toutefois très limitant et un critère de faisabilité dynamique exploitant le modèle centroïdal a été proposé [Fer20].

L’environnement est souvent représenté comme un ensemble de surfaces potentielles, rendant le problème discret. De nombreuses approches peuvent être utilisées pour le résoudre. La plus répandue est de résoudre un problème d’optimisation mixte-entier [Dei14]. Des contraintes garantissent la faisabilité cinématique du plan et les approximations convexes permettent de trouver un optimum global au problème. La dynamique centroïdale peut aussi être prise en compte [Pon16; AC17]. Tonneau propose une relaxation du problème reposant sur une minimisation de la norme L1 pour éviter de manipuler des variables entières [Ton20]. Cette méthode consiste à résoudre dans un premier temps le problème de faisabilité en choisissant des surfaces de contact, puis à optimiser les positions des pas dans ces surfaces.

D’autres approches procèdent par échantillonnage. Une méthode fondée sur des potentiels est utilisée par Escande [Esc13] pour explorer différentes configurations de contact obtenues à l’aide d’une cinématique inverse. Une recherche de graphe peut être réalisée avec un algorithme de type A^* [Che07; Lin18; Gri19a]. De ce cas, chaque état du robot est associé à un ensemble d’actions possibles, qui respectent des critères de faisabilité et mènent à un nouvel état. Le niveau de discrétisation utilisé pour les actions réalisables impacte directement la taille du graphe généré et donc le temps de résolution. Lin [Lin19] combine un apprentissage de l’évolution de la dynamique centroïdale avec une recherche de graphe, ce qui permet d’obtenir des garanties de stabilité dynamiques et non plus quasi-statiques.

Pour faciliter le problème de planification, une trajectoire guide de la base du robot peut être calculée avant de trouver une séquence de contacts le long de cette trajectoire [Bou09; Ton18a; Car16b; Son21]. Le problème ainsi décomposé en deux sous-problèmes de plus faible dimension (voir Fig. 7) est résolu suffisamment rapidement pour être utilisé en ligne. Des méthodes d’apprentissage peuvent être utilisées pour calculer la trajectoire guide ou les contacts dans un environnement complexe [Che21; Wel21].

Ainsi une grande variété d’approches permettent de résoudre le problème de planification. Mais la combinatoire associée à la discrétisation et l’exploration de l’environnement rend les approches souvent trop lentes pour être utilisées en ligne. Considérer un plus petit espace

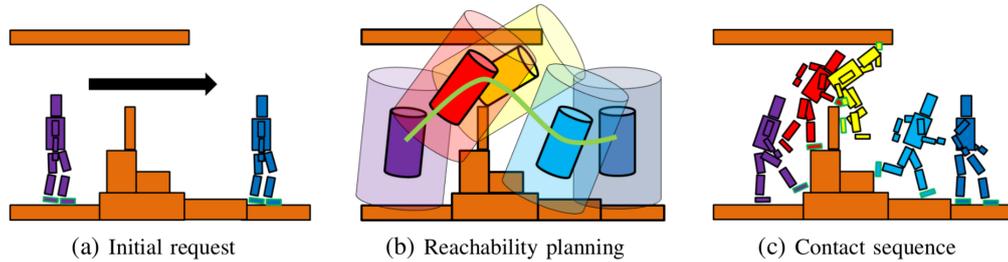


FIGURE 7 – Décomposition du problème de planification en deux sous-problèmes [Car16b]

d’exploration ou une décomposition du problème permet de diminuer les temps de calcul. Une des contributions de cette thèse est la formulation d’un tel contrôleur, qui utilise de la planification en temps réel pour choisir le plan de contact, l’adapte à fréquence élevée avant de résoudre un MPC. Un guide est utilisé pour ne considérer que les surfaces potentielles atteignables à chaque pas et l’horizon de la planification est suffisamment court pour permettre de calculer un nouveau plan de contact en ligne à chaque pas, donc à une fréquence de 10 Hz environ.

1.2.2 Systèmes complets

Beaucoup des contributions passées se sont focalisées sur un des aspects (planification ou perception). Des compétitions comme le DARPA Subterranean Challenge encouragent toutefois le développement de robots équipés de vision [Pet22], qui bénéficie de la maturité récente des robots quadrupèdes tout en y contribuant. Les épreuves du DARPA consistent à explorer des environnements contraints, sans GPS, et où les communications radios sont limitées. Les robots doivent trouver des objets et les localiser. Si la majorité des équipes utilisent des robots aériens [Pet22], ces dernières années ont vu arriver des robots à pattes. L’équipe CTU-CRAS-NORLAB utilise ainsi un robot hexapode [Ciz16], l’équipe CSIRO le robot quadrupède Spot de Boston Dynamics [Tam21] et l’équipe CERBERUS qui a gagné l’édition 2022 du Challenge utilise le quadrupède ANYmal [Tra22]. La robustesse et polyvalence des contrôleurs embarqués sur ces robots montrent les progrès récents dans le contrôle des robots à pattes.

Plusieurs approches permettent de combiner vision et locomotion. Elles utilisent généralement du contrôle fondé sur modèle, combiné éventuellement avec de l’apprentissage. Melon [Mel21] utilise ainsi les capteurs d’ANYmal pour générer une carte d’élévation. Cette carte d’élévation est utilisée pour la génération de la trajectoire, qui est optimisée pour un modèle réduit [Win18] en même temps que le plan de contact et les durées des pas. Pour accélérer la résolution du problème d’optimisation un démarrage à chaud est réalisé grâce à un modèle appris. L’horizon de quelques secondes permet d’utiliser le MPC à une fréquence de quelques Hertz, et les retards liés au temps de calcul de la trajectoire sont pris en compte. Gangapurwala [Gan22] découple la planification et l’optimisation de trajectoire. La planification est faite par apprentissage par renforcement à partir de la carte d’élévation et de l’état courant du robot. La trajectoire du modèle réduit est ensuite optimisée pour ce plan de contact suivant la méthode

proposée par [Bel18b]. Une correction sur les couples envoyés aux moteurs vient finalement compenser les erreurs de modèle et les retards qui rendaient difficile le suivi de trajectoire par le contrôleur corps-complet. Jenelten [Jen22] génère deux cartes d’élévation à partir des capteurs extéroceptifs : la première est une carte d’élévation standard. Elle est utilisée pour choisir les placements des pieds dans un problème d’optimisation sur modèle réduit. La deuxième est un *sol virtuel*, obtenu par filtrage de la première carte. Il permet de choisir une position de référence pour la base du robot. Les hypothèses utilisées dans l’optimisation de trajectoire limitent toutefois les performances de ce contrôleur. Dans le travail de Grandia [Gra22], l’accent est mis sur le traitement de la carte de niveau pour obtenir des primitives, comme un champ de distance et une segmentation des plans. Ces primitives sont utilisées dans la génération de trajectoire pour faciliter la planification, notamment par l’introduction d’une contrainte convexe pour la position des contacts et une contrainte d’évitement d’obstacles. Des expériences sur différents types d’environnements sont réalisées avec le robot ANYmal.

L’utilisation d’apprentissage par renforcement permet d’éviter les découplages et traitements des données capteurs, et de s’affranchir des problèmes de temps de calcul et de modélisation. Un premier contrôleur entièrement fondé sur de l’apprentissage est proposé par Rudin [Rud22], mais les résultats expérimentaux se limitent à des terrains moins complexes que les contrôleurs précédents.

Ainsi quelques travaux proposent aujourd’hui des contrôleurs permettant de se déplacer dans des terrains complexes grâce à l’utilisation de capteurs extéroceptifs. Les temps de calcul pour les contrôleurs fondés sur des modèles et la robustesse posent toutefois encore problème et des chutes ne sont pas rares lors de l’exploration des terrains du DARPA SubT Challenge. Le contrôleur avec planification réactive proposé dans cette thèse a pour vocation d’être utilisé dans un schéma de locomotion complet comprenant vision, localisation et planification. C’est ce que Thomas Corbères a réalisé sur le robot ANYmal en reprenant nos travaux.

Conclusion

Les difficultés liées à la locomotion des robots à pattes ne sont pas encore résolues. Si les approches fondées sur les modèles permettent aujourd’hui d’obtenir d’excellents résultats expérimentaux, les temps de calculs sont cruciaux et limitent les possibilités en termes de complexité du modèle utilisé et durée de l’horizon du problème d’optimisation. Les approches fondées sur les données se développent rapidement et commencent à montrer de bons résultats expérimentaux, malgré des temps d’entraînements très longs. Elles ne peuvent toutefois rien nous apprendre sur la locomotion animale et les principes qui la régissent.

Cette thèse s’est intéressée aux limites des approches fondées sur les modèles. Dans une première partie, nous sommes partis des contrôleurs reposant sur un MPC avec modèle réduit de la littérature que nous avons augmenté pour donner plus de liberté au contrôleur dans le choix des positions des pas et de leur durée. Dans un second temps, nous avons proposé un contrôleur utilisant un modèle complet de la dynamique et un coût astucieux pour optimiser le mouvement du robot en se passant complètement d’heuristiques et pré-calculs.

Les outils du roboticien

Contents

2.1	Modélisation	24
2.1.1	Centroïdale	24
2.1.2	Inertielle rigide	25
2.1.3	Corps-complet	26
2.1.4	Contacts	26
2.2	Contrôle Optimal	28
2.2.1	Présentation	29
2.2.2	Problème discret	30
2.2.3	Programmation dynamique	31
2.2.4	DDP	32
2.2.5	Programmation Différentielle Dynamique Cherchant la Faisabilité	33
2.3	Le robot quadrupède Solo	34
2.3.1	Open Dynamic Robot Initiative	34
2.3.2	Contrôleur	35

Introduction

De nombreux outils et compétences sont utilisées par les roboticiens, suivant le domaine de recherche étudié et l'approche choisie. Cette thèse a contribué à l'étude du contrôle fondé sur les modèles des robots à pattes. Différents modèles ont été utilisés pour contrôler le robot par commande prédictive. Les fondements théoriques de ces modèles et des outils utilisés seront présentés dans cette partie. Les différentes modélisations utilisées seront d'abord présentées. Nous verrons ensuite la formulation générale et quelques méthodes pour la résolution numérique d'un problème de contrôle optimal. Finalement nous parlerons du robot quadrupède Solo qui a été utilisé comme cadre expérimental.

2.1 Modélisation

Pour représenter la dynamique hybride des robots, nous avons vu que plusieurs modèles existent. Nous allons présenter deux modèles utilisés dans cette thèse. Le modèle centroïdal, qui s'intéresse uniquement à l'évolution de la position du CoM et du moment cinétique en fonction des forces de contact, et le modèle complet, qui prend en compte la dynamique de chacune des articulations du robot, qu'elle soit actionnée ou non.

2.1.1 Centroïdale

La dynamique centroïdale est régie par les équations de Newton et Euler. Ce n'est pas une approximation de la dynamique complète. Elle ne décrit en revanche que l'évolution du CoM et du moment cinétique du robot. L'équation de Newton s'intéresse à l'évolution de la position du CoM, \mathbf{c} , et peut s'écrire de la manière suivante :

$$m \ddot{\mathbf{c}} = \sum_{i=1}^{n_c} \mathbf{f}_i + m \mathbf{g} \quad (2.1)$$

où m est la masse du robot, \mathbf{f}_i la force au $i^{\text{ème}}$ contact avec l'environnement et \mathbf{g} l'accélération de gravité. $\mathbf{f} = [\mathbf{f}_1, \dots, \mathbf{f}_{n_c}]^T$ regroupe les n_c forces de contact avec l'environnement.

L'équation d'Euler décrit quant à elle les variations du moment cinétique \mathbf{L} du robot :

$$\dot{\mathbf{L}} = \sum_i (\mathbf{r}_i - \mathbf{c}) \wedge \mathbf{f}_i \quad (2.2)$$

où \mathbf{r}_i est la position du $i^{\text{ème}}$ contact. Toutes les quantités sont exprimées dans le repère monde.

Le moment cinétique s'exprime de la manière suivante :

$$\mathbf{L} = \sum_k (\mathbf{p}_k - \mathbf{c}) \wedge m_k \dot{\mathbf{p}}_k + \mathbf{I}_k \omega_k \quad (2.3)$$

avec \mathbf{p}_k les positions et vitesses de rotation du $k^{\text{ème}}$ corps qui compose le robot, m_k et \mathbf{I}_k leurs masses et matrices d'inerties respectives (exprimées dans les coordonnées globales) et ω_k leur vitesse angulaire.

Ces équations permettent de remarquer que sans changement des points de contact, la dynamique est continue. En revanche chaque création ou suppression de contact crée une discontinuité : on dit que la dynamique est *hybride* [Che13].

2.1.2 Inertielle rigide

Pour les robots quadrupèdes dont les pattes sont légères par rapport au corps (appelé base), on peut considérer les approximations suivantes :

- Le CoM est confondu avec le centre de la base du robot \mathbf{p} ;
- Les masses et inerties des pattes peuvent être négligées devant l'inertie de la base.

Ces simplifications permettent d'obtenir les équations suivantes :

$$m \ddot{\mathbf{p}} = \sum_i \mathbf{f}_i + m \mathbf{g} \quad (2.4)$$

$$\dot{\mathbf{L}} = \frac{d}{dt}(\mathbf{I}\boldsymbol{\omega}) \quad (2.5)$$

avec \mathbf{I} la matrice d'inertie de la base du robot dans le repère global, et $\boldsymbol{\omega}$ la vitesse angulaire de la base.

Soit \mathbf{x} l'état du robot, composé de la position \mathbf{p} de la base, de son orientation $\Theta = [\theta, \phi, \psi]^T$ (exprimée avec les angles d'Euler) et des vitesses de translation \mathbf{v} et de rotation $\boldsymbol{\omega}$ de la base :

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \Theta \\ \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \quad (2.6)$$

Quelques hypothèses permettent de rendre cette dynamique linéaire [Léz21] :

- Les angles de roulis et tangage θ et ϕ sont petits, ce qui permet de faire les approximations suivantes :

$$\begin{aligned} \dot{\Theta} &\approx R_z(\psi)\boldsymbol{\omega} \\ \mathbf{I} &\approx R_z(\psi) \mathbf{I}_b R_z(\psi)^T \end{aligned} \quad (2.7)$$

Où \mathbf{I}_b est dans le repère de la base et \mathbf{I} dans le repère global.

- Les vitesses de roulis et tangage sont petites, et les termes non-diagonaux des matrices d'inerties aussi :

$$\frac{d}{dt}(\mathbf{I}\boldsymbol{\omega}) = \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \wedge (\mathbf{I}\boldsymbol{\omega}) \approx \mathbf{I}\dot{\boldsymbol{\omega}} \quad (2.8)$$

- L'angle de lacet de référence est bien suivi, ce qui permet de faire l'approximation :

$$R_z(\psi) \approx R_z(\psi^*) \quad (2.9)$$

- La position de la base suit correctement la consigne de référence \mathbf{p}^* , et nous pouvons écrire :

$$\sum_i (\mathbf{r}_i - \mathbf{c}) \wedge \mathbf{f}_i \approx \sum_i (\mathbf{r}_i - \mathbf{p}^*) \wedge \mathbf{f}_i \quad (2.10)$$

Ces hypothèses nous permettent d'obtenir une version linéaire de la dynamique centrodiale : $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ où f est une fonction linéaire et le contrôle \mathbf{u} est le vecteur des forces de contact \mathbf{f} .

2.1.3 Corps-complet

On utilise la *configuration* d'un robot pour décrire de manière unique sa position dans l'espace. Ce vecteur contient la position $\mathbf{p} \in \mathbb{R}^3$ et l'orientation $\mathbf{q} \in \mathbb{H} \in \mathbb{R}^4$ (qui décrit un élément de $SO(3)$) d'un des corps qui le composent (généralement la base), ainsi que la configuration $\mathbf{q}^j \in \mathbb{R}^N$ de ses N articulations.

$$\mathbf{q} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \mathbf{q}^j \end{bmatrix} \quad (2.11)$$

La dynamique Lagrangienne complète du robot s'écrit de la manière suivante :

$$\mathbf{M}(\mathbf{q}) \begin{bmatrix} \ddot{\mathbf{p}} + \mathbf{g} \\ \dot{\omega} \\ \ddot{\mathbf{q}}^j \end{bmatrix} + \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \tau \end{bmatrix} + \sum_i \mathbf{J}_i(\mathbf{q})^T \mathbf{f} \quad (2.12)$$

où ω est la vitesse de rotation de la base (ou du corps choisi), $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{(N+6) \times (N+6)}$ est la matrice d'inertie généralisée, $\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{N+6}$ le vecteur des Coriolis et des effets centrifugaux, $\tau \in \mathbb{R}^N$ le vecteur des couples articulaires et $\mathbf{J}_i(\mathbf{q}) \in \mathbb{R}^{(N+6) \times 3}$ est la matrice Jacobienne associée au point de contact i .

2.1.4 Contacts

La dynamique Lagrangienne tout comme la dynamique centrodiale permettent de voir que les forces de contact avec l'environnement sont cruciales dans la modélisation et le contrôle des robots à pattes. Une modélisation correcte des contacts et des forces associées est donc nécessaire pour obtenir une bonne modélisation de la dynamique globale des robots à pattes.

Les forces respectent naturellement certaines propriétés. Tout d'abord, ces forces sont unilatérales (le robot ne peut que pousser sur le sol). Ainsi la composante normale d'une force \mathbf{f}_i^z est toujours positive. De plus, les forces respectent les contraintes de frottements. Le modèle de frottement de Coulomb est le plus répandu. Sans glissement, les composantes tangentielles $\mathbf{f}_i^{x,y}$ sont bornées :

$$\|\mathbf{f}^{x,y}\| \leq \mu \mathbf{f}^z \quad (2.13)$$

où μ est le coefficient de frottement. Il dépend notamment des matériaux en contact. Dans le cas d'un glissement, les forces tangentielles sont proportionnelles à la composante normale dans la direction opposée au glissement. Si $\dot{\mathbf{p}}_i^{x,y}$ est la vitesse du point de contact i :

$$\mathbf{f}_i^{x,y} = -\mu \mathbf{f}_i^z \frac{\dot{\mathbf{p}}_i^{x,y}}{\|\dot{\mathbf{p}}_i^{x,y}\|} \quad (2.14)$$

Les contacts sont modélisés de manière élastique ou rigide.

Modélisation élastique

Les modèles de contact souples considèrent la souplesse des matériaux qui permet une légère pénétration. Le point de contact \mathbf{p}_i peut légèrement rentrer dans la surface de contact et la force appliquée dépend de sa position et vitesse orthogonale en z :

$$\mathbf{p}_i^z \leq 0 \quad \Longrightarrow \quad \mathbf{f}_i^z = \max(-\mathbf{K}_p \mathbf{p}_i^z - \mathbf{K}_v \dot{\mathbf{p}}_i^z, 0) \quad (2.15)$$

On voit qu'une saturation est ajoutée pour respecter la contrainte d'unilatéralité de la composante orthogonale de la force. Sans contact il n'y a en revanche pas de force de contact :

$$\mathbf{p}_i^z \geq 0 \quad \Longrightarrow \quad \mathbf{f}_i^z = 0 \quad (2.16)$$

Cette modélisation est fidèle à la réalité pour des coefficients de ressort très élevés. Mais de tels coefficients rendent difficile l'intégration numérique et le conditionnement du problème.

Modélisation rigide

Un modèle simplifié de contact rigide est donc plus largement utilisé. Il consiste simplement à considérer que s'il y a un contact la force peut prendre n'importe quelle valeur positive, et s'il n'y a pas de contact ($\mathbf{p}_i^z \geq 0$) elle vaut 0. Comme ce modèle n'autorise pas la pénétration, lors d'une création de contact la vitesse du point de contact peut changer de manière instantanée, passant de $\dot{\mathbf{p}}_i^-$ à $\dot{\mathbf{p}}_i^+$. C'est ce qu'on appelle un impact. Des modèles existent pour décrire la relation entre l'état avant l'impact et l'état après impact, mais c'est toujours un problème ouvert aujourd'hui, particulièrement dans le cas où il y a plusieurs contacts.

Une approche classique consiste à considérer que le point de contact reste *collé* à la surface de contact après l'impact : $\dot{\mathbf{p}}_i^+ = 0$. Ce n'est pas une bonne hypothèse en général et la plupart des contrôleurs cherchent donc à éviter les impacts en imposant une vitesse des points de contact nulle au moment de la création d'un contact.

Et quand les hypothèses du contact sont violées...

On a défini ici ce qu'il se passe quand le système est en contact et les conditions qui sont nécessaires pour qu'il le reste. Si ses conditions sont violées, alors le contact se rompt et le système passe en mouvement libre. Un modèle de contact complet doit être capable de tenir compte de ce changement de paradigme. Deux phénomènes principaux sont alors modélisés : l'unilatéralité du contact, c'est-à-dire le fait que les forces ne permettent pas d'attirer les deux objets en contact ; et le glissement, c'est-à-dire le fait qu'un effort tangentiel trop important induit un mouvement colinéaire à la force.

Les modèles de contact rigide conduisent à des formulations dites linéaires-complémentaires, implémentés par exemple dans les classiques simulateurs Bullet [Bul] ou ODE [Ode]. Des relaxations diverses [Muj] permettent une résolution plus simple, plus rapide ou plus stable.

Les modèles que nous avons exposés ont pour objectif d'être utilisés comme contraintes prédictives dans nos OCPs. Comme nous l'avons mentionné, ils ont l'avantage d'être lisses tant que la phase de contact ne change pas. Il est alors nécessaire que l'utilisateur spécifie explicitement les instants de changement de contact, car la discontinuité correspondante empêche nos solveurs d'OCPs de considérer ce phénomène. Une limitation importante des approches de la locomotion fondées sur les modèles est justement cette incapacité à raisonner sur les changements de contact, et nous en discuterons au chapitre 4.

Notons ici simplement que quelques travaux pour inclure des simulateurs de contact complet (avec unilatéralité) dans un OCP existent. Hämäläinen le formule directement avec un simulateur unilatéral rigide, en exploitant un solveur sans gradient pour optimiser sa trajectoire [Häm15 ; Raj16]. La relaxation du modèle physique de MuJoCo permet d'obtenir une dynamique lisse au changement de contact, que le solveur peut alors exploiter directement [Tas12]. La récente ouverture du logiciel sous-jacent permettra sans doute de mieux exploiter cette propriété. Enfin, d'autres équipes cherchent à estimer des dérivés malgré les discontinuités en utilisant du lissage par randomisation, avec des résultats pour l'instant préliminaire [Lid22].

2.2 Contrôle Optimal

À partir de la modélisation du système, nous pouvons prédire son évolution en fonction d'un contrôle. En robotique, suivant le modèle utilisé, le contrôle et l'état ne sont pas les mêmes. Dans le cas de la dynamique inertielle rigide, l'état est généralement composé des positions, orientations et vitesses (de translation et rotation) de la base du robot, et le contrôle correspond aux forces de contact avec l'environnement. Dans le cas de la dynamique Lagrangienne, l'état est composé du vecteur configuration et de sa dérivée, et le contrôle contient les couples articulaires.

On peut alors essayer de trouver le contrôle à appliquer pour obtenir un *bon* comportement du robot. Pour cela il faut d'abord définir ce qu'est un bon comportement, puis le traduire sous forme de fonction à optimiser. En général cette fonction, appelée fonction coût, va contenir

des termes pour garantir la stabilité du robot, et d'autres pour atteindre des objectifs, de suivi de position ou de vitesse par exemple.

Ensuite on écrit un problème de contrôle optimal, qui prend en compte le modèle dynamique, la fonction de coût, et des contraintes liées aux conditions initiales et aux contraintes sur les forces ou les actionneurs par exemple. Ce problème est résolu par une méthode numérique et permet d'obtenir une trajectoire de l'état du robot et la politique de contrôle optimale associée.

Nous présenterons la formulation générale d'un OCP, et sa formulation discrète qui est couramment utilisée pour la résolution. Nous verrons ensuite un algorithme de résolution numérique, le DDP.

2.2.1 Présentation

Un système dynamique contrôlable peut être décrit par l'évolution de son état \mathbf{x} en fonction d'un contrôle \mathbf{u} :

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t) \quad (2.17)$$

Le *contrôle optimal* consiste à calculer à partir d'un état initial (éventuellement aussi d'un état final) une trajectoire de l'état associé à un contrôle qui minimise un coût. Le problème peut s'écrire sous la forme d'un problème d'optimisation :

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \quad & \int_0^T \ell(\mathbf{x}(t), \mathbf{u}(t), t) dt + \ell_T(\mathbf{x}(T)) \\ \text{tel que} \quad & \mathbf{x}(0) = \mathbf{x}_0 \quad (\text{éventuellement } \mathbf{x}(T) = \mathbf{x}_T) \\ & \forall t \in [0, T] : \\ & \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t) \\ & h(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0 \end{aligned} \quad (2.18)$$

où $\mathbf{x} : t \mapsto \mathbf{x}(t)$ et $\mathbf{u} : t \mapsto \mathbf{u}(t)$ sont l'état et le contrôle, deux fonctions du temps t ; $h(\cdot)$ regroupe les contraintes sur le chemin et \mathbf{x}_0 et \mathbf{x}_T sont les valeurs initiales et finales. ℓ est le coût intégral et ℓ_T le coût terminal.

Les méthodes pour la résolution numérique d'un tel problème sont généralement réparties en deux catégories [Tré05] :

- **Les méthodes directes** qui consistent à discrétiser l'état et le contrôle ce qui réduit le problème à un problème d'optimisation non linéaire.
- **Les méthodes indirectes** exploitent les principes d'optimalité du problème initial pour le reformuler en un problème aux dérivées. Elles utilisent en particulier directement le principe du maximum de Pontryagin pour obtenir le problème aux valeurs limites associé puis le résolvent par des méthodes de tir [Tré05]. Associée au principe

d'optimalité de Bellman, la programmation dynamique permet aussi une reformulation en des problèmes aux dérivées partielles, permettant des exploitations variées. Elle connaît aujourd'hui un regain d'intérêt en robotique, notamment par son utilisation comme solveur linéaire dans des méthodes de résolution directes.

Les méthodes directes sont faciles à mettre en œuvre, permettent de prendre en compte des contraintes sur l'état et sont efficaces en basse dimension, mais elles ne prennent pas en compte la dimension temporelle du problème et la sparsité qui en découle, elles sont gourmandes en mémoire et peuvent avoir une faible précision numérique. À l'opposé, les méthodes indirectes demandent plus de connaissances de la structure du système et sont moins stables numériquement, mais plus efficaces en toutes dimensions.

2.2.2 Problème discret

Les approches directes *transcrivent* le problème continu en un problème statique (de taille finie). Plusieurs transcriptions existent. Nous avons choisi de ne pas explorer les transcriptions par collocation [Pos14], mais plutôt des méthodes de tir, avec discrétisation du contrôle constant par morceau, que nous présentons ici.

Le temps est tout d'abord discrétisé en $n + 1$ *nœuds* : $[t_0, \dots, t_n]$; et \mathbf{X} and \mathbf{U} sont obtenus en discrétisant \mathbf{x} and \mathbf{u} à chaque nœud :

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(t_0) \triangleq \mathbf{x}_0 \\ \mathbf{x}(t_1) \triangleq \mathbf{x}_1 \\ \vdots \\ \mathbf{x}(t_n) \triangleq \mathbf{x}_n \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} \mathbf{u}(t_0) \triangleq \mathbf{u}_0 \\ \mathbf{u}(t_1) \triangleq \mathbf{u}_1 \\ \vdots \\ \mathbf{u}(t_{n-1}) \triangleq \mathbf{u}_{n-1} \end{bmatrix} \quad (2.19)$$

La formulation du problème d'optimisation non linéaire obtenu par discrétisation est donc :

$$\begin{aligned} \min_{\mathbf{X}, \mathbf{U}} \quad & \sum_{i=0}^{n-1} \ell_i(\mathbf{x}_i, \mathbf{u}_i) + \ell_T(\mathbf{x}_n) \\ \text{tel que} \quad & \mathbf{x}_0 = x_0 \quad \mathbf{x}_n = x_n \\ & \forall i \in [t_0, t_{n-1}] : \\ & \quad \mathbf{x}_{i+1} = f_i(\mathbf{x}_i, \mathbf{u}_i) \\ & \quad h_i(\mathbf{x}_i, \mathbf{u}_i) \leq 0 \end{aligned} \quad (2.20)$$

où $f_i(\cdot)$ est une version discrète de la dynamique continue du système obtenue par intégration numérique [Bet10] (par exemple par les méthodes de Runge-Kutta).

Des solveurs numériques comme Ipopt [Bir16] permettent de résoudre directement ce type de problème, mais ils n'utilisent pas la sparsité introduite par la temporalité, ce qui limite leurs performances pour des systèmes complexes. Pour ce faire, les roboticiens ont souvent recours à une méthode exploitant la programmation dynamique.

2.2.3 Programmation dynamique

La programmation dynamique (appelée *Dynamic Programming* ou **DP**) voit le jour dans les années 1950 avec les travaux de Bellman [Bel54]. Elle repose sur le principe d'optimalité de Bellman : toute sous-trajectoire d'une trajectoire optimale est optimale.

Le coût restant (appelé *cost-to-go* en anglais), ou fonction valeur (*Value function* en anglais), est défini pour un état de départ \mathbf{y} :

$$V_m(\mathbf{y}) = \min_{\mathbf{x}_{m:n}, \mathbf{U}_{m:n-1}} \sum_{i=m}^{n-1} \ell_i(\mathbf{x}_i, \mathbf{u}_i) + \ell_T(\mathbf{x}_n) \quad \mathbf{x}_m = \mathbf{y} \quad (2.21)$$

On remarque que $V_0(\mathbf{x}_0)$ correspond au coût associé au contrôle optimal solution de (2.20) et $\forall \mathbf{y}$, $V_n(\mathbf{y}) = \ell_T(\mathbf{y})$. On peut se ramener facilement à un problème d'optimisation sans contraintes en incorporant au coût intégral une fonction indicatrice associée à un poids virtuellement infini.

De plus le principe d'optimalité de Bellman nous permet d'écrire pour tout $0 < m < n$:

$$V_0(\mathbf{x}_0) = \min_{\mathbf{x}_{1:m}, \mathbf{U}_{0:m-1}} \sum_{i=0}^{m-1} \ell_i(\mathbf{x}_i, \mathbf{u}_i) + V_m(\mathbf{x}_m) \quad (2.22)$$

On peut déduire facilement la formule de récursion sur V :

$$V_i(\mathbf{x}_i) = \min_{\mathbf{u}_i} \ell_i(\mathbf{x}_i, \mathbf{u}_i) + V_{i+1}(f_i(\mathbf{x}_i, \mathbf{u}_i)) \quad (2.23)$$

En partant de $i = n$ où l'on sait que $V_n(\mathbf{x}_n) = \ell_T(\mathbf{x}_n)$ on peut trouver le contrôle optimal :

$$\begin{aligned} V_i(\mathbf{y}) &= \min_{\mathbf{u}} \ell_i(\mathbf{y}, \mathbf{u}) + V_{i+1}(f_i(\mathbf{y}, \mathbf{u})) \\ \mathbf{u}_i^x(\mathbf{x}) &= \arg \min_{\mathbf{u}} \ell_i(\mathbf{x}_i, \mathbf{u}_i) + V_{i+1}(f_i(\mathbf{x}, \mathbf{u})) \end{aligned} \quad (2.24)$$

Ce problème d'optimisation paramétrique ne peut être résolu que pour certains problèmes très particuliers où la fonction valeur peut être calculée analytiquement. Quand l'état est discret ou de très faible dimension on peut enregistrer dans un tableau les valeurs prises à chaque temps et chaque état, mais la combinatoire augmente rapidement et limite les cas d'application.

Les problèmes avec une dynamique linéaire et un coût quadratique, appelés *Linear Quadratic Regulator* (**LQR**) peuvent aussi être résolus avec de la programmation dynamique. Dans ce cas, la fonction valeur est quadratique et peut-être calculée analytiquement pour tout $i \in [t_0, t_n]$. Finalement le contrôle optimal peut-être calculé comme une fonction linéaire de l'état dont on peut calculer le coefficient directeur, appelé *feedback gain* ou gain de Ricatti.

2.2.4 DDP

Le *Differential Dynamic Programming* ou **DDP** est un algorithme permettant de résoudre numériquement des **OCP** non linéaires discrets sans contraintes autres que la dynamique et l'état initial. L'idée générale derrière le **DDP** est de partir d'un contrôle \mathbf{U} , de linéariser le système autour de ce contrôle et de résoudre le LQR correspondant, qui permet de trouver les gains d'anticipation et de rétroaction d'une politique optimale locale. On effectue ensuite une recherche de longueur de pas (*line search* en anglais) pour s'assurer de la convergence de l'algorithme. On réitère l'opération jusqu'à avoir convergé vers une solution.

Passé arrière

On introduit la fonction $Q_i(\delta\mathbf{x}, \delta\mathbf{u}) = \ell_i(\bar{\mathbf{x}}_i + \delta\mathbf{x}, \bar{\mathbf{u}} + \delta\mathbf{u}) + V_{i+1}(f_i(\bar{\mathbf{x}}_i + \delta\mathbf{x}, \bar{\mathbf{u}} + \delta\mathbf{u}))$, définie pour la trajectoire courante de l'état $\bar{\mathbf{x}}_i$ et du contrôle $\bar{\mathbf{u}}_i$. En reprenant (2.24) :

$$V_i(\mathbf{y}) = \min_{\delta\mathbf{u}} Q_i(\delta\mathbf{x}, \delta\mathbf{u}) \quad (2.25)$$

On utilise un polynôme de Taylor du second degré pour obtenir une approximation polynomiale de Q_i :

$$Q_i(\delta\mathbf{x}, \delta\mathbf{u}) \simeq \frac{1}{2} \begin{bmatrix} 1 \\ \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix}^T \begin{bmatrix} 0 & Q_{i,x}^T & Q_{i,u}^T \\ Q_{i,x} & Q_{i,xx} & Q_{i,xu} \\ Q_{i,u} & Q_{i,xu}^T & Q_{i,uu} \end{bmatrix} \begin{bmatrix} 1 \\ \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix} \quad (2.26)$$

avec :

$$Q_{i,x} = \frac{\partial Q_i}{\partial \mathbf{x}} \quad Q_{i,u} = \frac{\partial Q_i}{\partial \mathbf{u}} \quad Q_{i,xx} = \frac{\partial^2 Q_i}{\partial \mathbf{x}^2} \quad Q_{i,uu} = \frac{\partial^2 Q_i}{\partial \mathbf{u}^2} \quad Q_{i,xu} = \frac{\partial Q_{i,u}}{\partial \mathbf{x}} \quad (2.27)$$

Que l'on peut calculer :

$$\begin{aligned} Q_{i,x} &= \ell_{i,x} + f_{i,x}^T V_{i+1,x} \\ Q_{i,u} &= \ell_{i,u} + f_{i,u}^T V_{i+1,x} \\ Q_{i,xx} &= \ell_{i,xx} + f_{i,x}^T V_{i+1,xx} f_{i,x} + V_{i+1,x} f_{i,xx} \\ Q_{i,uu} &= \ell_{i,uu} + f_{i,u}^T V_{i+1,xx} f_{i,u} + V_{i+1,x} f_{i,uu} \\ Q_{i,xu} &= \ell_{i,xu} + f_{i,u}^T V_{i+1,xx} f_{i,u} + V_{i+1,x} f_{i,xu} \end{aligned} \quad (2.28)$$

où $\ell_{i,x}$, $\ell_{i,u}$, $\ell_{i,xx}$, $\ell_{i,xu}$ et $\ell_{i,uu}$ sont les dérivées de la fonction coût et $f_{i,x}$ et $f_{i,u}$ celles de la dynamique. $V_{i+1,x}$ et $V_{i+1,xx}$ sont le gradient et la hessienne de la fonction valeur V_{i+1} . En supposant V_{i+1} connue on peut donc calculer les dérivées de Q_i et utiliser ce modèle quadratique local pour trouver le gain optimal en calculant le zéro de la dérivée :

$$\frac{\partial Q_i(\delta\mathbf{x}, \delta\mathbf{u})}{\partial \delta\mathbf{u}} = 0 \quad (2.29)$$

$$\begin{aligned}\delta \mathbf{u}_i^*(\delta \mathbf{x}) &= \arg \min_{\delta \mathbf{u}} Q_i(\delta \mathbf{x}, \delta \mathbf{u}) \\ &= -Q_{i,uu}^{-1}(Q_{i,u} + Q_{i,xu}\delta \mathbf{x})\end{aligned}\tag{2.30}$$

Finalemment on définit les gains d'anticipation \mathbf{k}_i (*feedforward gain*), et de rétroaction \mathbf{K}_i (*feedback gain* - ou gain de Ricatti) tels que :

$$\delta \mathbf{u}_i^*(\delta \mathbf{x}) = \mathbf{k}_i + \mathbf{K}_i \delta \mathbf{x}\tag{2.31}$$

D'après (2.25), on peut calculer V_i en remplaçant $\delta \mathbf{u}$ par $\delta \mathbf{u}_i^*(\delta \mathbf{x})$: $V_i(\delta \mathbf{x}) = Q(\delta \mathbf{x}, \delta \mathbf{u}_i^*(\delta \mathbf{x}))$.

Ainsi en partant de $V_n(\mathbf{y}) = \ell_T(\mathbf{y})$, on peut calculer itérativement V_i pour tout i . C'est ce qu'on appelle la passe-arrière (*backward pass*),

Passé en avant

La passe-arrière ne nous donne pas un contrôle optimal, mais une politique optimale en fonction de la distance à notre trajectoire courante. On utilise cette politique pour calculer une nouvelle trajectoire dans la passe en avant (*forward pass*) :

$$\begin{aligned}x_0 &= \bar{\mathbf{x}}_0 \\ u_i &= \bar{\mathbf{u}}_i + \alpha \mathbf{k}_i + \mathbf{K}_i(x_i - \bar{\mathbf{x}}_i) \\ x_{i+1} &= f_i(x_i, u_i)\end{aligned}\tag{2.32}$$

Le pas α est calculé par recherche de longueur de pas pour s'assurer de la convergence de l'algorithme : initialisé à 1, il est diminué tant que le coût total associé aux nouvelles trajectoires d'état et de contrôle n'est pas meilleur que le coût de la trajectoire précédente.

2.2.5 Programmation Différentielle Dynamique Cherchant la Faisabilité

Cet algorithme peut être légèrement modifié pour accepter une trajectoire initiale non faisable. Le solveur Crocodyl [Mas20] propose ainsi un nouvel algorithme appelé Programmation Différentielle Dynamique Cherchant la Faisabilité (*Feasibility-driven Differential Dynamic Programming - FDDP*). Par rapport au DDP classique, le FDDP introduit des nœuds de discrétisation intermédiaires comme nouvelles variables d'optimisation. Leur objectif est de fermer les écarts \bar{f}_{i+1} entre l'intégration de la dynamique et la trajectoire de l'état :

$$\bar{f}_{i+1} = f_i(\mathbf{x}_i, \mathbf{u}_i) - \mathbf{x}_{i+1}\tag{2.33}$$

L'approximation quadratique de la fonction Q_i , dont on a négligé la dérivée seconde de la dynamique pour simplifier le calcul, devient alors :

$$\begin{aligned}
Q_{i,x} &= \ell_{i,x} + f_{i,x}^T \bar{V}_{i+1,x} \\
Q_{i,u} &= \ell_{i,u} + f_{i,u}^T \bar{V}_{i+1,x} \\
Q_{i,xx} &= \ell_{i,xx} + f_{i,x}^T V_{i+1,xx} f_x \\
Q_{i,uu} &= \ell_{i,uu} + f_{i,x}^T V_{i+1,xx} f_u \\
Q_{i,xu} &= \ell_{i,xu} + f_{i,u}^T V_{i+1,xx} f_u
\end{aligned} \tag{2.34}$$

où $\bar{V}_{i+1,x} = V_{i+1,x} + V_{i+1,xx} \bar{f}_{i+1}$ est le gradient de la fonction valeur après la déflexion due à l'écart \bar{f}_{i+1} . La nouvelle trajectoire de l'état \mathbf{x} est obtenue par intégration dans la *forward pass* :

$$\begin{aligned}
\bar{\mathbf{u}}_i &= \mathbf{u}_i + \alpha \mathbf{k}_i + \mathbf{K}_i (\bar{\mathbf{x}}_i - \times_i) \\
\bar{\mathbf{x}}_{i+1} &= f_i(\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i) - (1 - \alpha) \bar{f}_{i+1}
\end{aligned} \tag{2.35}$$

où $\alpha \in [0, 1]$ est le pas de la recherche de longueur de pas, \mathbf{k}_i et \mathbf{K}_i sont les gains d'anticipation et de rétroaction calculés dans la passe-arrière.

Le solveur Crocoddyl a été largement utilisé pendant cette thèse pour la résolution numérique d'OCPs non-linéaires. Il a l'avantage d'être suffisamment efficace pour permettre une résolution en ligne à fréquence élevée de l'OCP. En revanche, l'impossibilité de prendre en compte d'autres contraintes que la dynamique le rend difficile à utiliser sur certains problèmes (voir Chapitre 4). La recherche en cours sur la résolution numérique des OCPs porte notamment sur ce point, avec le développement de nouveaux solveurs [Jal22c ; Jal22a ; Jal22b].

2.3 Le robot quadrupède Solo

Cette thèse a eu pour cadre d'application le robot quadrupède Solo. Les simulations ont été effectuées avec un modèle de ce robot et les expériences sur ce robot. Si les méthodes déployées se veulent adaptables à grand nombre de robots à pattes, les spécificités de Solo doivent être comprises pour voir leur impact sur le comportement du contrôleur.

Solo est un petit robot quadrupède (22 cm de haut) de 2,5 kg. Il a douze degrés de liberté, trois à chaque jambe, qui lui donnent une grande liberté de mouvement pour chacun de ses pieds. Les actionneurs sont de faible rapport de réduction ce qui leur permet d'être contrôlés en couple. Des encodeurs sont utilisés pour connaître la position de chacun des actionneurs. Une centrale inertielle intégrant un filtre de Kalman étendu est monté sur la base du robot.

2.3.1 Open Dynamic Robot Initiative

La conception de petits robots peu chers à partir de pièces imprimées en 3D [Pas16] se développe rapidement grâce à la démocratisation des imprimantes 3D de bonne qualité.

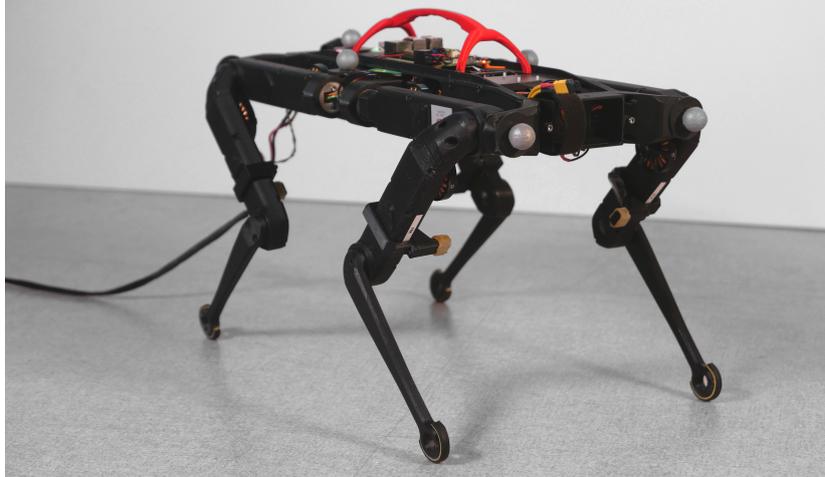


FIGURE 8 – Le robot quadrupède Solo

Elles permettent un prototypage et une construction rapide pour une utilisation dans des environnements variés.

Le projet ODRI (Open Dynamic Robot Initiative) naît en 2019 avec pour objectif de développer un actionneur pour contrôler en couple des robots construits principalement à partir de pièces imprimées en 3D et de composants du commerce. Ce projet est une collaboration entre le *Motion Generation and Control Group*, le *Dynamic Locomotion Group* et le *Robotics Central Scientific Facility* à l’institut Max-Planck, le *Machines in Motion Laboratory* à New York University’s Tandon School of Engineering et l’équipe Gepetto au Laboratoire d’Analyse et d’Architecture des Systèmes LAAS-CNRS.

L’actionneur, peu cher et peu complexe, doit permettre de développer des robots à pattes et manipulateurs fiables et à la maintenance facile. Tous les plans des pièces, cartes électroniques et les logiciels sont disponibles sous licence BSD-3 [Odr]. Un premier quadrupède est présenté avec l’actionneur [Gri19b] en 2020. Un robot bipède et d’autres robots ont été développés depuis.

2.3.2 Contrôleur

Le contrôle se fait en deux étapes : contrôle *haut-niveau* et contrôle *bas-niveau*. Le contrôle bas-niveau se fait par impédance. La boucle de contrôle bas-niveau tourne sur le robot à une fréquence de 10 kHz. Le schéma 9 illustre ce type de contrôle. Les consignes d’impédance reçues du contrôleur haut-niveau sont un couple d’anticipation τ^* , une configuration de référence \mathbf{q}^* et une vitesse de référence $\dot{\mathbf{q}}^*$. Les couples τ envoyés aux moteurs sont calculés par un PD+ :

$$\tau = K_p(\mathbf{q}^* - \mathbf{q}) + K_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}}) + \tau^* \quad (2.36)$$

La boucle de contrôle haut-niveau tourne quant à elle sur un ordinateur distant (Intel(R))

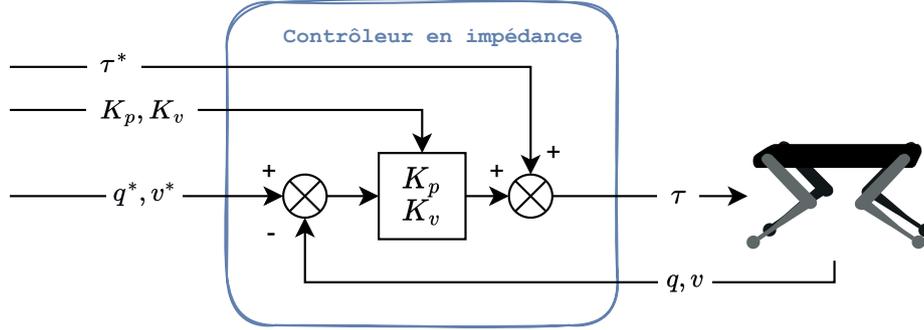


FIGURE 9 – Contrôleur en impédance d'un robot

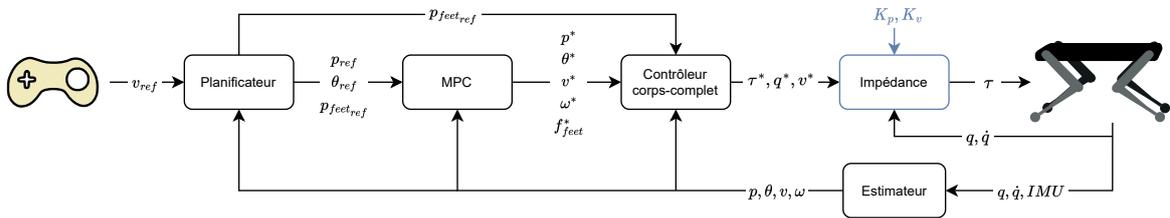


FIGURE 10 – Contrôleur du robot Solo [Léz21]

Core(TM) i5-9500 CPU @ 3.00GHz) relié par Ethernet (ou WIFI) à la carte de contrôle du robot. Elle calcule des consignes d'impédance τ^* , \mathbf{q}^* et $\dot{\mathbf{q}}^*$, et éventuellement les gains associés K_p et K_d qui sont envoyés aux cartes du robot à 1 kHz. Le contrôleur proposé par Pierre-Alexandre Léziart [Léz21 ; Léz22] a servi de base aux développements proposés dans cette thèse. Une brève présentation des différents éléments qui le composent est faite dans cette section. De plus amples détails sont disponibles dans l'article [Léz21]. Ce contrôleur prend en entrée une vitesse de référence donnée par l'utilisateur à travers une manette de jeu. Un MPC optimise les forces à appliquer puis un contrôleur corps-complet calcule les consignes à envoyer au contrôleur bas-niveau. Les gains K_p et K_d sont fixés et ne sont pas modifiés par le contrôleur. Le schéma 10 présente cette architecture.

Estimation d'état

L'estimateur utilisé sur Solo combine la cinématique directe et les mesures de l'IMU pour estimer la vitesse linéaire et angulaire de la base.

L'IMU intégrée à Solo comprend un filtre de Kalman étendu (en anglais *extended Kalman Filter - EKF*) qui estime la vitesse angulaire de la base, son orientation et l'accélération linéaire sans la gravité. La cinématique des jambes est mesurée grâce aux encodeurs. La vitesse des articulations est obtenue par différence finie des données des encodeurs.

La vitesse angulaire de la base est mesurée directement par le gyroscope de l'IMU. En revanche la vitesse linéaire doit être rectifiée en combinant les données des différents capteurs. Un filtre complémentaire permet de combiner les différentes mesures :

- Chaque pied en contact est supposé immobile (pas de glissement) ce qui permet - par cinématique directe - d'obtenir des informations partielles sur la vitesse de la base par rapport à ce pied. Grâce à l'orientation de la base et la vitesse angulaire estimées par le filtre de Kalman de l'IMU on peut calculer une estimation complète de la vitesse de la base. En moyennant sur les différents pieds en contact on obtient une information bruitée, mais sans dérive de la vitesse de la base. C'est donc une bonne source basse fréquence.
- L'intégration de l'accélération linéaire sans gravité permet de calculer une seconde estimation de la vitesse de la base. Celle-ci permet de percevoir les changements rapides dans la vitesse, mais est sujette à la dérive à cause de l'intégration. Elle constitue une source haute fréquence fiable.

Le filtre complémentaire combine ces deux sources en sommant un filtre passe-bas de l'estimation de la cinématique et un filtre passe-haut de l'intégration de l'IMU. La fréquence de coupure est choisie à 5 Hz.

Planificateur

Référence de la base Une trajectoire de référence en translation et rotation de la base est calculée sur un horizon temporel fixé par le planificateur. La position initiale correspond à l'estimation courante de la position du robot. Les vitesses de référence données par l'utilisateur grâce à une manette de jeu ($v_{ref,x}$, $v_{ref,y}$, ω_{ref}) sont ensuite intégrées pour obtenir les trajectoires en (x, y, yaw) sur l'horizon.

Position des contacts et heuristique de Raibert Le plan de contact est obtenu en utilisant une heuristique, que l'on appelle *l'heuristique de Raibert*. Elle permet de calculer une position des points de contacts adaptée au suivi d'une consigne de vitesse de la base à partir de la vitesse courante. En supposant que le robot marche sur sol plat, la position en 2D du prochain point de contact \mathbf{p} est donné par la formule :

$$\mathbf{p}(\mathbf{v}_{ref}) = \begin{bmatrix} p_{sh}^x + \frac{1}{2}t_{contact}v^x + k(v^x - v_{ref}^x) + \frac{1}{2}\sqrt{\frac{h}{\mathbf{g}}}\mathbf{v}^x \times \omega_{ref} \\ p_{sh}^y + \frac{1}{2}t_{contact}v^y + k(v^y - v_{ref}^y) + \frac{1}{2}\sqrt{\frac{h}{\mathbf{g}}}\mathbf{v}^y \times \omega_{ref} \end{bmatrix} \quad (2.37)$$

où $\mathbf{p}_{sh} = [p_{sh}^x \ p_{sh}^y]^T$ est la position des épaules, k est un coefficient de rétroaction, $\mathbf{v} = [v^x \ v^y]^T$ la vitesse courante de la base du robot, $\mathbf{v}_{ref} = [v_{ref}^x \ v_{ref}^y]^T$ la vitesse linéaire de référence, ω_{ref} la vitesse angulaire de référence autour de l'axe z (angle de lacet), h la hauteur du robot, \mathbf{g} la gravité et $t_{contact}$ la durée du contact.

Trajectoires des pieds de vol Les trajectoires de référence de chaque pied de vol i sont calculées séparément pour les coordonnées en X et Y et en Z .

Un polynôme de degré 5 est utilisé pour calculer la trajectoire en X et Y $\mathbf{p}_{i,ref}^{x,y}(t)$ en respectant des contraintes de continuité : les positions, vitesses et accélérations initiales de la courbe doivent correspondre à l'état courant du pied, et à la fin du pas la position est celle du contact et la vitesse et l'accélération sont nulles.

La trajectoire de référence en Z du pied de vol i , $\mathbf{p}_{i,ref}^z(t)$ est calculée de la même manière, mais cette fois-ci le polynôme est de degré 6 pour permettre à l'utilisateur de choisir la hauteur du pied à la moitié de la phase de vol.

MPC centroïdal

Un MPC résout ensuite un OCP linéaire exploitant le modèle inertiel-rigide du robot. Il tourne à 50 Hz avec un horizon temporel de 0.32 s. Il utilise les positions et vitesses de référence de la base données par le planificateur pour optimiser les forces de contact avec l'environnement. La séquence de contact et les positions des pieds sont aussi données par le planificateur. Le solveur utilisé pour la résolution numérique de l'OCP est OSQP [Ste20].

Contrôle corps-complet

Un contrôleur corps-complet tourne à 1 kHz pour calculer τ^* , \mathbf{q}^* et $\dot{\mathbf{q}}^*$. Il comprend une cinématique inverse [Esc14] qui utilise le modèle complet du robot [Car19b] pour calculer les positions, vitesses et accélérations articulaires. Les tâches permettent de suivre les trajectoires d'état du modèle inertiel-rigide et les positions des pieds de vol données par le planificateur.

Un QP est ensuite résolu pour calculer les couples d'anticipation τ^* envoyés au contrôleur bas-niveau. Pour cela il réalise un compromis entre les forces données par le MPC et l'accélération de la cinématique inverse, tout en garantissant le respect de la dynamique.

Conclusion

Cette section présente quelques outils du roboticien. Ce sont les outils qui seront utiles à la compréhension des travaux réalisés pendant cette thèse. Les différents modèles utilisés au cours de cette thèse ainsi que quelques fondements théoriques utiles à la résolution d'un problème de contrôle optimal sont ainsi présentés. Finalement, la dernière partie propose une description du robot quadrupède Solo et du contrôleur de référence qui sert de base aux développements de cette thèse.

Planification réactive et évitement d'obstacles

Contents

3.1	Planification réactive en environnements complexes	41
3.1.1	Choix des surfaces de contact	42
3.1.2	Adaptation des positions des pieds	44
3.1.3	Évitement des collisions pour les pieds de vol	45
3.2	Adaptation du contrôleur de Solo	46
3.2.1	Estimateur	46
3.2.2	Guide de trajectoire	47
3.2.3	Surfaces potentielles pour chaque pas	47
3.3	Étude expérimentale	48
3.3.1	Comment évaluer le contrôleur	48
3.3.2	Résultats qualitatifs	49
3.4	Analyse quantitative	49
3.4.1	Influence de chaque module sur la robustesse	49
3.4.2	Utilité de la planification des contacts	49
3.4.3	La nécessité de l'adaptation du pas	50
3.4.4	Évitement de collisions	50
3.4.5	Effet de la longueur de l'horizon du planificateur de surface	51
3.4.6	Temps de calcul de chaque module	51

Introduction

Les contrôleurs de locomotion ont souvent démontré des capacités de fonctionnement au-delà des conditions expérimentales pour lesquelles ils avaient été créés. Ainsi nous avons vu dans la Section 1.1 que des contrôleurs "aveugles" permettent aux robots de monter des escaliers ou de naviguer sur des terrains irréguliers en partant du principe que le sol est plat et donc en s'adaptant de manière réactive aux perturbations [Ble18; Sie21]. Cependant, ces contrôleurs sont insuffisants lorsque l'environnement contient des trous ou des obstacles trop hauts pour être escaladés.

Les environnements complexes nécessitent la planification du mouvement d'un robot plusieurs pas à l'avance (dans un horizon établi). Malheureusement, le calcul du mouvement de



FIGURE 11 – Abstract graphique du papier [Ris22] pour la conférence IROS

tous les pas à venir avant l'achèvement du pas actuel impose de fortes contraintes de temps de calcul [Hau08 ; Esc13]. La planification simultanée du mouvement et de l'emplacement des pas nécessite de s'attaquer à la combinatoire associée au choix discret des contacts, un problème de complexité exponentielle [Bre06], ce qui motive l'utilisation d'hypothèses simplificatrices. Un choix populaire consiste à considérer uniquement la dynamique du centre de masse d'un système [Car18 ; Fer20], ce qui ne garantit pas la faisabilité du mouvement planifié. De plus, ces approches restent non linéaires et exigeantes en termes de calcul.

L'alternative que nous présentons dans ce chapitre est de planifier les emplacements des pas avant de calculer le mouvement [Dei14 ; Gri19a ; Ton20]. Le découplage entre la planification des contacts et la génération des mouvements rend incertaine la faisabilité du plan. De plus, la fréquence de planification (10 Hz) reste incompatible avec des adaptations réactives aux perturbations externes ou aux erreurs d'estimation d'état. Il est donc nécessaire que le contrôleur du robot adapte localement le plan, mais "pas trop" car nous savons qu'une déviation de ce plan peut conduire aux défaillances qui ont justifié la planification en premier lieu. Que veut dire "pas trop"? **Notre problématique de recherche est de trouver le compromis entre le suivi strict d'un plan et son adaptation réactive.**

En supposant que le monde est une collection de surfaces de contact potentielles convexes, nous faisons l'hypothèse que l'information clé donnée par le planificateur de contact est le choix des prochaines surfaces de contact sur lesquelles marcher. Nous considérons donc les surfaces sélectionnées comme une contrainte dure qui ne peut être remise en cause avant le calcul d'un nouveau plan. Cette hypothèse conduit à deux problèmes :

- La position exacte des prochains contacts doit être adaptée en temps réel au niveau du contrôle pour permettre de gérer l'évitement des collisions, les erreurs d'estimation d'état et d'autres perturbations.
- Le planificateur de contacts doit être capable de prédire la position future des contacts, afin que le plan de contact soit robuste aux modifications locales que le contrôleur y apportera.

Nous proposons une implémentation sous la forme d’un schéma de contrôle en trois étapes :

- La première étape présentée en Section 3.1.1 consiste en un planificateur de contact, formulé comme un programme mixte-entier (MIP) [Dei14]. Pour s’assurer que les décisions du planificateur sont alignées sur le comportement du contrôleur, le plan de contact minimise un coût adapté de l’heuristique de Raibert (2.3.2).
- La deuxième étape, formulée comme deux programmes quadratiques (QP) présentés en Section 3.1.2 et Section 3.1.3, optimise les emplacements des pas dans les surfaces choisies et les trajectoires des pieds de vol pour tenir compte des perturbations locales et éviter les collisions. Cette étape a été réalisée par Thomas Corbères.
- La troisième étape présentée en Section 3.2 est une adaptation du contrôleur de locomotion de Solo (présenté en Section 2.3.2), qui suit un mouvement centroïdal de référence en combinant un MPC sur un modèle réduit avec un contrôleur de corps-complet pour générer les consignes du contrôleur bas-niveau.

L’hypothèse de notre méthode est validée empiriquement avec le robot Solo, sur différents terrains complexes où le robot doit suivre une vitesse de référence envoyée en temps réel par un utilisateur avec une manette de jeu. Les 6 prochains pas du robot sont planifiés toutes les 160 ms, tandis que le contrôle du corps entier du robot est calculé à 1 kHz. Des tests répétés en simulation démontrent la contribution de chaque partie du contrôleur à la robustesse globale.

Collaboration

Ce travail a été réalisé en collaboration avec Thomas Corbères, doctorant à l’Université d’Édimbourg. Il a formulé les deux QPs présentés dans les sections 3.1.2 et 3.1.3. Les expériences sur le robot ont été réalisées conjointement. Nous avons donc choisi d’être tous deux premiers auteurs du papier présenté à la conférence IROS [Ris22].

3.1 Planification réactive en environnements complexes

À partir de l’état courant du robot, d’une référence de vitesse de la base à trois dimensions et d’une décomposition de l’environnement en un ensemble de surfaces convexes, le contrôleur sélectionne d’abord les surfaces où un contact se produira sur un horizon de plusieurs pas. Les positions de contact dans les surfaces choisies sont ensuite adaptées pour le prochain pas, ainsi que la trajectoire qui amène le pied de vol au contact. Les surfaces de contact sont sélectionnées chaque pas pour les pas suivant, toutes les 160 ms dans notre cas. La trajectoire du pied de vol et les positions des contacts sont mises à jour à 1 kHz.

Les modules de sélection de surface et de positionnement du contact utilisent tous deux l’heuristique de Raibert. Nous présenterons dans cette section le planificateur de contacts, puis l’adaptation réactive de la position du prochain pas, et enfin le calcul de la trajectoire du pied de vol.

Le schéma global du contrôleur est présenté Fig. 12. On voit les trois blocs de planification réactive en bleu, suivis du contrôleur adapté de Solo (haut-niveau et bas-niveau) qui sera présenté dans la partie suivante.

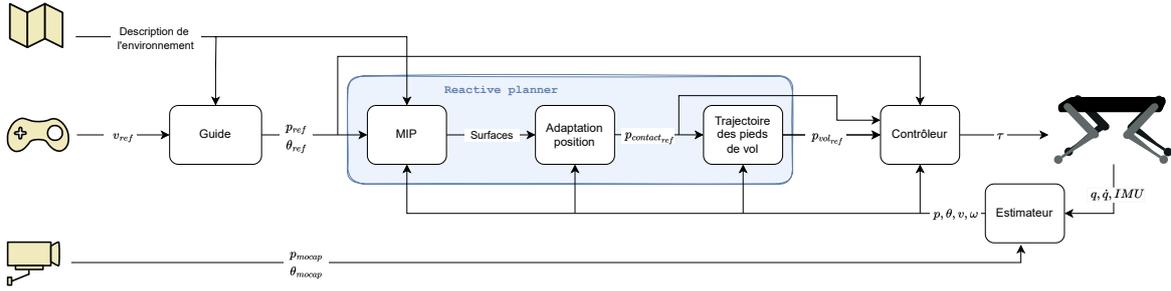


FIGURE 12 – Contrôleur pour la planification réactive sur le robot Solo

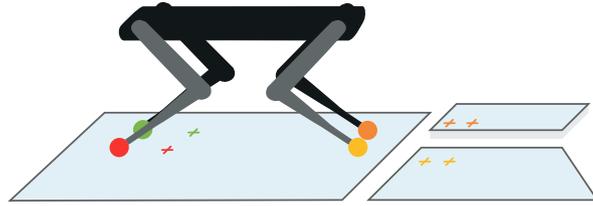


FIGURE 13 – Tâche de planification des contacts

3.1.1 Choix des surfaces de contact

Le planificateur de contacts résout le problème combinatoire de la sélection de surfaces de contact pertinentes pour le mouvement tout en appliquant des contraintes cinématiques pour garantir la faisabilité. Ce problème peut être formulé sous la forme d'un programme mixte-entier (MIP) [Dei14]. Nous utilisons une implémentation sous licence ouverte (*open-source*) pour la résolution, appelée SL1M [Ton20; Son21], modifiée pour que la faisabilité quasi-statique ne soit pas vérifiée. L'horizon n est le nombre de contacts à choisir. Par exemple dans le cas d'un trot (où deux contacts sont créés à chaque nouveau pas) $n = 4$ correspond à deux pas.

Les figures 11 et 13 illustrent ce problème : le monde est représenté comme un ensemble de surfaces convexes disjointes et le planificateur doit sélectionner une surface pour chacun des contacts de l'horizon.

Au cours de cette thèse, j'ai réécrit une grande partie de la bibliothèque SL1M [Ton20; Son21] pour pouvoir l'utiliser avec un robot quadrupède et de manière réactive.

Description de l'environnement

L'environnement est l'union de m surfaces de contact disjointes quasi plates¹. $\mathcal{S} = \bigcup_{j=1}^m \mathcal{S}_j$. Chaque surface \mathcal{S}_j est un polygone inscrit dans un plan 3D :

$$\mathcal{S}_j := \{\mathbf{p} \in \mathbb{R}^3 \mid \mathbf{S}_j \mathbf{p} \leq \mathbf{s}_j\} \quad (3.1)$$

1. c.-à-d. telles que leur cône de friction contient le vecteur gravité.

$\mathbf{S}_j \in \mathbb{R}^{h \times 3}$ et $\mathbf{s}_j \in \mathbb{R}^h$ sont respectivement une matrice et un vecteur définissant les demi-espaces délimitant la surface, avec h le nombre de demi-espaces.

Entrées et sorties du planificateur

Les entrées du planificateur sont les suivantes :

- L'état actuel du robot estimé par l'estimateur
- Une référence de vitesse 3D pour la base (translation en X-Y, angle de lacet) donnée par l'utilisateur grâce à une manette de jeu ou un contrôleur de plus haut niveau
- Une liste des positions/orientations prévues pour la base sur l'horizon, au moment de la création du contact dont les détails du calcul seront présentés Section 3.2.2
- Une liste des surfaces de contact potentielles pour chacun des contacts de l'horizon, dont l'obtention est détaillée Section 3.2.3

Le planificateur fournit en sortie les prochaines positions de contact à atteindre par le robot et le *plan de contact*. Le plan de contact est donné sous la forme d'une liste de surfaces de contact $\mathcal{S}_k^i \in \mathbb{R}^3, 1 \leq i \leq l$, avec l le nombre total d'effecteurs en contact et k décrivant la k -ième phase de contact de l'horizon. Dans notre cadre, suivant notre hypothèse de recherche, nous ne considérons que les surfaces de contact sélectionnées.

Formulation MIP du problème

Nous définissons l'ensemble de n variables $\mathbf{a}_i = [a_i^1, \dots, a_i^m] \in \{0, 1\}^m, 1 \leq i \leq n$, telles que $a_i^j = 1$ implique que le i -ème contact créé appartient à la j -ème surface de contact et $a_i^j = 0$ implique que le i -ème contact n'est pas dans avec la j -ème surface de contact ².

Nous résolvons le MIP suivant :

$$\begin{aligned}
 \text{cherche } & \mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_n] \in \mathbb{R}^{3 \times n} \\
 & \mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \{0, 1\}^{n \times m} \\
 \text{min } & \ell(\mathbf{P}) \\
 \text{tel que } & \mathbf{P} \in \mathcal{C} \cap \mathcal{F} \\
 & \forall i \in [1, n], \forall j \in [1, m] : \\
 & \quad \text{card}(\mathbf{a}_i) = 1 \\
 & \quad \mathbf{S}_j \mathbf{p}_i \leq \mathbf{s}_j + M(1 - a_i^j) \mathbf{1}
 \end{aligned} \tag{3.2}$$

où ℓ est une fonction objectif quadratique à minimiser et **card** est la cardinalité d'un vecteur, c'est-à-dire le nombre d'entrées différentes de 0. La contrainte **card**(\mathbf{a}_i) = 1 garantit qu'à chaque étape la position prévue se trouve exactement sur une surface de contact. Les variables \mathbf{a}_i indiquent les surfaces de contact qui ont été sélectionnées par le planificateur. $M \in \mathbb{R}$ est un nombre suffisamment grand permettant de satisfaire la dernière contrainte lorsqu'une surface n'a pas été choisie.

2. Pour éviter des indices supplémentaires, nous supposons sans perte de généralité que l'ensemble des surfaces de contact potentielles est le même sur l'horizon.

De plus, \mathcal{C} est un ensemble de contraintes initiales défini par l'utilisateur et \mathcal{F} est un ensemble de contraintes de faisabilité. Ces ensembles sont supposés être convexes. Dans ce travail, les contraintes initiales imposent que les positions des effecteurs non mobiles correspondent à l'état actuel après une étape. Les contraintes de faisabilité garantissent que la distance relative entre chaque effecteur est linéairement contrainte afin d'approcher les contraintes cinématiques [Ton18b].

L'heuristique de Raibert comme fonction de coût

Contrairement à [Ton20], nous proposons de formuler la fonction de coût ℓ comme la minimisation de la distance entre la position 2D du pied et une position 2D souhaitée :

$$\ell = \|\mathbf{p}_i^* - \mathbf{p}_{i,xy}\|^2 \quad (3.3)$$

La position souhaitée est l'approximation suivante de l'heuristique de Raibert :

$$\mathbf{p}_i^*(\mathbf{v}_{ref}) = \begin{bmatrix} p_{sh,i}^x + \frac{1}{2}t_{contact}v_{ref}^x \\ p_{sh,i}^y + \frac{1}{2}t_{contact}v_{ref}^y \end{bmatrix} \quad (3.4)$$

en utilisant les notations présentées dans Section 2.3.2.

Une fonction de coût supplémentaire est utilisée avec un poids dix fois plus faible pour limiter la distance entre le pied et l'épaule. La pénalisation de cette distance favorise les solutions où il y a une plus grande marge sur les contraintes cinématiques et permet ainsi de sélectionner une solution plus naturelle. Par exemple, lors de la montée d'un escalier, cela amènera le robot à changer de marche plus tôt, ce qui réduit le risque d'atteindre des singularités de jambe.

$$\ell_{sh} = \|\mathbf{p}_{sh,i} - \mathbf{p}_i\|^2 \quad (3.5)$$

3.1.2 Adaptation des positions des pieds

Étant donné la surface de contact choisie par le planificateur et l'état courant du robot, nous calculons les prochaines positions de contact $\mathbf{p}_i, 1 \leq i \leq n$ pour les pieds sur un horizon de n (pas nécessairement égal à l'horizon de planification des contacts). Nous supposons que les positions de contact optimales sont données par l'heuristique de Raibert, que nous étendons à 3 dimensions. Nous formulons un problème QP pour satisfaire au mieux les contraintes de surface tout en minimisant la violation de l'heuristique au sens des moindres carrés. Les variables de décision sont à la fois la vitesse de la base et les positions de contact. Chaque variable de position est définie comme un décalage $\alpha_i = [\alpha_i^x, \alpha_i^y, \alpha_i^z]^T \in \mathbb{R}^3$ par rapport à la position prédite par l'heuristique de Raibert :

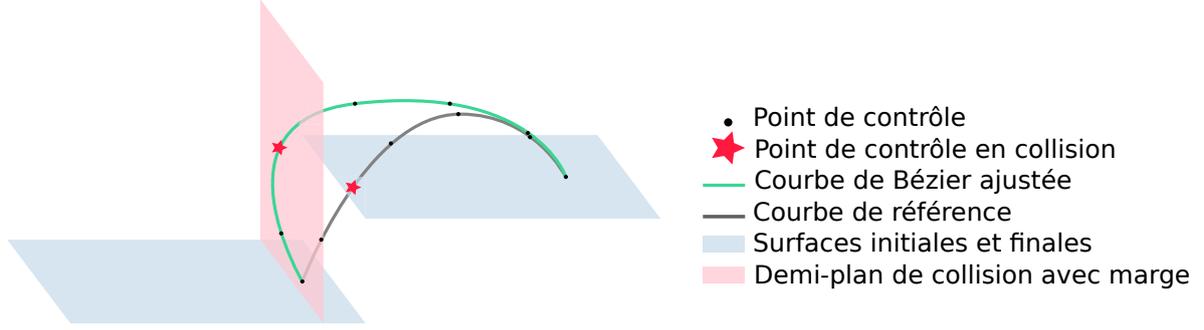


FIGURE 14 – Trajectoire du pied de vol adaptée pour éviter les collisions

$$\mathbf{p}_i = \begin{bmatrix} p_i^x(v_{ref}^x) + \alpha_i^x \\ p_i^y(v_{ref}^y) + \alpha_i^y \\ \alpha_i^z \end{bmatrix}$$

De même, la vitesse de référence \mathbf{v}_{ref}^+ est actualisée avec un décalage $\beta = [\beta^x, \beta^y]^T \in \mathbb{R}^2$ par rapport à la référence initiale \mathbf{v}_{ref} :

$$\mathbf{v}_{ref}^+ = \mathbf{v}_{ref} + \beta$$

Le QP qui en résulte s'écrit :

$$\begin{aligned} \min \quad & \frac{1}{2} (w_1 \|\beta\|^2 + w_2 \sum_{i=1}^n \|\alpha_i^{x,y}\|^2) \\ \text{tel que} \quad & \forall 1 \leq i \leq n, \quad \mathbf{S}_i(\mathbf{p}_i + \alpha_i) \leq \mathbf{s}_i \end{aligned}$$

Où la surface \mathcal{S}_i est la surface sélectionnée pour le i -ème contact, et w_1 et w_2 sont des poids définis par l'utilisateur, respectivement fixés à 1000 et 1.

3.1.3 Évitement des collisions pour les pieds de vol

La trajectoire d'un pied de vol $\mathbf{p}_i(t) : \mathbb{R} \mapsto \mathbb{R}^3$ est calculée comme un compromis entre une référence définie de manière unique $\mathbf{p}_{i,ref}(t)$ et l'ajustement nécessaire pour éviter les collisions avec l'environnement (Fig. 14). Les trajectoires sont des courbes de Bézier 3D de degré 7 définies dans la base de Bernstein.

La trajectoire de référence $\mathbf{p}_{i,ref}(t)$ d'un pied de vol i est présentée Section 2.3.2. Elle est ensuite discrétisée sur la durée du pas T en $n_T + 1$ échantillons avec un pas de temps t_k , $0 \leq t_k \leq T_k$.

On définit la trajectoire adaptée $\mathbf{p}_i(t)$ comme une courbe de Bézier de degré 7, de points de contrôle \mathbf{P}_i . À chaque temps t_k , on peut exprimer la courbe de Bézier comme une combinaison linéaire de ses points de contrôle :

$$\forall k \quad \exists \gamma_k \quad \text{tel que} \quad \mathbf{p}(t_k) = \gamma_k \mathbf{P}_i \quad (3.7)$$

Pour chaque surface de contact active pendant le mouvement (voir Fig. 14) et pour chaque échantillon $\mathbf{p}_{i,ref}(t_k)$, on vérifie si le segment reliant l'échantillon à ses voisins est en collision. Si c'est le cas, une contrainte de collision définie par le demi-espace traversé (Fig. 14) est créée pour $\mathbf{p}_i(t_k)$ (en utilisant une petite marge de sécurité). Ces contraintes sont regroupées en une matrice \mathbf{G} et un vecteur \mathbf{g} qui contraignent linéairement les points de contrôle \mathbf{P}_i . Nous formulons finalement un **QP** qui cherche à minimiser la distance entre $\mathbf{p}_i(t_k)$ et $\mathbf{p}_{i,ref}(t_k)$ pour tout k tout en garantissant le respect des contraintes de collision :

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{1}{2} \sum_{k=0}^{n_T} \|\mathbf{p}_i(t_k) - \mathbf{p}_{i,ref}(t_k)\|^2 \\ \text{tel que} \quad & \mathbf{p}_i(0) = \mathbf{p}_{i,ref}(0) \quad \dot{\mathbf{p}}_i(0) = \dot{\mathbf{p}}_{i,ref}(0) \quad \ddot{\mathbf{p}}_i(0) = \ddot{\mathbf{p}}_{i,ref}(0) \\ & \mathbf{p}_i(T) = \mathbf{p}_{i,ref}(T) \quad \dot{\mathbf{p}}_i(T) = \dot{\mathbf{p}}_i(T) = \mathbf{0} \\ & \mathbf{G}\mathbf{P}_i \leq \mathbf{g} \end{aligned}$$

3.2 Adaptation du contrôleur de Solo

Si le **MPC** présenté Section 2.3.2 et le contrôleur corps-complet ne sont pas modifiés pour nos expériences, ce n'est pas le cas de l'estimateur et de la génération de la trajectoire de la base sur l'horizon. Solo étant dépourvu de caméras, nous avons modélisé l'environnement et utilisé un système de capture de mouvements (*Motion Capture* ou **MoCap**) pour permettre la localisation dans cet environnement. De plus pour obtenir la référence en élévation de la base du robot, ainsi que ses angles de roulis et tangage, nous avons utilisé une carte d'élévation. Finalement les surfaces potentielles pour chaque pas ont été trouvées en regardant les surfaces accessibles depuis un endroit donné.

3.2.1 Estimateur

L'**IMU** utilisée sur le quadrupède Solo comprend un filtre de Kalman étendu qui estime la vitesse angulaire, l'orientation et l'accélération du corps sans gravité. Avec notre configuration, la position et l'orientation en lacet dans le monde ne sont pas des quantités observables et dériveraient si nous devions les évaluer par intégration. Pour cela, nous utilisons un système de **MoCap**.

Les données de la **MoCap** ne peuvent cependant pas être utilisées directement pour l'estimation de la vitesse (à cause des délais, de la fréquence limitée et des mauvaises dérivées). Les

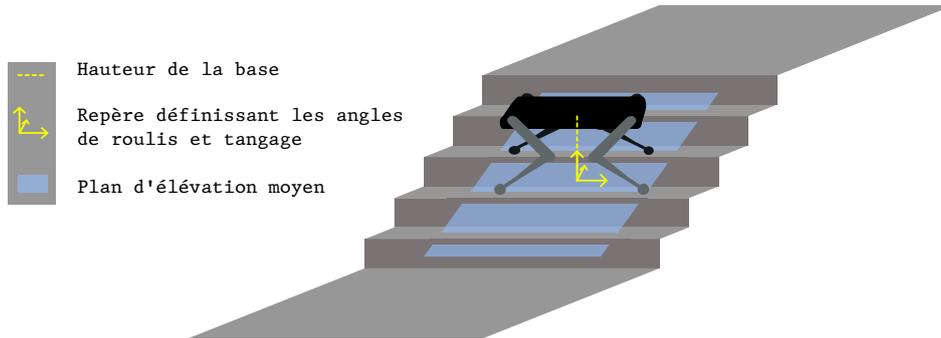


FIGURE 15 – Définition de la position de référence de la base

estimations de la vitesse linéaire $\mathbf{v} \in \mathbb{R}^3$ proviennent donc d'une approche linéaire découplée [Fla17] qui combine les données MoCap et IMU avec un filtre complémentaire [Car15b]. Pour mettre à jour l'estimation de la vitesse au temps $t + 1$, l'accélération précise de l'IMU $\dot{\mathbf{v}}_{IMU}^t$ est intégrée et lentement dé-biaisée par les mesures de vitesse de la MoCap \mathbf{v}_{MoCap}^t :

$$\mathbf{v}^{t+1} = \alpha(\mathbf{v}^t + \Delta t \dot{\mathbf{v}}_{IMU}^t) + (1 - \alpha)\mathbf{v}_{MoCap}^t \quad (3.8)$$

avec α un poids proche de 1 (0,97 dans notre cas).

3.2.2 Guide de trajectoire

Les positions et vitesses sur le plan horizontal ($X - Y$) et la rotation en lacet sont obtenues comme précédemment dans la Section 2.3.2.

La hauteur de la base et les angles de roulis et de tangage sont quant à eux estimés à partir de l'environnement, qui est représenté par une carte d'élévation dans ce module. Nous supposons que l'orientation du robot peut être définie à partir de la hauteur moyenne des points de la carte d'élévation les plus proches (moins de 20 cm de la position $X - Y$) actuelle). Un plan 3D est alors calculé en minimisant la distance avec ces points au sens du moindre carré. Les valeurs de roulis et de tangage du plan sont utilisées pour l'orientation de la base. La Fig. 15 illustre ce procédé.

La position $X - Y$ de la base, projetée sur ce plan, permet d'obtenir la position Z de la base (décalée d'une constante). La vitesse verticale est obtenue en différenciant les positions Z , tandis que les vitesses de roulis et tangage sont fixées à 0.

3.2.3 Surfaces potentielles pour chaque pas

Pour chaque surface de contact à trouver, on regarde la position prévue de la base donnée par le guide. À partir de cette position prévue, les zones atteignables par le pied sont approximées [Gei19]. En regardant l'intersection entre ces zones et les surfaces de l'environnement, on déduit les surfaces de contact potentielles pour ce pied. Cette méthode est utilisée pour chaque

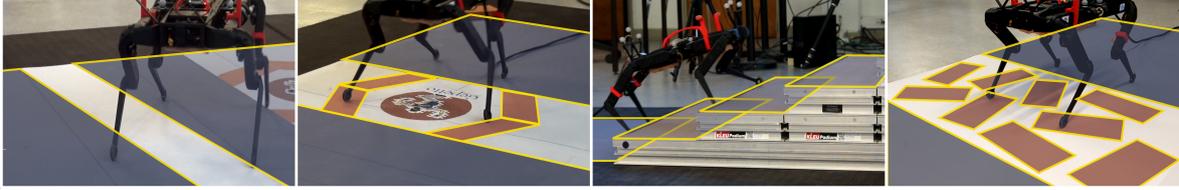


FIGURE 16 – Différents scénarios pour les expériences de planification réactive [Sol]

contact de l’horizon et permet de réduire la combinatoire du planificateur en ne cherchant que parmi les surfaces atteignables et non dans tout l’environnement.

3.3 Étude expérimentale

Nous avons testé notre hypothèse de recherche et notre schéma de contrôle sur le quadrupède Solo. En simulation et lors d’expériences, le robot a été mis au défi dans des environnements complexes. Le contrôleur s’est avéré capable de générer des consignes pour une marche stable et robuste sur des terrains avec des trous, des ponts, des marches ou des escaliers. Plusieurs scénarios ont été conçus pour démontrer la robustesse du contrôleur.

3.3.1 Comment évaluer le contrôleur

Une évaluation qualitative du schéma de contrôle a d’abord été effectuée à la fois en simulation et dans des expériences réelles. Des résultats plus précis ont ensuite été obtenus en simulation pour mettre en évidence l’utilisation d’un module ou d’un autre et évaluer l’influence d’un paramètre donné. Des environnements spécifiques ont été conçus en simulation et construits pour des expériences réelles lorsque cela était possible (Fig. 16) :

- **Terrain plat** utilisé en poussant le robot pour évaluer la robustesse générale du contrôleur aux perturbations.
- **Crevasse** au sol jusqu’à 15 cm de largeur, utilisé pour montrer la capacité d’éviter une zone interdite tout en maintenant l’équilibre pour n’importe quel angle d’approche.
- **Pont**, deux passerelles minces non parallèles (10 cm) sur lesquelles un emplacement précis des pieds est requis et où les positions relatives des pieds sont difficiles à définir.
- **Escalier**, droit ou en colimaçon, avec plusieurs hauteurs de marches et largeurs. Des trous entre les marches ont été ajoutés, ainsi que des irrégularités dans la hauteur des marches. Les escaliers permettent de montrer l’utilité du module d’évitement de collision.
- **Pavés**. Un ensemble de pavés permet de traverser une zone impraticable, comme le feraient des pierres dans une rivière. Cela illustre la nécessité du plan de contact et permet de tester si le plan de pas est adapté assez rapidement pour s’adapter aux changements rapides de la vitesse de référence. Dans des expériences réelles, les pavés étaient des briques de 22 cm sur 11 cm. Elles étaient soit plates, soit d’une hauteur de 3 ou 6 cm et étaient placées aléatoirement pour ajouter plus ou moins de difficulté. Une carte du terrain était toujours réalisée et fournie au contrôleur.

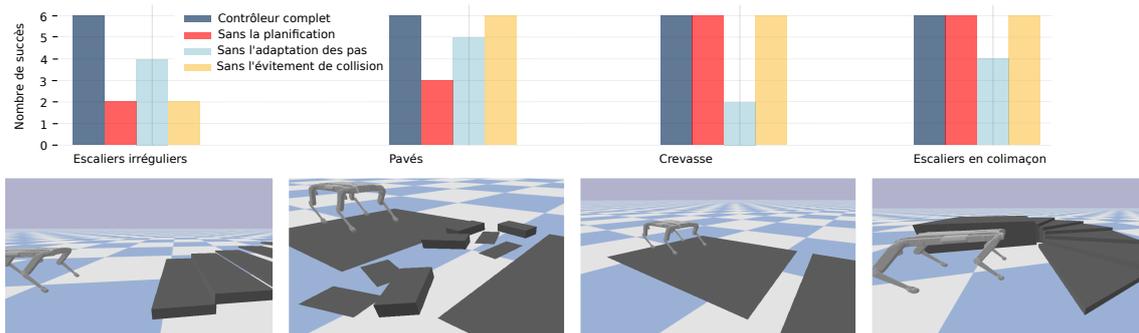


FIGURE 17 – Répétition des tests avec un contrôleur modifié pour évaluer l’avantage de chaque module. Pour chaque scénario, six profils de vitesse, résumant la gamme des capacités du robot, ont été exécutés en simulation et les résultats de chaque test sont rapportés ici.

3.3.2 Résultats qualitatifs

Tous les scénarios ont été essayés avec succès en simulation, avec différentes vitesses de référence et positions initiales du robot. Lors des expériences réelles, le robot parvenait à se remettre de fortes perturbations sur terrain plat, pouvait franchir un trou matérialisé par une ligne tracée au sol et tourner au-dessus. Il marchait sur un pont qui nécessitait d’éloigner son pied gauche du droit. Sur les pavés, le robot a réussi à placer correctement ses pieds et à maintenir son équilibre même avec des variations rapides de la vitesse de référence, comme des consignes pour s’arrêter ou tourner. Les escaliers étaient montés avec des angles d’approche variables et le robot pouvait se retourner dessus sans tomber. Des vidéos montrent ces expériences [Sol].

3.4 Analyse quantitative

3.4.1 Influence de chaque module sur la robustesse

Pour évaluer la contribution de chaque composant à la robustesse du contrôle, nous avons créé des contrôleurs alternatifs dans lesquels un module avait été désactivé. Ils ont ensuite été comparés les uns aux autres en simulation, sur des scénarios conçus pour mettre en évidence les cas limites (Fig. 17). Quatre scénarios différents ont été sélectionnés, et pour chacun d’eux, six profils de vitesse différents ont été appliqués. Ceci a permis d’évaluer la robustesse de notre méthode et de mettre en évidence l’influence de chacun des modules, comme décrit dans les sections suivantes.

3.4.2 Utilité de la planification des contacts

Nous avons remplacé le planificateur de contact par une heuristique naïve où la surface sélectionnée est la plus proche de la position prédite par l’heuristique 2D de Raibert (la planification est donc restreinte à un horizon $n = 1$). Dans les scénarios de crevasse et d’escalier en colimaçon, le planificateur de contact est remplacé par l’heuristique sans aucune incidence

sur les taux de réussite (Fig. 17). La nécessité de la planification des contacts est mise en évidence dans les scénarios d’escaliers irréguliers et de pavés. Les cas d’échec surviennent lorsque la projection de l’heuristique 2D se trouve sur une surface qui ne peut être atteinte à partir de l’état actuel du robot, alors que ce choix de surface est rejeté par le planificateur de contact, justifiant la nécessité de tester plusieurs surfaces candidates (et donc d’utiliser notre planificateur).

3.4.3 La nécessité de l’adaptation du pas

Sur un sol plat, la planification de l’emplacement des pas à une fréquence élevée (1 kHz) s’est avérée utile pour maintenir le robot stable et réagir rapidement aux perturbations. Pour évaluer l’importance du module d’adaptation des pas, nous avons essayé le schéma de contrôle en utilisant directement les positions des pas calculées par le planificateur de contact. Comme le plan de contact n’est calculé qu’une fois par pas, les positions des contacts sont alors fixes pendant toute la durée du pas. La figure 17 illustre que dans la plupart des cas, ne pas adapter la position des pas conduit à des échecs du contrôleur.

Nous avons également calculé la distance entre les positions de contact prédites par le planificateur de contact et les positions de contact finales atteintes sur quatre scénarios différents. Le tableau 3.1 montre que la distance moyenne est faible (< 5 mm), ce qui confirme que la position prédite est pertinente dans le cas nominal. La mesure de la distance maximale varie de 1 à 4 cm dans selon les scénarios. Elle est alors du même ordre de grandeur que la longueur des pas. Cela illustre une fois de plus l’importance d’adapter de manière réactive la position des pas pour maintenir l’équilibre.

TABLE 3.1 – Distance entre la position de contact prévue et la position adaptée

Scénario	Distance moyenne (m)	Distance maximale (m)
Plat	0.0018	0.012
Crevasse	0.0019	0.026
Pont	0.0021	0.025
Pavés	0.0037	0.043

3.4.4 Évitement de collisions

De manière surprenante, la désactivation de l’évitement de collisions pour les pieds de vol n’entraîne pas une augmentation du taux d’échec dans trois des quatre environnements considérés (Fig. 17). L’environnement des escaliers irréguliers est composé de marches qui se chevauchent et sont séparées par des trous. Le pied peut alors se coincer entre deux marches. Avec le module d’évitement des collisions, la trajectoire de l’effecteur évite les bords de l’escalier et recule dans un premier temps, comme le montre la figure 14 et la vidéo, ce qui permet d’éviter des chutes. Ce scénario met en évidence la nécessité d’éviter les collisions dans de tels cas.

3.4.5 Effet de la longueur de l’horizon du planificateur de surface

La longueur de l’horizon n du planificateur paraît intuitivement crucial. Cependant, dans tous les scénarios énumérés dans la Section 3.3.1, après avoir défini une position cible et comparé le taux de réussite obtenu avec différentes valeurs de n , aucune différence significative n’a été observée. La planification avec un horizon de deux pas semble donc suffisante pour éviter les cas d’échec observés sans le planificateur dans la section 3.4.2. Comme prévu, le temps de planification augmente avec l’horizon et un horizon $n = 8$ (4 pas pour du trot) était trop coûteux pour le calcul en temps réel sur des scénarios avec beaucoup de surfaces potentielles comme les pavés.

TABLE 3.2 – Effet de la longueur de l’horizon de planification sur le taux de succès et le temps de calcul

Longueur de l’horizon	Temps de calcul (s)	Temps de calcul maximal (s)	Taux de succès
4	0.026	0.065	100%
6	0.045	0.141	100%
8	0.058	0.236	100%
10	0.092	0.447	100%
20	0.303	2.218	72%

Les résultats présentés dans le tableau 3.2 montrent que la longueur de l’horizon n’a aucun impact sur le taux de réussite, sauf pour un horizon de 10 pas ($n = 10$). Ceci est dû à la façon dont la trajectoire du guide est calculée. En effet, la hauteur du guide est définie par rapport à un ajustement local de l’environnement qui n’est plus valable sur des horizons lointains. D’autre part, le temps de calcul du planificateur de contact augmente rapidement avec le nombre de pas planifiés et dépasse la durée d’un pas (160 ms) pour plus de 3 pas planifiés, ce qui signifie que le plan ne peut être mis à jour à chaque pas.

3.4.6 Temps de calcul de chaque module

Pendant les expériences sur le robot, nous avons mesuré le temps de calcul de chaque module pour nous assurer qu’ils étaient conformes aux exigences en matière de temps réel. Les résultats sont présentés dans le tableau 3.3. Le temps de calcul maximal du planificateur de contact de 137 ms est inférieur à 160 ms, la durée d’un pas. Les deux QP des modules d’adaptation du pas et d’évitement des collisions sont résolus assez rapidement pour être calculés à 1 kHz dans la boucle de contrôle principale.

Conclusion

Nous avons présenté une méthode complète pour la planification et le contrôle en temps réel de la locomotion d’un robot à pattes sur des terrains difficiles. Ce travail a conduit à la rédaction d’un article, présenté en conférence internationale [Ris22].

TABLE 3.3 – Temps de calcul des différents modules du contrôleur

Module	Temps moyen (ms)	Temps maximal (ms)
Guide	0.122	0.238
Planificateur	58.4	137.4
Adaptation des pas	0.0270	0.0825
Évitement de collision	0.099	0.656
Contrôleur adapté	2.00	12.01

Notre contrôleur planifie une séquence de contacts réalisable pour le robot Solo à environ 10 Hz, tandis que les positions des contacts et les trajectoires des pieds de vol sont adaptées localement à 1 kHz, en synchronisation avec le contrôle de bas niveau du robot. Nous avons rapporté les détails de l’implémentation complète pour notre robot quadrupède Solo et le code est accessible sous licence ouverte. Nos expériences valident empiriquement notre approche sur des scénarios qui mettent en évidence la contribution de chaque module du schéma de contrôle. En réponse à notre problématique de trouver un compromis entre suivre strictement un plan de contact et l’adapter de manière réactive, les expériences suggèrent que s’engager sur un plan de surfaces de contact tout en permettant d’ajuster localement les positions des effecteurs permet d’obtenir un contrôle robuste et réactif.

Cependant, comme la surface de contact sélectionnée pour le pied en cours de vol ne peut pas changer, des échecs peuvent se produire si le robot est poussé. Une re-planification de la prochaine surface de contact pourrait être étudiée pour empêcher une chute dans ce cas. Les mêmes contraintes de temps de calcul limitent également l’exhaustivité de notre méthode d’évitement des collisions, car les seules surfaces vérifiées pour les collisions sont celles impliquées dans la création du contact, ce qui empêche le robot d’enjamber les obstacles de manière robuste. Vérifier tous les obstacles susceptibles d’entrer en collision serait une autre piste d’amélioration, éventuellement en réduisant la fréquence du QP d’évitement des collisions.

Les développements futurs pourraient également inclure l’adaptation de la trajectoire de base au plan de contact. Si le planificateur de contact ne parvient pas à trouver un plan faisable (si un obstacle empêche le robot d’aller dans la direction de la vitesse de référence par exemple), le robot tombe. Ce problème est actuellement traité en considérant que la vitesse de référence est donnée par un utilisateur qui a une connaissance de l’environnement et qui n’enverra pas le robot vers des zones impossibles à atteindre. Cependant, le planificateur de contact pourrait être étendu pour produire une trajectoire de base adaptée afin de résoudre ce problème.

L’utilisation de planification réactive permet de choisir les emplacements des contacts, mais n’offre pas la possibilité d’optimiser leur durée. Une telle amélioration pourrait permettre de gagner en stabilité sur des terrains difficiles. L’utilisation d’un MPC prenant en compte la dynamique complète du robot pourrait aussi être utilisé pour remplacer les modules d’adaptation de la position du prochain contact et de la trajectoire du pied de vol.

MPC avec optimisation de la durée des pas

Contents

4.1	Présentation de l'OCP	54
4.1.1	Où il est question de chaîne de Markov	55
4.1.2	Modèle inertiel rigide avec état augmenté	56
4.1.3	Coûts	57
4.1.4	Contraintes	58
4.2	Résultats de marche en simulation	59
4.2.1	Utilisation de la matrice de contacts pour changer la durée des pas	60
4.2.2	Performances sans perturbations	60
4.2.3	Performances avec force perturbatrice	61
4.3	Apprentissage pour le démarrage à chaud de Crocodyl	62
4.3.1	Génération des données	62
4.3.2	Réseau de neurone et apprentissage	64
4.3.3	Résultats de l'apprentissage	65
4.4	Simulation avec Crocodyl	68

Introduction

Dans ce chapitre, on s'intéresse à un autre paramètre de la marche : la durée des pas. Au cours d'une marche sur sol plat sans perturbation, on peut se contenter de fixer la durée des pas à une valeur bien définie. En revanche s'il y a des perturbations extérieures ou si le terrain n'est pas régulier, il semble intéressant de pouvoir adapter de manière réactive la durée des pas. Notre intuition nous dit que des pas lents peuvent permettre de dépenser moins d'énergie, mais que pour se remettre d'une perturbation il peut être nécessaire de faire quelques pas rapides pour créer des contacts au bon endroit.

La littérature s'intéresse au problème de l'adaptation réactive des durées des pas depuis plusieurs années. Les premiers travaux ont été réalisés pour des robots humanoïdes en utilisant le modèle LIPM [Aft12; Kry15]. La résolution d'un QP par un MPC a ainsi permis l'optimisation simultanée des positions et durées des pas [Her10a]. Le contrôle du DCM a été aussi utilisé [Kha16; Gri17] pour adapter les durées des pas. Certains de ces travaux

réalisent une optimisation en plusieurs temps [Kry15; Kha16]. Une telle décomposition du problème est utilisée par l'algorithme TOPP, qui optimise dans un premier temps un chemin du CoM puis le paramétrise temporellement [Pha14; Car16a]. Les méthodes développées pour des robots quadrupèdes diffèrent légèrement. On retrouve toutefois des optimisations en plusieurs étapes [Far16]. Plusieurs travaux proposent des formulations au sein d'un problème d'optimisation unique, mais sans l'intégrer dans un MPC [Car18; Win18; Pon18]. Neunert [Neu17] et Tassa [Tas12] utilisent quant à eux des modèles de contact dérivables par morceaux (avec une composante élastique) pour ne pas avoir à spécifier la séquence de contacts dans le MPC corps-complet. Finalement le MPC corps-complet proposé par Katayama [Kat22] permet l'optimisation des durées, mais pas des positions des pas.

Nous proposons dans ce chapitre, Section 4.1, la formulation d'un OCP unique utilisant le modèle inertiel rigide avec optimisation des positions et durées des pas. Cette formulation est contrainte par la nécessité d'avoir un problème Markovien (ce qui garantit l'application directe du principe de Bellman) pour pouvoir être utilisé avec un solveur par DDP (présenté Section 2.2.3). En relâchant la contrainte d'efficacité calculatoire (durée de résolution et nombre d'itérations), nous montrons dans la Section 4.2 la capacité de ce contrôleur à produire une marche robuste même en présence de perturbations. Nous proposons ensuite dans la Section 4.3 l'apprentissage d'une "mémoire du mouvement" pour le démarrage à chaud d'un solveur DDP, dont nous évaluons la qualité dans la Section 4.3.3. Finalement nous présentons dans la Section 4.4 les résultats lors de tests de marche en simulation. Bien qu'améliorés par la mémoire du mouvement, ces résultats restent insuffisants pour la réalisation d'expériences sur le robot.

Encadrement

Ce travail a été réalisé dans le cadre d'une visite au laboratoire d'ingénierie industrielle de l'Université de Trente en Italie et Andrea Del Prete a participé à l'encadrement pour ce chapitre.

4.1 Présentation de l'OCP

Nous allons commencer par présenter la formulation du problème de contrôle optimal. Les contraintes en temps de calculs associées à une optimisation dans un MPC résolu en temps réel nous ont poussé à chercher une formulation compatible avec une résolution par DDP, que nous appellerons formulation markovienne et que nous présentons section 4.1.1.

Pour optimiser les durées des pas, nous avons donc ajouté à l'état du modèle inertiel rigide les positions et durées des pas. Une difficulté de l'optimisation du temps vient du modèle de contact utilisé : nous devons spécifier pour chaque nœud de discrétisation de l'OCP les points de contact, qui définissent la dynamique. En optimisant la durée d'un pas on risque de se trouver avec des nœuds dont on ne connaît pas les points de contact actifs. Nous avons donc choisi d'assigner un nombre fixe de nœuds d'optimisation à chaque pas, mais de faire varier l'intervalle entre deux nœuds (qui est donc le pas d'intégration).

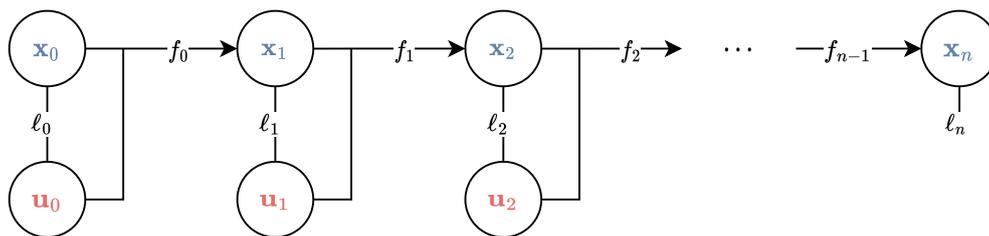


FIGURE 18 – Illustration du problème d’optimisation sous forme de chaîne de Markov

Cette approche a toutefois l’inconvénient de faire varier l’horizon d’optimisation, car le nombre de nœuds total du problème ne peut pas varier. Une solution triviale est alors de mettre à 0 les durées de chaque pas (et donc l’intervalle entre tous les nœuds), ce qui permet de supprimer complètement les coûts intégraux de l’OCP. Pour éviter cette solution, nous avons ajouté des contraintes sur les durées des nœuds et la durée totale de l’horizon, que nous présenterons dans cette partie.

4.1.1 Où il est question de chaîne de Markov

Pour respecter les contraintes en temps de calcul imposées par la résolution de l’OCP dans un MPC, nous prévoyons d’utiliser un solveur de résolution numérique fondé sur le DDP. Cet algorithme présenté section 2.2.3, repose sur une discrétisation de la forme 4.1.

$$\begin{aligned}
 \min_{\mathbf{x}_{0:n}, \mathbf{U}_{0:n-1}} \quad & \sum_{i=0}^{n-1} \ell_i(\mathbf{x}_i, \mathbf{u}_i) + \ell_T(\mathbf{x}_n) \\
 \text{tel que} \quad & \mathbf{x}_0 = x_0 \\
 & \forall i \in [t_0, t_{n-1}] : \\
 & \mathbf{x}_{i+1} = f_i(\mathbf{x}_i, \mathbf{u}_i)
 \end{aligned} \tag{4.1}$$

Cette discrétisation définit des nœuds d’optimisation, associés à un instant i , un état \mathbf{x}_i et un contrôle \mathbf{u}_i . Le problème doit respecter le principe d’optimalité de Bellman, c’est-à-dire qu’une sous-trajectoire d’une trajectoire optimale est optimale. Cela veut dire pour la discrétisation que si on enlève les m premiers nœuds de discrétisation du problème, la solution optimale du problème défini par les $n - m$ nœuds restants n’est pas modifiée. Pour prédire les états suivant à partir de l’état m on ne peut donc dépendre que de l’état \mathbf{x}_m , du contrôle \mathbf{u}_m , mais pas des états antérieurs. La chaîne des nœuds d’optimisation (qui contiennent états et contrôles) successifs est donc une chaîne de Markov (voir figure 18).

Concrètement dans le cadre d’un problème de contrôle optimal avec optimisation du temps, cela impacte la modélisation du problème. On pourrait vouloir utiliser un problème paramétrique pour optimiser le temps où un vecteur de paramètres p contenant les durées des pas serait optimisé en parallèle de l’état et du contrôle :

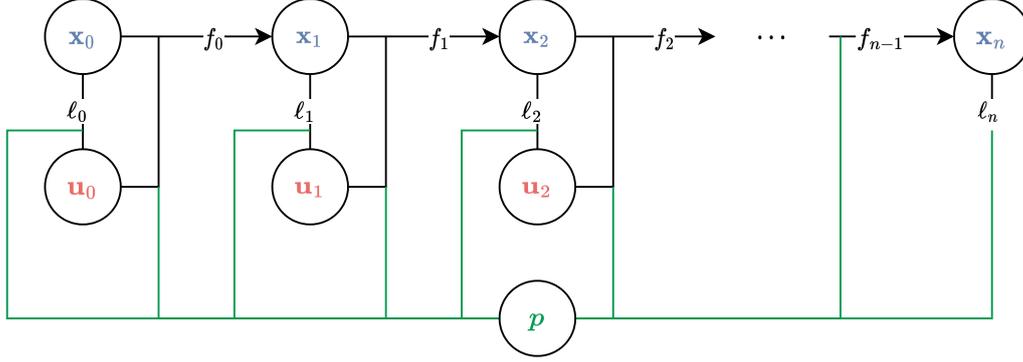


FIGURE 19 – Illustration du problème d'optimisation paramétrique

$$\begin{aligned}
 \min_{\mathbf{x}_{0:n}, \mathbf{U}_{0:n-1}, p} \quad & \sum_{i=0}^{n-1} \ell_i(\mathbf{x}_i, \mathbf{u}_i, p) + \ell_T(\mathbf{x}_n, p) \\
 \text{tel que} \quad & \mathbf{x}_0 = \mathbf{x}_0 \\
 & \forall i \in [t_0, t_{n-1}] : \\
 & \quad \mathbf{x}_{i+1} = f_i(\mathbf{x}_i, \mathbf{u}_i, p)
 \end{aligned} \tag{4.2}$$

Un tel problème, représenté figure 19, n'est clairement pas une chaîne de Markov. Dans un premier temps, nous allons donc proposer une formulation d'OCP permettant d'optimiser ces paramètres supplémentaires et gardant une structure explicitement Markovienne.

4.1.2 Modèle inertiel rigide avec état augmenté

Nous avons choisi d'utiliser le modèle inertiel rigide présenté Section 2.1. L'optimisation des positions et durées des pas ne peut se faire directement avec ce modèle, dont l'état est composé de la position, orientation et vitesses de la base. La dynamique est continue tant qu'il n'y a pas de changement de contact.

Pour optimiser les positions des contacts \mathbf{r}_i et durées des nœuds Δt , nous avons choisi de les ajouter à l'état du robot. Ils ne sont pas modifiés par la dynamique continue au cours d'un pas, mais à chaque nouveau pas, nous ajoutons une dynamique discrète qui change les contacts et la durée d'un nœud. La durée d'un pas est alors la somme des durées de nœuds qui le composent.

Le vecteur d'état contient donc l'état inertiel rigide du robot, les positions en 2D r_i de chacun des quatre pieds du robot (pour un pied qui ne touche pas le sol, c'est sa dernière position au sol) et la durée du nœud Δt :

$$\mathbf{x} = [\mathbf{p} \quad \Theta \quad \mathbf{v} \quad \omega \quad \mathbf{r}_1 \quad \dots \quad \mathbf{r}_4 \quad \Delta t]^T \tag{4.3}$$

en reprenant les notations de Section 2.1.2.

Tous les nœuds associés à la dynamique continue ont pour contrôle les forces de contact 3D \mathbf{f}_i entre le $i^{\text{ème}}$ pied en contact et l'environnement :

$$\mathbf{u} = [\mathbf{f}_1 \quad \dots \quad \mathbf{f}_{n_c}]^T \quad (4.4)$$

À chaque changement de pas, on utilise le contrôle suivant :

$$\mathbf{u} = [\mathbf{v}_1 \quad \dots \quad \mathbf{v}_{n_c} \quad \Delta t_{next}]^T \quad (4.5)$$

où v_i est la vitesse linéaire moyenne en 2D du pied qui crée un contact sur la durée de la phase de vol précédente et Δt_{next} est la durée des nœuds du prochain pas. Si D_i est la durée de la phase de vol précédent la création de contact pour le pied i , r_i^- sa position précédente et r_i^+ sa nouvelle position, on a : $r_i^+ - r_i^- = D_i * v_i$.

4.1.3 Coûts

Pour guider la résolution vers une bonne solution, nous utilisons un coût qui est la somme de plusieurs coûts, qui sont présentés dans cette partie. Pour simplifier la notation des poids de coûts, ils apparaîtront comme suit :

$$\|\xi\|_w^2 = \sum_{i=1}^n w_i \xi_i^2, \quad \mathbf{w} = (w_1, \dots, w_n) \quad (4.6)$$

où les poids w seront souvent réduits à un scalaire.

Le coût de fonctionnement $\ell_t(\mathbf{x}, \mathbf{u})$ est défini comme la somme des coûts suivants.

Suivi de référence

L'objectif principal du robot est de suivre une référence donnée par l'utilisateur grâce à une manette de jeu. Cette référence en vitesse est transformée en une série d'états de référence de la dynamique inertielle rigide sur tout l'horizon de l'OCP :

$$l_x(\mathbf{x}_t) = \Delta t \|\mathbf{x}_{inertiel} - \mathbf{x}^*\|_{w_x}^2 \quad w_x = [2. \quad 2. \quad 10. \quad 0.25 \quad 0.25 \quad 10. \quad 0.21_3 \quad 0. \quad 0. \quad 0.3]^T \quad (4.7)$$

où $\mathbf{x}_{inertiel}$ est la partie de l'état correspondant à l'état inertielle rigide du robot, \mathbf{x}_t^* est l'état inertielle rigide de référence à l'instant t .

Régularisation

Des coûts de régularisation sont ajoutés aux forces, à la durée des nœuds :

$$\begin{aligned}
l_u(\mathbf{u}) &= \Delta t \|\mathbf{u} - \mathbf{u}^*\|_{w_u}^2 & w_u &= 0.007 \\
l_t(\mathbf{x}) &= \|\Delta t - \Delta t^*\|_{w_t}^2 & w_t &= 0.1
\end{aligned} \tag{4.8}$$

où \mathbf{u}^* est une référence pour le contrôle qui contient les forces de compensation de gravité. Δt^* est une durée de référence pour un nœud qui correspond à un pas de durée intermédiaire.

On régularise aussi la position de chaque pied i :

$$l_p^i(\mathbf{x}) = \Delta t \|\mathbf{r}_i - \mathbf{P}_i^{x,y}\|_{w_p}^2 \quad w_p = 1.5 \tag{4.9}$$

où \mathbf{P}_i est la position des épaules du robot dans le repère global. Elle est approximée par :

$$\mathbf{P}_i = \mathbf{c} + \begin{bmatrix} a_i^x \cos \psi - a_i^y \sin \psi \\ a_i^x \sin \psi + a_i^y \cos \psi \\ a_i^y \theta - a_i^x \phi \end{bmatrix} \tag{4.10}$$

où $(a_i^x, a_i^y, 0)$ est la position de la $i^{\text{ème}}$ épaule dans le repère de la base.

Énergie

Nous ajoutons un coût pour favoriser les solutions plus économes en énergie. On suppose que les trajectoires avec de fortes accélérations des pieds seront plus énergivores. Pour des raisons de simplicité, ce n'est pas l'accélération des pieds de vol que l'on pénalise, mais la vitesse moyenne \mathbf{v} .

Nous utilisons le coût suivant sur les nœuds de fin de pas à la dynamique discrète :

$$\begin{aligned}
l_e(\mathbf{u}) &= \sum_{i=1}^{n_e} \|\mathbf{v}_i\|_{w_e}^2 & w_e &= 0.001 \\
l_v(\mathbf{u}) &= \frac{1}{(\Delta t + \epsilon)^2} & w_v &= 0.1
\end{aligned} \tag{4.11}$$

4.1.4 Contraintes

Les contraintes classiques d'un OCP centroïdal sont aussi ajoutées au problème, comme définies dans [Cor21] :

* L'état initial \mathbf{x}_{t_0} doit être égal à l'état courant du robot \mathbf{x}_0 :

$$\mathbf{x}_{t_0} = \mathbf{x}_0 \tag{4.12}$$

- * La distance entre les épaules et les points de contact est limitée (les pieds sont reliés par des jambes qui ont une longueur maximale) :

$$h_{\text{épaule}} \leq h_{\text{max}} \quad (4.13)$$

$$|\mathbf{r}_i - \mathbf{P}_i^{x,y}| \leq d_{\text{max}} \quad (4.14)$$

Les épaules sont fixes dans le repère de la base, donc la hauteur des épaules $h_{\text{épaule}}$ peut être calculée à partir de l'état. Les valeurs maximales h_{max} et d_{max} dépendent du robot, et en particulier de la longueur de ses jambes.

- * Pour éviter le glissement, les forces de contact doivent rester dans le cône de frottement qui est approximé par des inégalités linéaires pour chaque pied i en contact :

$$\begin{aligned} -\mu f_i^z &\leq f_i^x \leq \mu f_i^z \\ -\mu f_i^z &\leq f_i^y \leq \mu f_i^z, \\ f_{\text{min}}^z &\leq f_i^z \leq f_{\text{max}}^z \end{aligned} \quad (4.15)$$

où μ est le coefficient de frottement, $f_{\text{min}}^z/f_{\text{max}}^z$ is the minimum/maximum vertical contact force.

- * La variable d'ajustement de la durée des pas Δt étant aussi le pas d'intégration, les performances de l'intégration dépendent directement de sa valeur. Pour s'assurer que la dynamique discrète calculée reste réaliste, nous imposons une limite maximale à Δt :

$$\Delta t \leq \Delta t_{\text{max}} \quad (4.16)$$

- * Comme la plupart des coûts sont des coûts intégraux, le solveur va être attiré vers des solutions qui minimisent la durée totale de son horizon T . Pour limiter ce phénomène, nous ajoutons des bornes inférieures à T et à chaque Δt :

$$\begin{aligned} T_{\text{min}} &\leq T \\ \Delta t_{\text{min}} &\leq \Delta t \end{aligned} \quad (4.17)$$

La valeur choisie pour Δt_{min} est 0.0075 s, ce qui correspond à une durée minimale de pas de 0.06 s. La durée minimale de l'horizon T_{min} est définie par :

$T_{\text{min}} = \sum(\Delta t_{\text{min}}$ si la durée du pas est optimisée Δt sinon).

4.2 Résultats de marche en simulation

Pour évaluer les performances de ce contrôleur, nous avons effectué plusieurs tests en simulation avec le robot quadrupède Solo. L'OCF est formulé et résolu avec IpOpt, une librairie fondée sur la méthode des points intérieure [Wac06 ; Wri97], à une fréquence de 50 Hz. Le contrôleur, illustré figure 20 est inspiré du contrôleur de Solo présenté Section 2.3.2. La différence vient du fait que les positions des pieds sont désormais choisies par le MPC ainsi que les durées des pas.

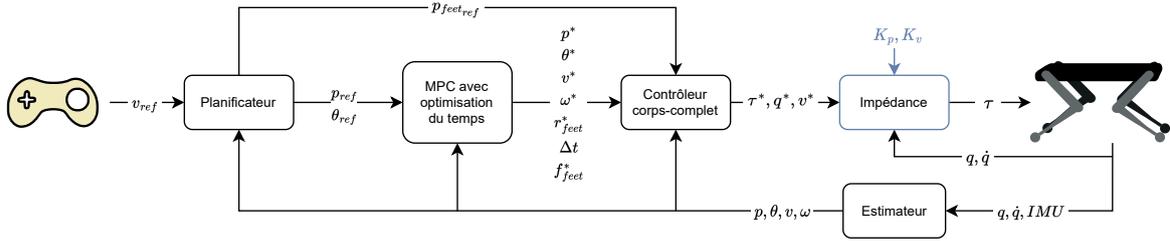


FIGURE 20 – Contrôleur pour l’optimisation des positions et durées des pas

4.2.1 Utilisation de la matrice de contacts pour changer la durée des pas

Le contrôleur repose sur l’utilisation d’une matrice de contacts pour connaître le statut des quatre pieds (phase de vol ou contact). Cette matrice est notamment utilisée pour connaître la séquence de contacts sur l’horizon de l’OCP et créer les nœuds d’optimisation correspondants. Elle contient quatre colonnes (une pour chaque pied) et autant de lignes qu’il y a de nœuds à la dynamique continue dans l’horizon de l’OCP. Dans notre cas, elle contient 16 lignes. Elle est mise à jour à chaque appel du MPC, toutes les 20 ms. À chaque appel, le premier nœud d’optimisation est supprimé (on avance dans l’horizon. On supprime donc la première ligne de la matrice de contacts) et un nouveau nœud est ajouté à la fin (On ajoute à la matrice de contacts la séquence de contacts correspondant au pas à optimiser à la fin de l’horizon). L’estimateur et le contrôleur corps-complet utilisent cette matrice pour connaître l’état courant des contacts. Le module de calcul de la trajectoire des pieds de vol l’utilise pour connaître la durée restante du pas courant.

À chaque début de pas, la matrice de contacts est modifiée pour prendre en compte le choix de durée du prochain pas de l’OCP. Un pas aura par défaut 8 nœuds de 20 ms, donc une durée totale de 160 ms. Le nombre de lignes du premier pas dépendra de la durée du pas choisi par l’OCP, comme illustré par le tableau 4.1.

4.2.2 Performances sans perturbations

Les premiers tests réalisés ont consisté à comparer le MPC avec optimisation du temps et le même MPC avec les durées des pas fixées dans une simulation de marche sans perturbation.

Tout d’abord nous avons observé l’évolution de la durée du pas en fonction de la vitesse de référence. Pour des vitesses de référence faibles, les pas durent longtemps (0.28 s). En revanche quand on augmente la vitesse de référence, la durée des pas diminue, jusqu’à se stabiliser autour de 0.14 s, ce qui est proche de la durée nominale de 0.16 s. La figure 21 illustre cette relation.

Nous avons ensuite fait varier la vitesse de référence en X , en Y et en lacet. Pour chacun des axes, nous avons cherché les intervalles limite de vitesse que les deux MPC pouvaient suivre et avons observé qu’ils étaient similaires. Nous avons ensuite comparé le suivi de la vitesse de référence. Nous n’avons pas observé de différence sensible sur le suivi de la vitesse de référence ou les couples.

TABLE 4.1 – Différentes matrices de contacts

$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$
--	--	---

Matrice à la fin d'un pas Matrice au début du pas suivant (Durée de 240 ms) Matrice au début du pas suivant (Durée de 80 ms)

On peut donc dire que sur terrain plat sans perturbation, optimiser la durée des pas n'apporte pas de gain de performance par rapport au même contrôleur sans optimisation de la durée.

4.2.3 Performances avec force perturbatrice

Nous avons ensuite réalisé des simulations avec perturbations. Pour cela nous avons appliqué une force extérieure sur la base du robot pendant une durée fixée.

Nous avons remarqué que le contrôleur avec optimisation du temps permet de rattraper une chute dans plus de situations que le contrôleur avec durées des pas fixées. De plus si on observe le suivi de la vitesse de référence, on voit que l'optimisation de la durée des pas permet de se remettre plus rapidement d'une perturbation, comme l'illustre la figure 22.

En traçant la durée des pas, on voit qu'elle diminue quand la force est appliquée, comme l'illustre la figure 23. Cela correspond à l'intuition selon laquelle il faut faire des pas rapides pour rattraper une chute, alors que sans perturbation on peut faire des pas plus lents.

Ces simulations montrent bien l'intérêt de l'optimisation de la durée des pas dans le cas d'une marche avec perturbations.

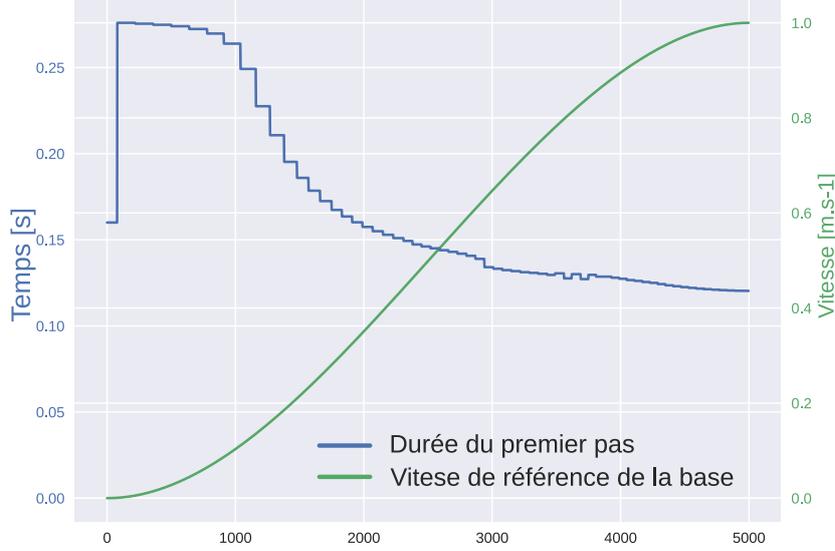


FIGURE 21 – Relation entre la durée des pas et vitesse de référence de la base

4.3 Apprentissage pour le démarrage à chaud de Crocodyl

Les temps de calcul d’IpOpt ne sont toutefois pas suffisants pour pouvoir passer de la simulation aux expériences sur le robot réel. Nous avons donc implémenté l’OCP en utilisant Crocodyl (présenté Section 2.2.5). Le DDP ne permettant pas d’utiliser d’autres contraintes que la dynamique, nous avons remplacé les contraintes par des coûts barrière associés à de forts poids (de l’ordre de 1000 pour le temps minimal).

Les premiers tests ont montré que sans démarrage à chaud adapté, Crocodyl avait tendance à choisir des solutions dans de mauvais minima locaux (par exemple avec une durée de pas négative), voire à diverger complètement. Nous avons donc décidé d’utiliser IpOpt pour générer des données qui sont ensuite apprises par un réseau de neurones, utilisé pour le démarrage à chaud de Crocodyl [Man18].

4.3.1 Génération des données

Nous avons commencé par déterminer le format de l’entrée et de la sortie du réseau de neurones.

On pourrait utiliser comme entrées l’état initial $\mathbf{x}_0 \in \mathbb{R}^{19}$, l’état de référence $\mathbf{X}^* \in \mathbb{R}^{12 \times 16}$ (qui dépend de la vitesse de référence $\mathbf{v}^* \in \mathbb{R}^6 = [v_x^* \ v_y^* \ 0 \ 0 \ 0 \ \omega_z^*]^T$ donnée par l’utilisateur), la matrice de contacts $G \in \{0,1\}^{4 \times 16}$ et la durée total du pas courant (en nombre de nœuds de 20 ms). Mais de nombreuses informations sont redondantes dans ces trois entrées.

Dans le cas d’un trot, la matrice de contact peut en réalité être représentée par deux entiers :

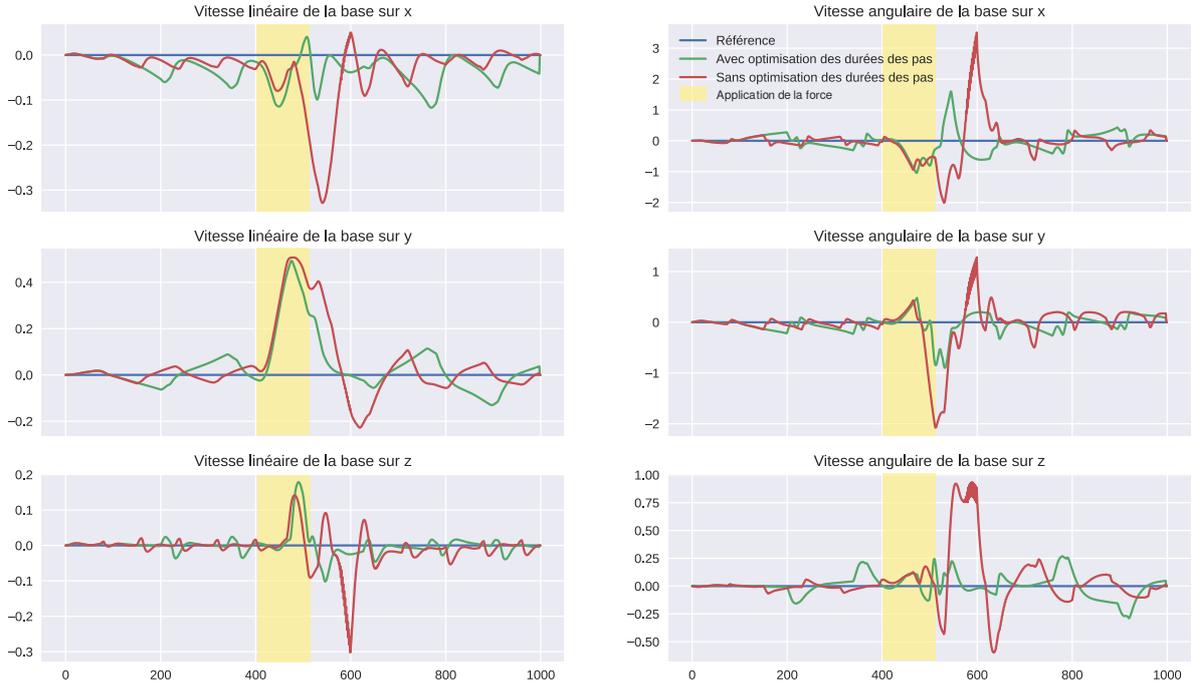


FIGURE 22 – Évolution de la durée des pas quand on applique une force de perturbation

- Un entier $g_0 \in \{0, 1\}$ qui indique la paire de jambes en contact au premier pas :
 $g_0 = 0 \Leftrightarrow \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}$ et $g_0 = 1 \Leftrightarrow \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}$.
- Un entier $n_{first} \in \{1, 15\}$ qui indique le nombre de nœuds pour le premier pas.

Le nombre de nœuds d'optimisation de l'OCP dépend de la matrice de contacts, et plus spécifiquement du nombre de pas à optimiser. En effet, à chaque nouveau pas, un nœud à la dynamique discrète est créé pour optimiser la durée du nouveau pas et la position des nouveaux contacts. Nous avons donc décidé de faire autant de réseaux de neurones qu'il y a de matrice de contact différente (28). g_0 et n_{first} ne sont donc pas dans le vecteur d'entrées des réseaux de neurones.

Nous avons ensuite choisi un vecteur d'entrées qui ne contient pas d'informations redondantes :

$$\text{in} = \left[n_{total} \quad z_0 \quad \theta_0 \quad \phi_0 \quad \mathbf{v}_0 \quad \omega_0 \quad \mathbf{r}_{1,0} \quad \cdots \quad \mathbf{r}_{4,0} \quad v_x^* \quad v_y^* \quad \omega_z^* \right]^T \quad (4.18)$$

où n_{total} est le nombre total de nœuds dans la phase courante (nœuds écoulés et nœuds restants), z_0 , θ_0 et ϕ_0 sont la hauteur initiale et les angles de roulis et tangage initiaux de la base du robot, \mathbf{v}_0 et ω_0 ses vitesses de translation et rotation initiales. Finalement, pour tout $i \in [1, 4]$, $\mathbf{r}_{i,0}$ est la position initiale du pied i (ou position du dernier contact si le pied est en phase de vol). Il y a donc 21 composantes

En sortie nous avons les trajectoires des états \mathbf{X} et contrôles \mathbf{U} discrétisés. De la même manière que pour le vecteur d'entrée, les états contiennent des informations redondantes (les positions des contacts et la durée du nœud ne changent pas au cours d'un pas). Nous avons donc changé le format pour conserver les informations nécessaires pour reconstruire

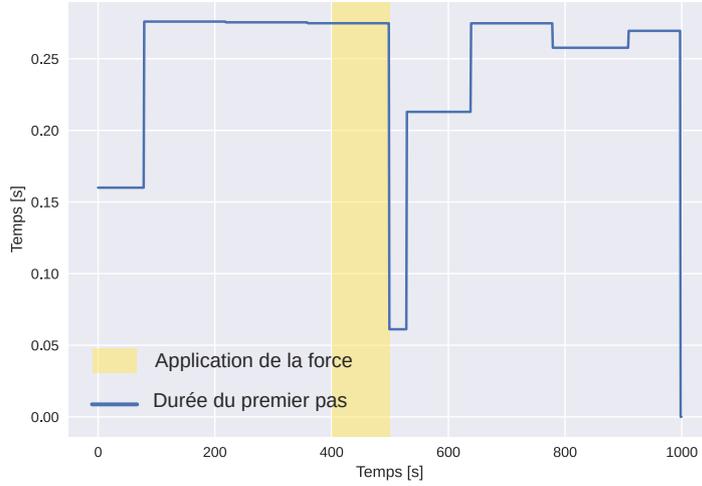


FIGURE 23 – Évolution de la durée des pas quand on applique une force de perturbation

une supposition initiale de l'état et du contrôle :

- * La trajectoire discrète de l'état inertiel - rigide $\mathbf{X}_{base} \in \mathbb{R}^{12*16}$
- * Les positions des pieds qui créent un contact $\mathbf{R} \in \mathbb{R}^{4*n_{pas}}$
- * Les durées des nœuds de chaque nouveau pas $\in \mathbb{R}^{n_{pas}}$
- * Les forces de contact $\mathbf{U}_{forces} \in \mathbb{R}^{6*16}$
- * Les vitesses moyennes des pieds de vol sur le pas précédent $\mathbf{V} \in \mathbb{R}^{4*n_{pas}}$

où n_{pas} est le nombre de nouveaux pas dans l'horizon d'optimisation.

Pour générer les données, nous avons d'abord fait tourner des nombreuses simulations avec IpOpt. Nous avons enregistré les entrées et sorties à chaque appel de l'OCP. Nous avons ensuite traité ces données en regardant la distribution de chaque composante du vecteur d'entrées, ce qui nous a permis de choisir un intervalle de variation pour chacune. Nous avons généré le nombre voulu de vecteurs d'entrée dont chaque composante est choisie aléatoirement en suivant une distribution uniforme sur son intervalle de variation. Finalement nous avons résolu l'OCP avec IpOpt, puis avec Crocoddyl en utilisant la solution de IpOpt pour le démarrage à chaud et enregistré la solution de Crocoddyl qui servira à entraîner le réseau de neurones. Cette dernière étape permet d'améliorer la précision de convergence (Crocoddyl et son DDP étant moins sensible à la propagation des erreurs numériques dans la ligne temporelle).

4.3.2 Réseau de neurone et apprentissage

Nous utilisons un perceptron multicouche linéaire pour l'apprentissage, avec la fonction d'activation "ReLU". Les données d'entrée sont normalisées. La fonction coût utilisée pour

l'apprentissage est l'erreur quadratique moyenne. Nous utilisons l'algorithme d'optimisation Adam et un taux d'apprentissage qui décroît au cours de l'optimisation.

4.3.2.1 Paramètres de l'apprentissage

De nombreux paramètres ont été réglés pour obtenir de bons résultats.

Le nombre d'échantillons dans la base de donnée a été étudié. Nous avons remarqué que si l'apprentissage permet d'améliorer la convergence à partir de 5000 échantillons environ, les meilleurs résultats sont obtenus pour des réseaux entraînés à partir de 20000 échantillons.

Le nombre d'époques utilisé lors de l'apprentissage et le taux d'apprentissage ont été aussi ajustés en parallèle de la taille du réseau de neurones. En effet, un réseau de neurones trop petit ne sera pas être assez spécifique, alors qu'un réseau trop grand sera très long à converger, avec un risque de sur-apprentissage qui nuirait à la généralisation.

Nous avons finalement choisi un réseau avec trois couches cachées de 64, 128 et 256 neurones.

4.3.2.2 Filtrage des données d'apprentissage

Un filtrage des données d'apprentissage est effectué au cours de l'apprentissage. Il consiste à enlever les échantillons qui ont la plus grande erreur quadratique moyenne. Nous enlevons 0,1% des échantillons toutes les 5000 époques.

4.3.3 Résultats de l'apprentissage

4.3.3.1 Définition d'un critère de succès

Afin d'évaluer l'impact d'un type de démarrage à chaud sur la convergence de Crocodyl, nous avons mis au point plusieurs métriques. La première nous permet d'estimer si un OCP a été bien résolu. Pour cela nous avons détecté trois types de problèmes potentiels, qui caractérisent un échec :

- Le premier est un problème de convergence : le solveur numérique a atteint le nombre maximal d'itérations sans converger.
- Le second est la sélection d'une solution à coût négatif. Cela correspond à une solution dont le pas d'intégration est négatif sur un pas au moins. Dans ce cas le coût barrière qui borne ce temps est violé, mais le solveur tombe dans un minimum où de grandes erreurs sur la régularisation de l'état et du contrôle associées à un pas d'intégration négatif font diminuer considérablement le coût global.
- Le troisième est la sélection d'une solution dont le coût est trop grand par rapport à celui de la vérité terrain. Cela correspond à une convergence vers un minimum local qui sera souvent trop mauvais pour nos expériences de locomotion. Nous choisissons une distance de 1 obtenue empiriquement lors de l'analyse des données.

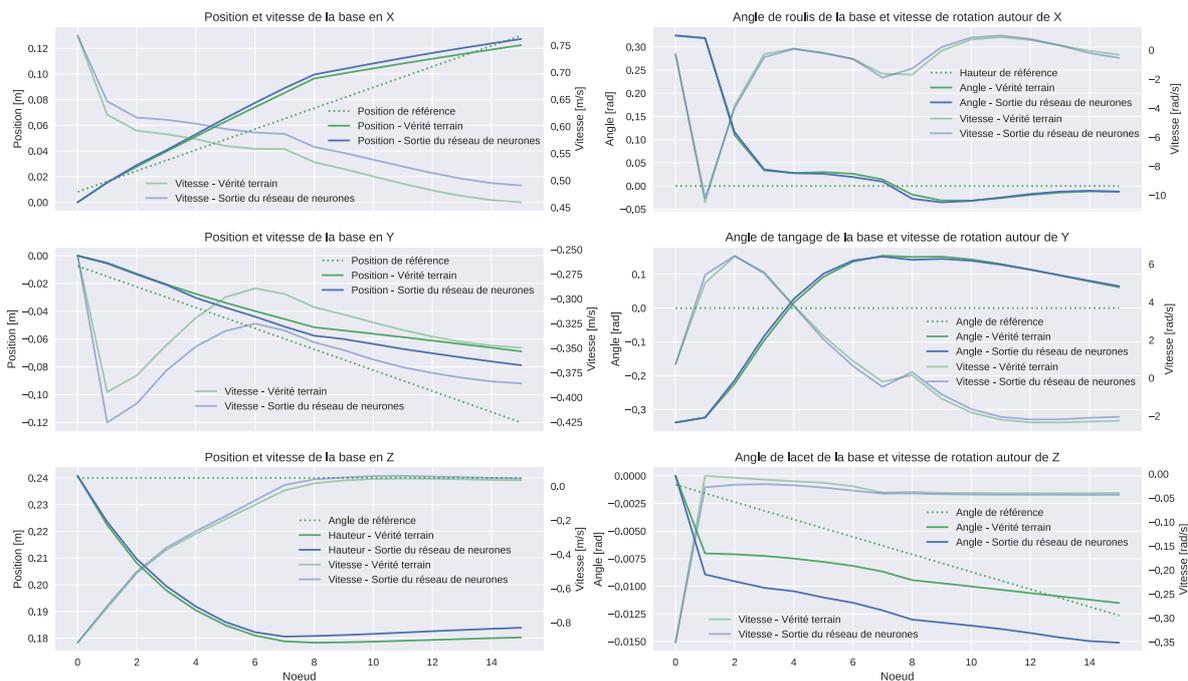


FIGURE 24 – Comparaison de l'état de la base entre la vérité terrain et l'estimation du réseau de neurones

Nous pouvons donc définir ce qu'est un succès par opposition. Cependant, même en cas de succès, les solutions peuvent être de plus ou moins bonne qualité. Nous considérons que la vérité terrain donnée par une supposition initiale provenant de la solution de IpOpt est la solution parfaite. Nous pouvons alors mesurer l'erreur relative sur le coût par rapport à cette solution. Si Crocodyl converge vers cette solution, les coûts seront les mêmes et l'erreur nulle. EN revanche il arrive fréquemment que Crocodyl converge vers un autre minimum local légèrement moins bon. L'erreur relative augmentera alors.

Ces métriques nous ont permis d'évaluer les performances de différents réseaux de neurones pour le démarrage à chaud de Crocodyl. Nous avons commencé par les évaluer sur la base de données d'apprentissage, avant de s'intéresser à leur capacité de généralisation.

4.3.3.2 Performances sur les données d'entraînement

Nous avons commencé par évaluer la qualité de l'apprentissage en comparant la vérité terrain avec la sortie du réseau. Nous avons regardé les valeurs des erreurs quadratiques moyennes pour l'état et le contrôle et observé visuellement les résultats pour des échantillons pris au hasard dans la base de données. La figure 24 montre la vérité terrain d'un échantillon (Position et orientation de la base) et l'estimation donnée par le réseau de neurones.

Pour s'assurer de l'intérêt d'un bon démarrage à chaud, nous avons ensuite comparé les performances de Crocodyl pour la résolution de notre OCP avec différents démarrages à chaud. La première supposition initiale, appelée vérité terrain, vient de la base de données

d'entraînement. C'est la solution de Crocoddyl obtenue avec un démarrage à chaud donné par IpOpt. La seconde solution initiale est une solution "par défaut" calculée en prenant les valeurs nominales pour chaque variable (qui sont les valeurs utilisées pour la régularisation). Enfin, la troisième solution initiale est la solution obtenue avec le réseau de neurones. Les résultats d'un réseau de neurone (les résultats sont du même ordre de grandeur pour tous les réseaux, nous avons donc choisi arbitrairement le réseau correspondant à la matrice de contact (0, 7)) sont présentés dans le tableau 4.2. On voit bien l'intérêt d'un bon démarrage à chaud pour faciliter la convergence vers une solution acceptable. Avec le réseau de neurone, Crocoddyl converge dans 93% des cas, alors qu'en utilisant une supposition initiale par défaut, ce n'est que dans 52%. On voit que l'erreur relative moyenne sur les bons échantillons est aussi plus faible pour le démarrage à chaud utilisant le réseau de neurones.

TABLE 4.2 – Convergence de Crocoddyl sur la base de donnée avec différentes suppositions initiales

Démarrage à chaud	Vérité terrain	Valeurs par défaut	Réseau de neurone
Bonne convergence	1000.0%	52.18%	93.20%
Problème de convergence	0.0%	0.09%	0.01%
Coût négatif	0.0%	31.52%	5.81%
Coût trop grand	0.0%	16.22%	0.98%
Nombre moyen d'itérations	1.02	16.55	6.49
Erreur relative sur le coût	0.	92%	0.1%

4.3.3.3 Performances sur de nouvelles données

La capacité de généralisation des réseaux a ensuite été évaluée en observant la convergence et l'erreur relative moyenne sur des échantillons générés aléatoirement (donc pas présents dans la base d'entraînement). La Tableau 4.3 montre que la convergence est toujours améliorée par l'utilisation du réseau de neurones. L'erreur relative augmente légèrement par rapport aux résultats sur la base d'entraînement, mais reste bien plus faible que dans le cas de la supposition initiale par défaut.

TABLE 4.3 – Convergence de Crocoddyl sur de nouvelles données avec différentes suppositions initiales

Démarrage à chaud	Vérité terrain	Valeurs par défaut	Réseau de neurone
Bonne convergence	100.0%	45.61%	78.66%
Problème de convergence	0.0%	0.21%	0.00%
Coût négatif	0.0%	41.80%	17.1%
Coût trop grand	0.0%	12.38%	4.18%
Nombre moyen d'itérations	1.00	16.55	6.49
Erreur relative sur le coût	0.0%	18%	4%

Si nous voyons bien l'intérêt d'utiliser le réseau de neurones, les performances restent toutefois insuffisantes, avec seulement 80% des échantillons qui convergent bien. Nous observons une légère différence entre les résultats avec les valeurs par défaut sur la base d'entraînement

et sur des nouvelles données. Nous expliquons ces différences par le filtrage des données d’entraînement effectué lors de l’apprentissage.

4.4 Simulation avec Crocodyl

Avec ces résultats encourageants sur l’apprentissage et la convergence, nous avons essayé de reproduire les expériences en simulation présentées Section 4.2 en utilisant Crocodyl pour résoudre l’OCP. Nous avons choisi d’utiliser la solution décalée du dernier OCP résolu pour le démarrage à chaud au cours d’un pas, quand seul le dernier nœud a besoin d’une supposition initiale par défaut. En revanche, les réseaux de neurones sont utilisés au début de chaque pas, où la matrice de contact change et décale les nœuds d’optimisation.

Sans perturbations, les simulations s’arrêtent toutes au bout de quelques pas à cause d’une mauvaise convergence de Crocodyl. Pour pouvoir étudier plus en détail le comportement, nous avons choisi de résoudre l’OCP avec IpOpt quand un problème était détecté. Cela nous a permis d’observer le comportement lors de locomotion avec différentes vitesses de référence sur des simulations plus longues.

Le Tableau 4.4 présente le pourcentage de problèmes pour les différents types de suppositions initiales : avec un réseau de neurone au début du pas ou avec la solution précédente décalée au cours du pas. Ces résultats ont été obtenus lors de simulations avec des vitesses de références choisies aléatoirement et variant au cours de la simulation, mais sans perturbations extérieures.

TABLE 4.4 – Problèmes de convergence de Crocodyl lors de simulations

Démarrage à chaud	Réseau de neurone	Décalage de la solution précédente
Pourcentage d’échec	6.02%	2.06%
Nombre total d’appels	847	5534

Ce tableau nous permet de voir que le taux d’échec pour le réseau de neurones est plus faible que ceux estimés dans la Section 4.3.3. Cela s’explique par le fait que la variabilité des entrées du réseau est moins importante lors d’une tâche de locomotion que quand elles sont générées aléatoirement.

Ces résultats sont donc cohérents avec ceux trouvés dans la partie précédente. L’apprentissage améliore la convergence de Crocodyl mais pas suffisamment pour permettre une utilisation sur le robot.

Conclusion

Nous proposons dans ce chapitre une formulation d’OCP permettant l’optimisation simultanée du modèle inertiel rigide, des emplacements des pas et de leur durée. La formulation est d’abord présentée, avant d’être évaluée lors d’expériences de marche sur quadrupède en simulation. L’intérêt de l’optimisation de la durée des pas est montré en cas de perturbations.

Pour pouvoir faire des expériences en temps réel sur le robot Solo, nous implémentons l'OCP dans le solveur Crocoddyl, et apprenons avec un réseau de neurones une bonne supposition initiale pour le démarrage à chaud du solveur. Une dernière partie évalue les performances du contrôleur utilisant Crocoddyl en simulation. Les performances ne sont toutefois pas assez bonnes pour permettre d'effectuer des expériences sur le robot.

Ce travail nous a permis de voir les limites du DDP. Sans possibilité d'ajouter des contraintes et avec une grande dépendance vis-à-vis de la supposition initiale, le solveur Crocoddyl n'est pas approprié pour l'optimisation du temps dans un OCP inertiel-rigide non linéaire. Nous avons cependant pu observer en simulation grâce à un solveur doté d'une bonne capacité de globalisation (IpOpt) l'intérêt de l'optimisation du temps en cas de perturbation. L'utilisation d'un réseau de neurones pour le démarrage à chaud permet d'améliorer la convergence d'un algorithme plus rapide, mais moins robuste (Crocoddyl), mais pas suffisamment pour reproduire les résultats obtenus avec le premier solveur. L'utilisation d'un autre solveur numérique serait souhaitable pour pouvoir effectuer des expériences sur le robot.

L'étude montre donc que, si l'apprentissage peut améliorer le comportement d'un solveur, il ne suffit pas à se substituer complètement aux propriétés de convergence quand le problème est difficile : il est donc nécessaire d'améliorer aussi l'algorithme de résolution numérique. Enfin, nous n'avons pas pu évaluer les propriétés géométriques des variables de temps qui nous avons introduites et de leur impact sur le paysage de la fonction de coût. Même si un solveur globalisant comme IpOpt arrive à s'en défaire, il est possible que nous puissions réduire la difficulté pour un solveur comme Crocoddyl par une reformulation plus astucieuse, qui induirait moins de non-linéarité ou de non-convexité.

Contrôle avec un MPC corps-complet

Contents

5.1	Calcul des consignes pour le contrôleur en impédance	73
5.1.1	Retour linéaire corps-complet pour le calcul des couples de référence	74
5.1.2	Consignes d'état pour le contrôle d'impédance	74
5.1.3	Intégration	74
5.1.4	Interpolation	74
5.2	Formulation du problème de contrôle optimal	74
5.2.1	Coût pour la trajectoire du pied de vol	75
5.2.2	Coûts d'impact	75
5.2.3	Régularisation	75
5.2.4	Contraintes	76
5.2.5	Suivi de vitesse de référence	76
5.3	Résultats	76
5.3.1	Comparaison des consignes d'état articulaires	77
5.3.2	Marche	78

Introduction

Les systèmes dynamiques de grande dimension forment des **OCP** dont le nombre de variables entraîne des difficultés pour atteindre les objectifs de temps de calcul, ce qui explique la large utilisation des modèles réduits. Nous avons proposé dans les deux premières parties de cette thèse des contributions sur des **MPC** exploitant des modèles réduits de la dynamique, leur permettant de produire des marches plus stables et adaptées à des terrains difficiles. Nous avons cependant remarqué que l'ajout de certains paramètres de la marche dans la formulation ne pouvait pas se faire directement avec le modèle inertiel rigide. Ce chapitre nous permettra de nous affranchir des limites de ces modèles réduits et d'explorer les potentialités offertes par l'utilisation de la dynamique complète dans le problème d'optimisation.

Si de tels contrôleurs ont déjà permis d'obtenir un contrôle stable de la locomotion, deux limites des contrôleurs de la littérature ont été identifiées : l'utilisation systématique de pré-calculs pour guider le résultat et le manque de recherche sur leur utilisation avec des robots dépourvus de capteurs de couple.

Plusieurs sortes de pré-calculs sont utilisés. Meduri [Med22] suivant Budhiraja [Bud19] guide par exemple le solveur corps-complet avec une trajectoire centroïdale, optimisée en alternance avec la trajectoire cinématique. Mais les pré-calculs les plus généralement réalisés concernent les trajectoires des pieds de vol. Elles peuvent être de simples polynômes [Léz21], ou nécessiter des calculs plus avancés pour prendre en compte les obstacles sur le chemin, comme nous avons vu Section 3. Dans tous les cas, elles sont généralement données en entrée de l’OCP sous la forme d’un coût de suivi de la position du pied [Dan21]. Si quelques approches [Win18 ; Mel20] donnant au contrôleur la liberté de choisir les trajectoires des pieds ont été étudiées, elles ont conduit à des calculs plus lents. Dans ce chapitre, nous proposons un coût efficace pour l’OCP, qui permet au solveur de trouver par lui-même les trajectoires des pieds de vol et de nous affranchir de ces pré calculs courants dans la littérature.

Deux familles principales de robots existent aujourd’hui : les robots contrôlés en position et les robots contrôlés en couple. La plupart des robots sont dits contrôlés en position, comme le robot quadrupède Solo (présenté Section 2.3) et sont dépourvus de capteurs de couple aux articulations. Ils possèdent en revanche des encodeurs aux articulations, qui leur permettent de mesurer les positions et les vitesses articulaires. Sans retour sur le couple et à cause des imprécisions de modélisation des actionneurs, il est difficile de suivre directement une consigne de couple sur ces robots. Le contrôle de ces robots se fait donc généralement par impédance, c’est-à-dire en envoyant des consignes de position, vitesse et couple articulaires qui sont suivies grâce à un contrôleur de type PD+ (*Proportionnel plus feedforward*) tel que celui présenté Section 2.3.2. Les robots contrôlés en couple sont quant à eux équipés de capteurs de couple au niveau des articulations. Ils sont donc contrôlés directement par des consignes de couple articulaires, dont le suivi est assuré à haute fréquence par le contrôleur bas-niveau grâce aux capteurs de couple.

Nous nous proposons donc d’étudier comment l’utilisation de la dynamique complète peut permettre de définir des OCPs effectifs sur des robots contrôlés en impédance. Pour les MPCs corps-complet développés pour les robots à pattes [Dan21 ; Mas20 ; Kat22], l’OCP calcule le couple articulaire et les trajectoires d’état sur un horizon, mais ne peut toujours pas être résolu à une fréquence suffisamment élevée pour correspondre au contrôleur de bas niveau des robots [Wen17]. Pour pallier cette limitation, les gains de Riccati [Dan22a] peuvent être utilisés pour ajuster les couples envoyés aux servo-contrôleurs du robot. Pour les robots avec un retour de couple, c’est suffisant. Pour les robots contrôlés en position, nous proposons deux méthodes pour calculer les consignes d’état des articulations, ainsi que les gains de rétroaction envoyés au contrôleur de bas-niveau.

Ce chapitre propose donc une solution pour l’optimisation de la marche par un MPC corps-complet sans pré-calculs et adaptée aux robots contrôlés par impédance. Les deux méthodes pour calculer les référence d’état articulaires nécessaires au contrôle en impédance sont présentées dans la Section 5.1. La première est une interpolation par spline de l’état discrétisé calculé par l’OCP. La seconde est l’intégration des couples calculés avec les gains de Riccati. La formulation de l’OCP est présentée dans la Section 5.2 avec un accent mis sur la conception du coût pour la trajectoire du pied de vol qui permet de s’affranchir des pré-calculs. Enfin, l’évaluation expérimentale et les résultats sont présentés dans la Section 5.3,

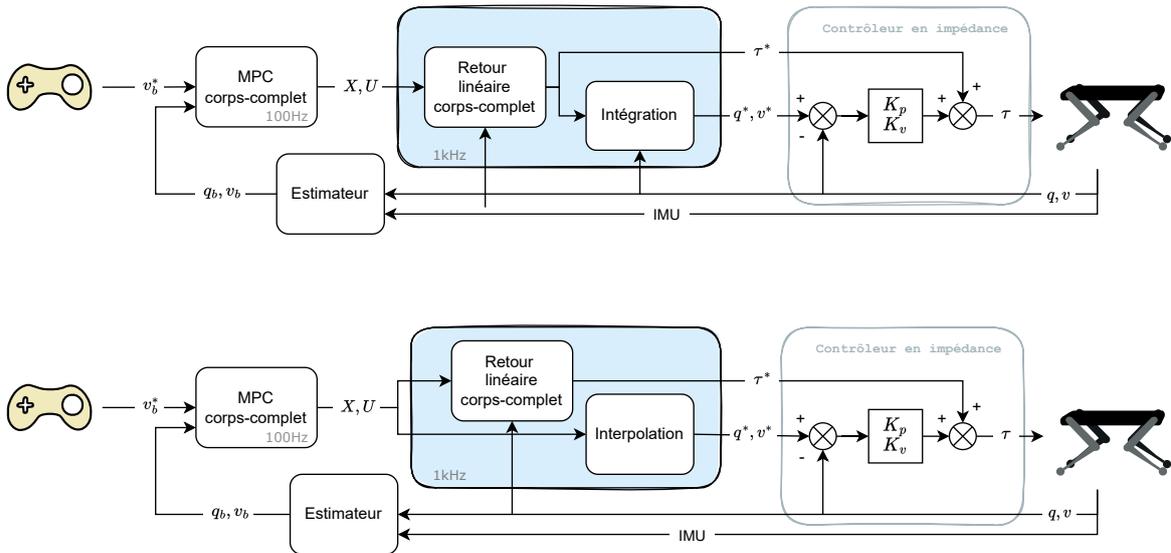


FIGURE 25 – Architecture du contrôleur de marche de Solo, utilisant l’intégration - Section 5.1.3 (haut)- ou l’interpolation -Section 5.1.4 (bas).

notamment une comparaison des deux méthodes de calcul des références articulaires.

Collaboration

Ce travail a été réalisé en collaboration avec Alessandro Assirelli, étudiant au laboratoire d’ingénierie industrielle de l’Université de Trente, dont j’ai participé à l’encadrement pendant son stage au sein de l’équipe Gepetto.

5.1 Calcul des consignes pour le contrôleur en impédance

La plupart des OCP pour la locomotion sont écrits, comme le nôtre, en considérant que les couples articulaires sont l’entrée de commande. Le MPC corps-complet est donc généralement utilisé en supposant que le robot est capable de suivre correctement les couples de référence $\mathbf{u}_0 = \pi(\mathbf{x})$. Or, seuls quelques robots sont équipés de capteurs de couple et d’un retour de couple au niveau du contrôleur bas-niveau. Pour les autres robots, comme le robot Solo, les actionneurs sont pilotés par impédance (souvent appelé "contrôle en position"), suivant un état de référence donné (Section 2.3.2). Dans cette section, nous proposons une méthode permettant de relier le MPC corps-complet à la commande par impédance. Nous commençons par décrire les différentes méthodes de calcul des références τ^* , \mathbf{q}^* et $\dot{\mathbf{q}}^*$, qui sont résumés figure 25.

Ces méthodes de rééchantillonnage sont ensuite évaluées et comparées Section 5.3.

5.1.1 Retour linéaire corps-complet pour le calcul des couples de référence

L'OCP fournit les trajectoires discrètes du couple de référence des articulations. Le pas de discrétisation est de 12 ms dans notre cas. Pour calculer les couples de référence à envoyer à 1 kHz, un retour linéaire corps-complet est réalisée en utilisant les gains de Riccati \mathbf{K}_i :

$$\tau^* = \tau_0 + \mathbf{K}_0(\mathbf{x} - \mathbf{x}_0) \quad (5.1)$$

où $\tau_0 = \tau_0(\mathbf{x}_0)$ est le couple de référence initial calculé par l'OCP, \mathbf{x}_0 l'état initial donné à l'OCP toutes les 12 ms, \mathbf{x} est l'état mesuré estimé à 1 kHz.

5.1.2 Consignes d'état pour le contrôle d'impédance

Il existe plusieurs façons de calculer les états de référence envoyés au contrôleur bas-niveau dans (2.36). Deux possibilités sont présentées dans les sections suivantes.

5.1.3 Intégration

La première est l'intégration des couples de référence. L'accélération \mathbf{a} est obtenue en calculant la dynamique directe (implémentée dans la bibliothèque Pinocchio [Car19b] en utilisant l'Articulated Body Algorithm, ABA [Fea83]), à partir des mesures de l'état courant et des couples de référence τ^* de (5.1).

$$\begin{aligned} \mathbf{a} &= ABA(\mathbf{q}, \dot{\mathbf{q}}, \tau^*, \mathbf{f}) \\ \dot{\mathbf{q}}^* &= \dot{\mathbf{q}} + \mathbf{a} \cdot dt \\ \mathbf{q}^* &= \mathbf{q} + \dot{\mathbf{q}}^* \cdot dt \end{aligned} \quad (5.2)$$

où les forces \mathbf{f} sont calculées par le solveur de contrôle optimal suivant (2.12) (et sont nulles lorsqu'il n'y a pas de contacts).

5.1.4 Interpolation

La deuxième proposition consiste à choisir \mathbf{x} comme une interpolation polynomiale de la trajectoire d'état discrétisée \mathbf{X} calculée par l'OCP. L'interpolateur de Krogh [Kro70] calcule une interpolation des points de la trajectoire associés à leurs dérivées. Les trois premiers échantillons $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2$ de la trajectoire d'état sont utilisés pour contraindre l'interpolation.

5.2 Formulation du problème de contrôle optimal

Cette section présente la formulation de l'OCP utilisé dans notre contrôleur de locomotion. Il est implémenté avec Crocoddyl [Mas20]. Les équations de la dynamique ont été présentées dans la section 2. Les instants et durées des contacts sont prédéfinis et transmis pour définir les contraintes de contact utilisées pour chaque nœud de l'OCP.

Pour simplifier la notation des poids de coûts, ils apparaîtront comme suit :

$$\|\xi\|_w^2 = \sum_{i=1}^n w_i \xi_i^2, \quad \mathbf{w} = (w_1, \dots, w_n) \quad (5.3)$$

où les poids w seront souvent réduits à un scalaire.

Le coût de fonctionnement $\ell_t(\mathbf{x}, \mathbf{u})$ est défini comme la somme des coûts suivants.

5.2.1 Coût pour la trajectoire du pied de vol

Dans un problème typique de locomotion, un quadrupède suit des trajectoires prédéfinies des pieds de vol. Dans ce travail, nous remplaçons la trajectoire de référence des pieds par le coût suivant :

$$\ell_{vol}^k(\mathbf{x}) = \frac{1}{e^{\gamma h_k}} \|\mathbf{v}_k^{xy}\|_{w_{vol}}^2 \quad (5.4)$$

où k est un pied qui n'est pas en contact ; $\mathbf{v}_k^{xy} = (v_k^x, v_k^y) \in \mathbb{R}^2$ contient les composantes longitudinale et latérale de la vitesse linéaire du pied ; h_k est la hauteur du pied ; $w_{fly} = 5 \cdot 10^4$, $\gamma = 50$ sont les hyperparamètres. Le solveur cherchera à augmenter l'altitude lorsqu'une vitesse élevée \mathbf{v}^k est nécessaire et restera près du sol en cas de petite vitesse. Nous montrerons dans les expériences que ce coût permet d'obtenir des trajectoires de pieds en forme de cloche sans avoir besoin (comme dans les précédents OCP corps-complet) d'imposer une trajectoire de référence ou même des emplacements de contact.

5.2.2 Coûts d'impact

Deux coûts sont utilisés pour gérer correctement l'impact d'un pied sur le sol. Ils ne sont activés que lorsqu'un pied k passe d'une phase sans contact à une phase de contact. Le premier impose que la hauteur du pied au contact h_k soit la hauteur h_0 du pied à l'initialisation. La seconde impose que la vitesse linéaire du pied $\mathbf{v}_k \in \mathbb{R}^3$ soit égale à 0 lorsqu'il crée le contact.

$$\begin{aligned} \ell_{h,k}(\mathbf{x}) &= \|h_k - h_0\|_{w_h}, & w_h &= 10^4 \\ \ell_{i,k}(\mathbf{x}) &= \|\mathbf{v}_k\|_{w_i}, & w_i &= 10^4 \end{aligned} \quad (5.5)$$

5.2.3 Régularisation

Trois coûts sont utilisés pour régulariser (donc limiter) les variables d'état, de contrôle et les forces de contact :

$$\begin{aligned} \ell_x(\mathbf{x}) &= \|\mathbf{x} - \mathbf{x}^*\|_{w_x}^2, & w_x &= [\mathbf{0}_7 \ 300 \cdot \mathbf{1}_{12} \ \mathbf{0}_6 \ 20 \cdot \mathbf{1}_{12}]^T \\ \ell_u(\mathbf{u}) &= \|\mathbf{u}\|_{w_u}^2, & w_u &= 10^4 \\ \ell_f(\mathbf{x}, \mathbf{u}) &= \|\mathbf{f} - \mathbf{f}^*\|_{w_f}^2, & w_f &= 10^2 \end{aligned} \quad (5.6)$$

où $\mathbf{x}^* = [\mathbf{q}_0 \mathbf{0}_{n_v}]^T$ avec \mathbf{q}_0 la configuration initiale du robot. Et pour les pieds en contact, la force de référence est une fraction du poids du robot :

$$\mathbf{f}^* = \left[0, 0, \frac{m g}{n_c} \right]^T \quad (5.7)$$

avec n_c le nombre de pieds en contact.

La vitesse de l'articulation est également régularisée par un coût terminal :

$$\ell_T(\mathbf{x}) = \|(\mathbf{x}_T)\|_{w_{term}}^2, \quad w_{term} = [\mathbf{0}_{n_q} \ 10 \cdot \mathbf{1}_{n_v}^3]^T \quad (5.8)$$

avec n_q la dimension de la position de l'état.

5.2.4 Contraintes

On utilise l'algorithme de résolution FDDP présenté Section 2.2.5. Cet algorithme ne peut appliquer aucune autre contrainte que la dynamique (2.12). Les limites sur les vitesses et le contrôle des articulations sont implémentées avec les pénalités suivantes :

$$\begin{aligned} \ell_{bv}(\mathbf{x}) &= \|\min(\max(\mathbf{x}, \bar{\mathbf{x}}), \underline{\mathbf{x}})\|_{w_{b_x}}^2, & w_{b_x} &= [\mathbf{0}_{n_q+6} \ \mathbf{1}_{n_v-6}]^T \\ \ell_{bu}(\mathbf{u}) &= \|\min(\max(\mathbf{u}, \bar{\mathbf{u}}), \underline{\mathbf{u}})\|_{w_{b_u}}^2, & w_{b_u} &= 10^4 \end{aligned} \quad (5.9)$$

où $\bar{\mathbf{x}}$, $\underline{\mathbf{x}}$, $\bar{\mathbf{u}}$ et $\underline{\mathbf{u}}$ sont les limites de vitesse et de couple adaptées au robot.

Un coût barrière supplémentaire a été ajouté sur la hauteur de chaque pied de vol k pour empêcher l'OCP de sélectionner des solutions où un pied de vol pénétrerait dans le sol :

$$\ell_g(\mathbf{x}) = \|\max(h_k, h_0)\|_{w_g}^2, \quad w_g = 10^3 \quad (5.10)$$

5.2.5 Suivi de vitesse de référence

Enfin, la vitesse du robot est suivie à l'aide d'un coût dédié :

$$\ell_t(\mathbf{x}) = \|\mathbf{v}_{base} - \mathbf{v}_{base}^*\|_{w_t}^2, \quad w_t = 8 \cdot 10^5 \quad (5.11)$$

dans laquelle $\mathbf{v}_{base} \in \mathbb{R}^6$ est la vitesse spatiale de la base du robot et $\mathbf{v}_{base}^* \in \mathbb{R}^6$ une vitesse de référence obtenue à partir d'une manette de jeu par exemple (voir Fig. 25).

5.3 Résultats

Des expériences ont été réalisées pour comparer certains choix de conception et évaluer les performances du contrôleur. Elles ont été réalisées avec le robot quadripède Solo.

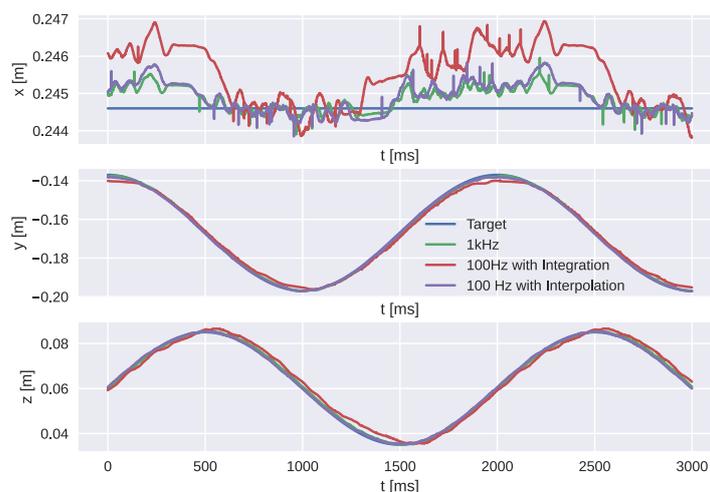


FIGURE 26 – Expérience 5.3.1.1. Comparaison du suivi d’une trajectoire des pieds dans une expérience à 3 DoFs

Le schéma de contrôle est adapté du contrôleur présenté Section 2.3.2. Le planificateur, MPC et contrôleur corps-complet ont été remplacés par le MPC corps-complet et le calcul des consignes d’impédance présentés dans ce chapitre.

5.3.1 Comparaison des consignes d’état articulaires

5.3.1.1 Expériences sur modèle réduit

Pour étudier l’impact du rééchantillonnage des trajectoires d’état sur le comportement du robot (voir section 5.1), nous avons conçu un MPC pouvant fonctionner à 1 kHz, en utilisant un modèle réduit du robot, avec seulement trois degrés de liberté (DoF). L’objectif du MPC est de suivre une trajectoire de position du pied (courbe sinusoïdale dans le plan frontal).

Cela nous a permis de comparer les performances de suivi du MPC évalué à 1 kHz en utilisant directement les consignes calculées par l’OCP, avec le même MPC évalué à 100 Hz en utilisant des gains de Riccati et un rééchantillonnage. Les couples et les états des articulations ont été calculés à l’aide de gains de Riccati et d’interpolation ou d’intégration, comme décrit Section 5.1.

La figure 26 montre les trajectoires de la position du pied de ces trois contrôleurs lors d’une expérience sur le robot réel. Nous pouvons constater qu’elles sont assez similaires. L’impact de l’exécution du MPC à une fréquence plus basse et de l’utilisation des gains Riccati et du rééchantillonnage n’a pas dégradé le comportement du robot. L’intégration montre un suivi légèrement moins bon que l’interpolation.

5.3.1.2 Expériences de locomotion en simulation

Pour une étude plus poussée des différences entre l’intégration et l’interpolation, nous avons réalisé des expériences de locomotion. Sur le robot, l’intégration (5.1.3) ne fonctionnait

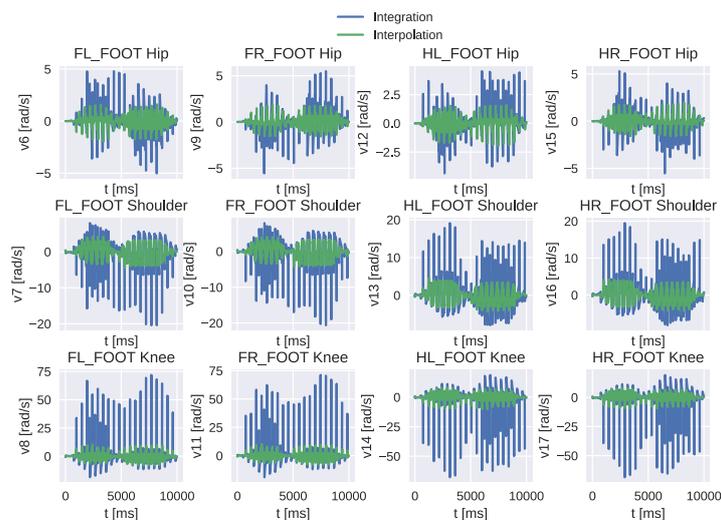


FIGURE 27 – Expériences 5.3.1.2 de locomotion en simulation sans bruit. Références de vitesse des articulations

pas, nous avons donc réalisé les expériences en simulation. Nous avons comparé les schémas d'interpolation (5.1.4) et d'intégration (5.1.3) pour calculer les états articulaires cibles entre deux nœuds de l'OCP.

La figure 27 montre les consignes de vitesse articulaire calculées par chaque moyen, dans une simulation sans bruit. Nous pouvons voir que les vitesses interpolées sont plus lisses que les vitesses intégrées, qui présentent des pics plus importants. Le suivi de la vitesse de la base du robot semble légèrement meilleur pour l'interpolation, comme on peut le voir sur la figure 28.

De plus, lorsque nous ajoutons du bruit dans la simulation (dans les couples appliqués aux articulations et les données mesurées), nous observons que la cible de vitesse intégrée devient plus bruyante, comme le montre la figure 29.

Ces expériences nous conduisent à sélectionner le schéma avec interpolation pour les expériences de locomotion sur le robot.

5.3.2 Marche

Les expériences de marche réalisées sur le robot ont démontré la robustesse du contrôleur. Le robot a été capable de suivre des références de vitesse linéaire jusqu'à 0,5 m/s, ainsi que des références angulaires jusqu'à 1 rad/s. Des tests de récupération après perturbation (on pousse le robot) ont été expérimentés avec succès, le contrôleur étant capable de se remettre d'un impact (voir Fig. 30) et de s'adapter à une force constante le poussant. Une vidéo d'accompagnement montre quelques séquences de ces expériences [Wbc].

La figure 31 montre les performances de suivi de la vitesse de base. La trajectoire en cloche des pieds de vol est visible sur la figure 32. Nous pouvons observer que lorsque la

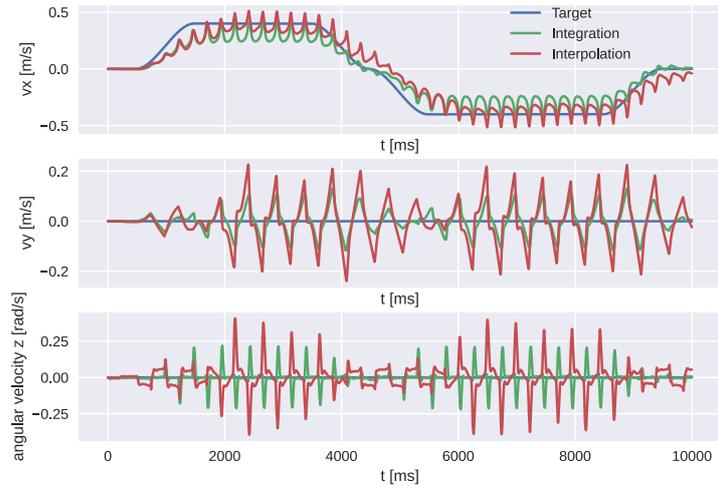


FIGURE 28 – Expériences 5.3.1.2 de locomotion en simulation sans bruit. Comparaison du suivi de la vitesse de base entre l’interpolation et l’intégration

vitesse linéaire de la base augmente, le pied de vol monte plus haut.

Conclusion

Dans ce chapitre, nous avons présenté notre travail sur un nouveau contrôleur de locomotion pour les robots à pattes, fondé sur un MPC corps-complet. Ce travail a conduit à la rédaction d’un article, en cours d’examen [Ass23].

Les références polynomiales traditionnelles des trajectoires des pieds de vol ont été remplacées dans le MPC par un coût innovant, laissant ainsi une liberté totale au solveur pour choisir le meilleur mouvement du pied. Nous avons également proposé une solution générique pour implémenter le MPC sur un robot sans contrôle de couple bas-niveau, en calculant les références d’impédance à l’aide d’une interpolation polynomiale. Des expériences en simulation et sur le quadrupède Solo ont été réalisées pour justifier certains choix de conception : avec un modèle réduit à trois degrés de liberté, l’interpolation des états articulaires ne génère aucune perte de performances par rapport à un OCP fonctionnant à des fréquences plus élevées. Enfin, la robustesse du contrôleur a été évaluée lors d’expériences de marche avec perturbations. Le coût pour le pied de vol a permis d’obtenir un mouvement réactif du pied, pour lequel une vitesse de la base plus élevée implique une plus grande hauteur du pied de vol.

Ce type de contrôleur pourrait être couplé avec les travaux présentés Section 3 pour permettre à des robots équipés de vision de raisonner sur les pas en connaissant l’environnement. L’adaptation de la position du pas et de la trajectoire de vol peut être effectuée par le MPC centroïdal et un module de planification réactive permettrait de choisir les surfaces de contact pour les prochains pas. L’optimisation des durées des pas pourrait aussi être étudiée dans un MPC corps-complet. Cela pourrait permettre de donner plus de liberté à l’optimisation pour choisir une allure de marche adaptée à la consigne de référence et aux perturbations

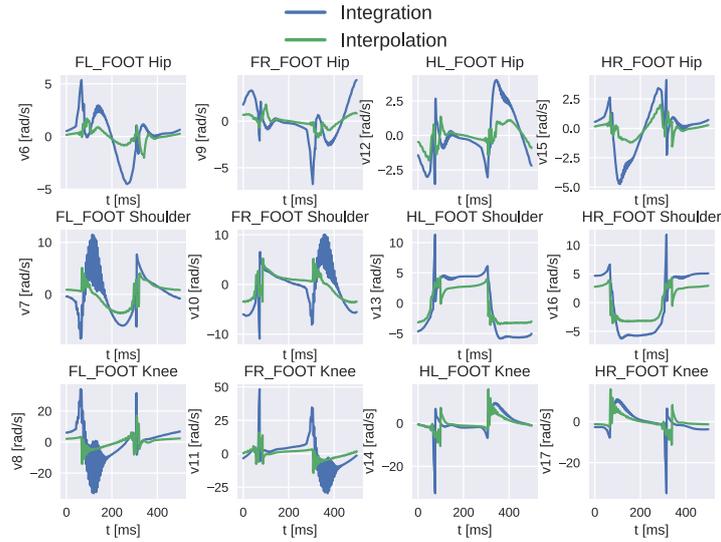


FIGURE 29 – Expérience 5.3.1.2. Références de vitesse d’articulation dans une simulation avec bruit



FIGURE 30 – Expérience 5.3.2. Rattrapage de chute avec MPC corps-complet. Vidéo [Wbc]

extérieures. Un avantage apporté par l’utilisation du modèle complet de la dynamique est la capacité de raisonner sur tout le corps pour trouver un mouvement qui satisfait les objectifs. Cela pourrait permettre d’effectuer d’autres types de mouvements, comme des sauts. Finalement une comparaison avec un contrôleur corps-complet avec une approche fondée sur les données pourrait être intéressante. Le contrôleur obtenu par apprentissage par renforcement proposé par Aractingi [Ara22] résout un problème similaire et pourrait donc être un candidat intéressant pour une telle comparaison.

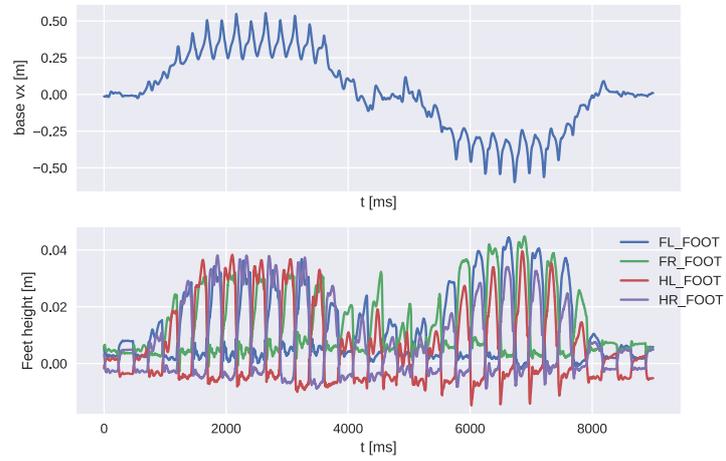


FIGURE 31 – Expériences 5.3.2 de locomotion sur le robot. Suivi de la vitesse de base

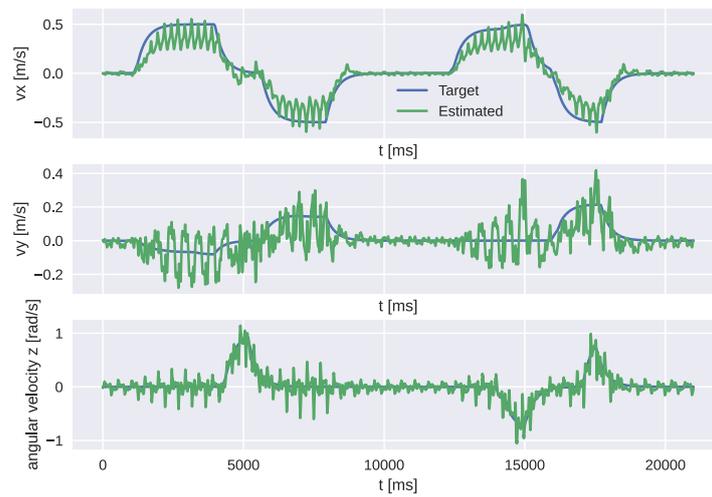


FIGURE 32 – Expériences 5.3.2 de locomotion. Vitesse de base et trajectoire des pieds de vol

Conclusion et perspectives

Contributions

Dans cette thèse, nous nous sommes proposé de contribuer à la locomotion des robots à pattes en questionnant le choix des modèles actuellement utilisés. En particulier, nous avons exploré plusieurs formulations de contrôle optimal pour la marche des quadrupèdes et leur utilisation dans des cadres expérimentaux variés. Cela nous a permis de mieux cerner les limites des modèles de contrôle optimal existants et de proposer des ajouts dont nous avons évalué expérimentalement les capacités.

Dans le Chapitre 3 nous avons étudié l'interaction entre un module de planification réactive et un schéma de contrôle par commande prédictive exploitant le modèle centroïdal, afin de permettre à un robot de raisonner sur son environnement lors du choix des contacts. Nous avons proposé une architecture de contrôle reposant sur un planificateur qui choisit les surfaces des prochains contacts. La position du prochain contact est ensuite adaptée localement dans la surface sélectionnée. Ces informations sur les contacts sont utilisées par le MPC centroïdal qui calcule les forces à appliquer. Finalement un contrôleur corps-complet génère les consignes articulaires qui suivent des trajectoires pour les pieds de vol évitant les collisions avec l'environnement. Ce contrôleur a été évalué expérimentalement sur le quadrupède Solo sur un ensemble de terrain 3D avec obstacles. Nous avons montré que chacun des modules qui le compose permet d'améliorer la robustesse globale. Ce travail a été présenté en conférence internationale [Ris22].

Nous nous sommes ensuite (Chapitre 4) intéressés au choix de la durée des pas dans un MPC centroïdal. Nous avons formulé un OCP dont l'état est augmenté pour contenir les positions des pas et leur durée. Une reformulation de l'OCP permet à chaque changement de pas de choisir la position des nouveaux contacts et la durée du prochain pas, qui est la somme des durées des nœuds d'optimisation qui le composent. Des coûts et contraintes ont été définis pour obtenir une formulation markovienne. Des simulations avec un solveur avec de bonnes capacités de globalisation (IpOpt) ont montré les performances du contrôleur lors de tâches de locomotion, même en présence de perturbations. Mais ce type de solveur n'est pas assez efficace pour une mise en œuvre sur le système réel. Nous avons ensuite proposé une mémoire de mouvements pour le démarrage à chaud d'un solveur DDP. Nous avons utilisé un réseau de neurones pour apprendre les valeurs initiales des variables à optimiser, ce qui améliore grandement la convergence. Les performances restent toutefois insuffisantes pour réaliser des expériences sur le robot.

Dans la troisième partie de la thèse (Chapitre 5) nous avons mis en place un MPC avec un modèle complet du robot. Nous avons proposé un coût innovant qui permet d'optimiser la position des pieds de vol sans devoir fournir de trajectoire précalculée. Nous avons aussi proposé une architecture permettant d'utiliser un tel MPC sur des robots contrôlés en impédance. Nos

expériences sur le robot quadrupède Solo ont montré les performances du contrôleur pour de la marche, même en cas de perturbations.

Perspectives pour le contrôle de la locomotion des robots marcheurs

Cette thèse propose plusieurs contributions aux limites des approches fondées sur les modèles. Une combinaison de ces différentes contributions serait intéressante à étudier à court terme :

- L'utilisation de planification réactive avec un MPC corps-complet. Thomas Corbères [Mas22] a réalisé ce travail en adaptant l'architecture que nous avons proposée pour Solo [Ris22] (qui est présentée chapitre 3) au robot ANYmal. L'utilisation des caméras du robot permet de s'affranchir de la dépendance à un système de MoCap pour la localisation dans l'environnement. Ce contrôleur permet à ANYmal de monter et descendre des escaliers avec planification réactive.
- L'optimisation des durées des phases dans un OCP avec un modèle complet de la dynamique serait intéressant pour s'affranchir de l'allure fixe. Un tel contrôleur pourrait prendre la décision de marcher, trotter ou sauter en fonction d'une consigne. D'autres types d'allures et de mouvements pourraient apparaître en fonction de l'environnement et de la consigne.

Le travail présenté chapitre 4 a mis en lumière les limites des solveurs numériques disponibles à l'heure actuelle sur des problèmes difficiles. La recherche de nouveaux algorithmes de résolution numérique adaptés à des problèmes non linéaires et non convexes semble donc essentielle. Les travaux de Wilson Jallet s'inscrivent dans cette optique [Jal22b].

Sur le long terme, les perspectives pour la locomotion des robots marcheurs sont nombreuses. Améliorer les performances de la marche sur tout type de terrain est nécessaire avant qu'ils soient utilisables et les approches fondées sur les données sont prometteuses en ce sens. Mais elles sont limitées par le manque de compréhension qui entoure leur résolution du problème et le manque de garanties qui en découle. La recherche sur l'utilisation conjointe des modèles et des données pourrait apporter des résultats intéressants.

Perspectives pour la robotique

Nous avons pu voir au cours de cette thèse que les robots marcheurs ont gagné en maturité. Cependant, la recherche sur les utilisations possibles et applications est limitée. Dans un contexte de réchauffement climatique et montée des tensions à l'échelle mondiale, il semble important de mener une vraie réflexion sur les applications des robots dans nos sociétés. Il me semble que cette recherche gagnerait à être faite par des collaborations pluri-disciplinaires. L'apport de roboticiens, écologues, sociologues ou médecins pourrait permettre de proposer des axes de recherche pour la robotique dictés par l'utilité, le fonctionnement actuel par projets et financements d'industriels favorisant les thématiques de recherches appliquées à des développements productivistes.

Contrôle par admittance appliqué à l'exosquelette Atalante

Contents

A.1	Contrôleur d'Atalante	87
A.1.1	Planificateur	87
A.1.2	Modèle réduit pour le contrôle de la marche	88
A.1.3	Cinématique inverse	88
A.1.4	Estimation d'état	88
A.2	Stabilisation par admittance	88
A.2.1	Contrôle du DCM	89
A.2.2	Admittance du CoM	89
A.2.3	Admittance au niveau des chevilles	89
A.3	Compensation de flexibilités	90
A.3.1	Calcul du décalage à appliquer	91
A.3.2	Application du décalage	91
A.3.3	Impact sur la trajectoire du pied de vol	91
A.4	Résultats expérimentaux	92

Introduction

Un exosquelette des membres inférieurs pourrait redonner de l'autonomie à de nombreuses personnes souffrant de troubles de la marche. Il pourrait leur permettre de se tenir debout, de marcher et de se déplacer dans un environnement conçu pour des personnes valides, avec des escaliers, des pentes et des cailloux. Les exosquelettes capables de faire cela sont encore rares, et ils nécessitent tous une action du patient ou du physiothérapeute pour assurer la stabilisation. Habituellement, des béquilles sont utilisées, mais elles sont très fatigantes à utiliser lors de longues séances, et privent le patient de l'usage de ses bras et de ses mains.

L'exosquelette Atalante de Wandercraft est un des rares exosquelettes à réaliser une marche dynamique sans béquilles [Mar22]. Cela se fait par la génération hors ligne de trajectoires de marche exécutées sur le robot [Gur18b]. Atalante permet à un patient entraîné de marcher sans l'aide d'un kinésithérapeute. Mais pour les patients non entraînés, ou si une perturbation apparaît, le robot n'a pas de mécanisme de stabilisation et une chute est susceptible de



FIGURE 33 – Exosquelette Atlante de Wandercraft sur la gauche. Essais cliniques avec assistance d’un physiothérapeute pour assurer la sécurité sur la droite.

se produire. Un portique est donc utilisé pour éviter les chutes et garantir la sécurité de l’utilisateur. Pour utiliser le robot dans un environnement non préparé, un portique mobile est utilisé, mais il interdit l’utilisation de l’exosquelette dans les endroits où le portique ne convient pas : espaces étroits, mais aussi endroits avec des escaliers ou un sol irrégulier.

Afin de pouvoir utiliser l’exosquelette sans portique tout en assurant la sécurité du patient, un mécanisme de contrôle de l’équilibre, souvent appelé stabilisateur, doit être utilisé. Nous présentons ici les résultats expérimentaux obtenus en ajoutant un stabilisateur sur l’exosquelette Atalante. Il est composé d’un contrôleur à rétroaction sur le DCM [Car19a], qui calcule la force de contact à appliquer pour compenser les déviations de la trajectoire de marche et rejeter les perturbations. Pour réaliser cette force sur un robot contrôlé en position, on utilise un contrôle par admittance au niveau du centre de masse [Kaj03a]. Les flexibilités mécaniques de l’exosquelette Atalante [Vig18] ont rendu nécessaire l’inclusion d’un compensateur de flexibilité.

Nous commencerons par présenter l’architecture et les différents éléments du contrôleur utilisé sur Atalante. Nous présenterons ensuite le stabilisateur inspiré du stabilisateur proposé par Caron pour le robot humanoïde HRP4 [Car19a]. Nous verrons comment la compensation de flexibilités pour le robot Atalante permet d’améliorer les performances. Nous aborderons finalement les expériences réalisées pour valider le schéma de contrôle, tout d’abord avec un mannequin puis avec des pilotes valides et paraplégiques.

Encadrement

Ce travail a été réalisé au sein de l’entreprise Wandercraft, sous l’encadrement de Stanislas Brossette et Guilhem Boéris.

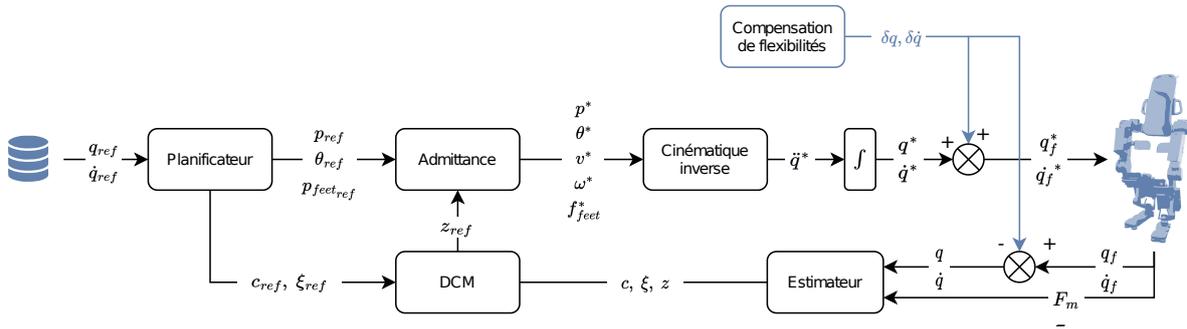


FIGURE 34 – Présentation du schéma de contrôle utilisé pour la stabilisation de l'exosquelette Atalante

A.1 Contrôleur d'Atalante

L'architecture Notre schéma de contrôle (voir Fig. 34) est composé des éléments suivants :

- Un planificateur calcule les consignes cinématiques nominales $*_{nom}$
- Un stabilisateur calcule des consignes adaptées $*_{ref}$ pour compenser une déviation de l'état centroïdal du système
- Une cinématique inverse hiérarchique donne une consigne d'accélération articulaire $\ddot{\mathbf{q}}^*$ qui est ensuite intégrée
- Une compensation de flexibilités applique un décalage sur les consignes articulaires \mathbf{q}^* et $\dot{\mathbf{q}}^*$
- Le contrôleur en position bas-niveau suit ces consignes \mathbf{q}_f^* et $\dot{\mathbf{q}}_f^*$
- Un estimateur d'état estime l'état du robot \mathbf{q}_f et $\dot{\mathbf{q}}_f$

Dans cette partie, nous nous proposons de présenter ces différents éléments.

A.1.1 Planificateur

Les trajectoires de marche sont générées hors ligne par la méthode de la dynamique zéro hybride (voir Section 1.1.4.1) combinée à une optimisation par collocation directe [Gur18b; Her18]. Une trajectoire est générée spécifiquement pour chaque patient en prenant en compte ses dimensions et son poids.

La marche est composée d'un cycle périodique constitué d'un pas gauche et d'un pas droit. Ces pas n'ont qu'une phase de simple support et peuvent être joués dans une boucle alternée infinie. Pour entrer dans la boucle à partir d'une position immobile debout, ou sortir de la boucle vers la même position debout, des pas de départ et d'arrêt sont également générés hors ligne. Ces pas ont une phase de double support et une phase de simple support.

Afin de prendre en compte les incertitudes du modèle et du terrain, une fonctionnalité optionnelle a été implémentée dans le planificateur, appelée *changement au contact*. Elle utilise les capteurs de force des pieds pour détecter l'instant de l'impact. Dès qu'un impact est détecté et même si la trajectoire du pas n'a pas encore été exécutée entièrement, le planificateur passe au pas suivant. Ces transitions créent des discontinuités dans les consignes articulaires, qui doivent donc être lissées.

En utilisant ce planificateur pour calculer les consignes articulaires (position et vitesse) et en les suivant directement avec le contrôleur bas-niveau (Proportionnel Intégral Dérivé - PID - à fort gain), des patients paraplégiques ont pu, avec un peu d'entraînement, marcher sans béquilles ou toute autre assistance [Gur18b]. Cependant, cela n'est possible qu'avec des patients entraînés sur un sol parfaitement plat. On note par exemple que l'exosquelette ne marche pas de manière stable lorsqu'il est chargé d'un mannequin passif. Nous proposons donc la formulation d'un stabilisateur qui permet d'améliorer la robustesse de la marche.

A.1.2 Modèle réduit pour le contrôle de la marche

Le DCM, aussi appelé *capture-point* [Pra06a], peut être calculé en utilisant le modèle du pendule inverse linéarisé (LIPM) [Kaj01] :

$$\xi = \mathbf{c} + \frac{\dot{\mathbf{c}}}{\omega} \quad (\text{A.1})$$

où \mathbf{c} est la position du CoM, $\dot{\mathbf{c}}$ sa vitesse, $\omega = \sqrt{\frac{g}{h}}$ la fréquence propre du pendule inverse, g étant la gravité et h la hauteur du CoM (remarquons que ω n'a ici rien à voir avec la vitesse angulaire).

Ces concepts, bien qu'abscons pour le novice, sont classiques en robotique bipède. On réfère le lecteur intéressé à [Wie02] pour une introduction pédagogique.

A.1.3 Cinématique inverse

Le stabilisateur utilise une cinématique inverse hiérarchique pour calculer l'accélération articulaire à appliquer pour suivre des consignes cinématiques. Les consignes sont les positions des pieds, de la base, du centre de masse (CoM) et une régularisation de la configuration articulaire.

A.1.4 Estimation d'état

Pour calculer la position du CoM et du DCM, nous supposons que le pied de support est immobile et à plat sur le sol pendant le pas (ou un pied choisi arbitrairement pendant les phases de double support). Avec cette hypothèse, le repère de la semelle du pied d'appui peut être choisi comme repère global et nous pouvons utiliser une cinématique directe avec les positions et les vitesses articulaires mesurées par les encodeurs pour estimer la position et la vitesse du CoM, puis du DCM en utilisant l'équation A.1.

A.2 Stabilisation par admittance

Le stabilisateur que nous avons mis en place sur l'exosquelette Atalante est inspiré de celui proposé par Caron [Car19a]. Il permet de compenser des déviations de trajectoires nominales. Il est composé d'un contrôle du DCM qui calcule des consignes pour le ZMP permettant de

corriger des déviations du **DCM** de sa valeur nominale. Ces consignes sont ensuite appliquées grâce à une admittance à deux niveaux : le **CoM** et les chevilles.

A.2.1 Contrôle du DCM

Nous utilisons un contrôle du **DCM** par rétroaction pour calculer le **ZMP** à appliquer afin de compenser les déviations de la trajectoire nominale :

$$\mathbf{z}_{ref} = \mathbf{z}^{nom} - \left(1 + \frac{K_p}{\omega}\right) e_\xi - K_i \int e_\xi + \frac{K_d}{\omega} e_z \quad (\text{A.2})$$

où \mathbf{z}_{ref} est le **ZMP** de référence à réaliser, \mathbf{z}^{nom} la valeur nominale du **ZMP** donné par le planificateur, e_ξ est l'erreur entre le **DCM** de référence et estimé, et e_z l'erreur en **ZMP** de référence et estimé. K_p , K_i et K_d sont les gains de rétroaction à régler.

A.2.2 Admittance du CoM

Cette consigne de **ZMP** est suivie en utilisant de l'admittance. Deux stratégies d'admittance peuvent être employées, au niveau du **CoM** ou au niveau des chevilles. L'admittance au **CoM** consiste à calculer une consigne d'accélération du centre de masse à partir de l'erreur entre le **ZMP** de référence et le **ZMP** estimé. On obtient donc la formule suivante :

$$\ddot{\mathbf{c}} = \ddot{\mathbf{c}}^d + A_c (\mathbf{z} - \mathbf{z}) \quad (\text{A.3})$$

où \mathbf{z} est la position du **ZMP** estimée par l'estimateur et A_c le gain d'admittance du **CoM**.

A.2.3 Admittance au niveau des chevilles

Les allures de marche utilisées sur l'exosquelette ont majoritairement des phases de simple support. Nous avons donc utilisé l'admittance à la cheville uniquement lors de ces phases, ce qui nous permet d'avoir l'équivalence entre **ZMP** et centre de pression du pied de support. La stratégie d'admittance qu niveau de la cheville agit directement sur la vitesse angulaire du pied de support :

$$\begin{bmatrix} \dot{\theta} \\ \dot{\phi} \end{bmatrix} = A_a (\tau - \mathbf{z} \times \mathbf{F}) \quad (\text{A.4})$$

où θ et ϕ sont les angles de roulis et de tangage du pied de support, τ et \mathbf{F} sont le couple et la force de contact entre le pied et l'environnement (donc entre le robot et l'environnement).

Les expériences réalisées avec admittance à la cheville n'ont pas été concluantes et la validation expérimentale présentée section A.4 est donc effectuée sans cette partie.

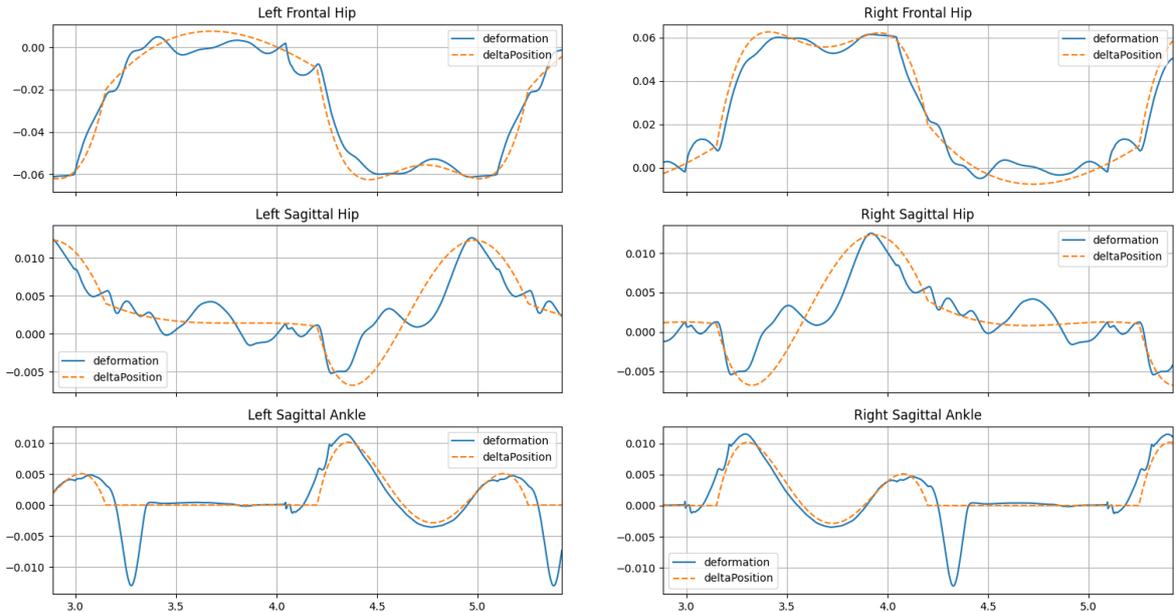


FIGURE 35 – Déformations dues aux flexibilités mécaniques de l’exosquelette et décalages appliqués pour les compenser

A.3 Compensation de flexibilités

Comme la plupart des robots, l’exosquelette d’Atalante présente des flexibilités mécaniques, principalement au niveau des articulations. Elles sont décrites dans [Vig18] et un observateur a été proposé dans [Vig19] utilisant des IMUs pour estimer les déformations dues aux flexibilités et mettre à jour les estimations de position et de vitesse de l’état.

Ces flexibilités dégradent le suivi de la marche. En particulier, le pied de vol est environ 3 cm en dessous de sa position de consigne. En conséquence, l’impact se produit souvent trop tôt. Une phase de double support apparaît lorsque le changement de contact à l’impact (voir Section A.1.1) n’est pas utilisé. Elle se traduit par des pas plus courts et une démarche de marche plus lente. La robustesse de la marche souffre également de ces flexibilités puisque le pied oscillant n’impacte pas le sol avec la bonne orientation.

Un contrôle en boucle fermée de la position du pied de vol a été essayé, sans succès. Malgré la précision de l’observateur, le bruit et le retard créaient des instabilités dans le contrôle. Le contrôle en boucle fermée de l’état flexible a été testé avec succès en simple support [Vig20]. Cependant, les tests sur des trajectoires de marche, qui présentent des mouvements plus rapides et des impacts périodiques, n’ont pas été satisfaisants, car les performances de ce contrôleur sont limitées par le bruit et le retard. Par conséquent, nous avons décidé de compenser les flexibilités avec un décalage calculé hors ligne pour une trajectoire de marche donnée. Nous avons appliqué ce décalage directement sur les cibles articulaires en position et vitesse données au contrôleur bas-niveau. Afin de le rendre transparent pour le stabilisateur, nous l’avons soustrait aux mesures de l’encodeur utilisées par l’estimateur d’état.

A.3.1 Calcul du décalage à appliquer

Pour déterminer le décalage articulaire à appliquer, nous avons utilisé le simulateur Jiminy [Jim]. Il s'agit d'un simulateur open-source codé en C++ exploitant la librairie Pinocchio [Car19b]. Il permet de réaliser des simulations rapides et répétables et un prototypage rapide en Python. Nous avons utilisé un modèle flexible du robot fondé sur le modèle dynamique des flexibilités présenté dans [Vig20]. Bien que le modèle ne corresponde pas parfaitement à la réalité, le comportement du robot en simulation semble similaire à nos observations expérimentales.

Nous avons simulé la marche du robot et observé les déformations liées aux flexibilités sur chaque articulation. Nous avons ensuite appliqué un décalage sur les consignes de position et vitesse des articulations sagittales et frontales des hanches et sur la cheville sagittale de la jambe d'appui. Nous avons utilisé des courbes de Bézier pour ajuster ce décalage au cours du pas pour chaque articulation. La figure 35 présente les déformations estimées sur ces articulations, ainsi que les polynômes ajustés. Pour être robuste face à une grande variété de patients, nous avons choisi les points de passage des courbes de Bézier pour correspondre uniquement aux variations globales des déformations.

A.3.2 Application du décalage

Nous avons choisi d'appliquer ce décalage directement avant l'envoi des consignes au contrôleur bas-niveau. Pour rendre ce décalage invisible pour le contrôleur haut-niveau, nous avons soustrait le même décalage des positions et vitesses articulaires mesurées par les encodeurs et transmis au contrôleur haut-niveau.

Les expériences réalisées dans une simulation avec flexibilités pour différents profils de patients ont montré que cette compensation de flexibilités pouvait stabiliser des marches autrement instables.

A.3.3 Impact sur la trajectoire du pied de vol

Un des problèmes causés par les flexibilités était le décalage entre la consigne de position du pied de vol et sa position réelle. Trop bas, le pied de vol restait en pratique au sol sur la première partie du pas et l'impact avait lieu avant la fin du pas. En désactivant le *changement à l'impact*, nous avons pu comparer avec les trajectoires obtenues avec la compensation de flexibilités.

Nous avons donc réalisé des expériences avec patient dans l'exosquelette et étudié les positions estimées des pieds de vol avec et sans compensation de flexibilités. La figure 36 montre la trajectoire en Z d'un pied de vol dans le cas d'une marche sans compensations et avec compensation. Nous pouvons y voir la courbe violette qui correspond à l'estimation à partir d'un modèle du robot avec flexibilités qui se rapproche de la courbe bleue de la trajectoire désirée quand on active la compensation de flexibilités.

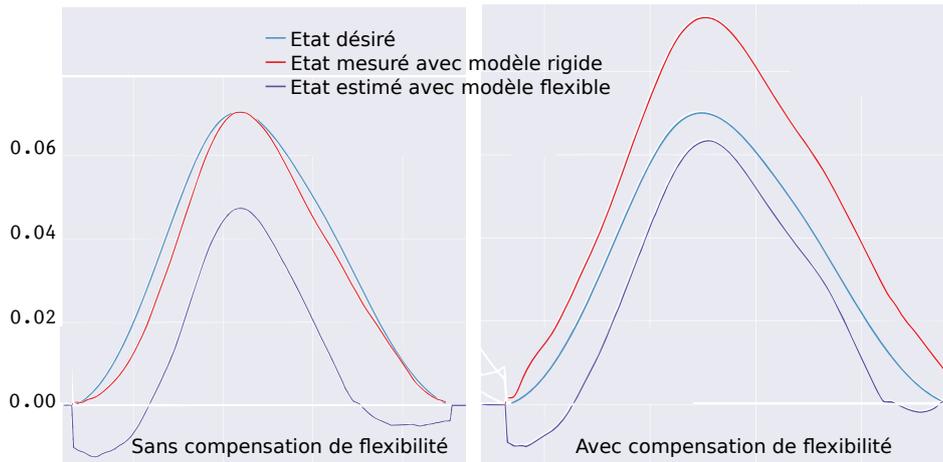


FIGURE 36 – Trajectoires en Z des pieds de vols lors d’expériences sans (gauche) et avec (droite) compensation de flexibilités

A.4 Résultats expérimentaux

Pour assurer la sécurité des patients, nous avons réalisé de premières expériences avec un mannequin dans l’exosquelette.

Bien que le mannequin utilisé ait une taille et un poids comparables à ceux d’un humain, nous avons observé expérimentalement qu’il est plus difficile d’obtenir une marche stable avec lui qu’avec un patient, valide ou handicapé. Sans stabilisation, la marche avec mannequin ne durait pas plus de quelques pas. Le stabilisateur a permis à l’exosquelette de marcher sans tomber avec le mannequin, de résister à des perturbations et d’imprécisions dans le modèle (poids ajouté sur le mannequin non pris en compte dans le modèle).

Nous avons ensuite réalisé des expériences avec des patients dans l’exosquelette. Tout d’abord des volontaires au sein de l’entreprise, puis des patients paraplégiques volontaires. Nous avons obtenu une marche stable sans entraînement pour la totalité des patients. Nous avons ensuite comparé la marche avec et sans stabilisation pour les patients entraînés. Nous avons fait des tests à l’aveugle (le patient ne savait pas s’il marchait avec ou sans le stabilisateur) et avons observé chez la plupart des patients un sentiment de stabilité accru par rapport à la marche sans stabilisation.

Des expériences pour évaluer de manière plus quantitative l’apport du stabilisateur étaient prévues, mais n’ont pas pu être réalisées.

Conclusion

Ce travail préliminaire réalisé en début de thèse a permis de développer le premier stabilisateur permettant une marche autonome sur l’exosquelette Atalante. Le stabilisateur est fondé sur un contrôle du DCM par admittance au niveau du centre de masse. Une compensation de flexibilité a été ajoutée pour améliorer ses performances sur l’exosquelette Atalante. Le



FIGURE 37 – Expériences de marche avec mannequin et patient valide. Lien pour des vidéos : [\[Vid\]](#)

stabilisateur a été validé lors d'expériences de marche avec mannequin et avec des pilotes paraplégiques. Une vidéo des expériences est disponible [\[Vid\]](#).

L'ajout de l'admittance à au niveau des chevilles est une perspective qui avait commencé à être étudiée. Une meilleure compensation de flexibilités mériterait aussi d'être utilisée pour améliorer le contrôle.

Bibliographie

- [Ris22] Fanny RISBOURG, Thomas CORBÈRES, Pierre-Alexandre LÉZIART, Thomas FLAYOLS, Nicolas MANSARD et Steve TONNEAU. « Real time footstep planning and control of the Solo quadruped robot in 3D environments ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022 (cf. p. 7, 40, 41, 51, 83, 84).
- [Ass23] Alessandro ASSIRELLI, Fanny RISBOURG, Gianni LUNARDI, Thomas FLAYOLS et Nicolas MANSARD. « Whole-Body MPC without Foot References for the Locomotion of an Impedance-Controlled Robot ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2023 (cf. p. 7, 79).
- [Ris] « Procédé de mise en mouvement de l'exosquelette ». Brev. FR2021/052335. Fanny RISBOURG, Gabriele BUONDONNO, Roch MOLLERO, Stanislas BROSSETTE et Guilhem BOÉRIS. 2021 (cf. p. 7, 11).
- [Lau17] Jean-Paul LAUMOND, Mehdi BENALLEGUE, Justin CARPENTIER et Alain BERTHOZ. « The yoyo-man ». In : *The International Journal of Robotics Research* (2017) (cf. p. 1).
- [Lau16] Jean-Paul LAUMOND. « Robotics : Hephaestus does it again ». In : *Robots and Art*. Springer, 2016 (cf. p. 1).
- [Ros94] M.E. ROSHEIM. *Robot Evolution : The Development of Anthrobotics*. Wiley inter-science publication. Wiley, 1994 (cf. p. 2).
- [Rai86] M.H. RAIBERT. *Legged Robots that Balance*. Artificial Intelligence. MIT Press, 1986 (cf. p. 2).
- [Wal] *GE's Walking Truck - ca. 1965*. <https://www.youtube.com/watch?v=coN09FpDb6E> (cf. p. 2).
- [McK91] P. MCKERROW, H. ZIMA et B. CHAPMAN. *Introduction to Robotics*. ACM Press frontier series. Addison-Wesley, 1991 (cf. p. 2).
- [Mos68] Ralph S MOSHER. « Handyman to hardiman ». In : *Sae Transactions* (1968) (cf. p. 2).
- [Hir84] Shigeo HIROSE. « A Study of Design and Control of a Quadruped Walking Vehicle ». In : *The International Journal of Robotics Research* (1984) (cf. p. 2).
- [Rob60] James Kerr ROBERGE. « The mechanical seal ». Thèse de doct. Massachusetts Institute of Technology, 1960 (cf. p. 2).
- [Hir07] Masato HIROSE et Kenichi OGAWA. « Honda humanoid robots development ». In : *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences* (2007) (cf. p. 3).

- [Hir04] Hirohisa HIRUKAWA, Fumio KANEHIRO, Kenji KANEKO, Shuuji KAJITA, Kiyoshi FUJIWARA, Yoshihiro KAWAI, Fumiaki TOMITA, Shigeoki HIRAI, Kazuo TANIE, Takakatsu ISOZUMI et al. « Humanoid robotics platforms developed in HRP ». In : *Robotics and Autonomous Systems* (2004) (cf. p. 3).
- [Bel18a] C. Dario BELLICOSO, Fabian JENELTEN, Christian GEHRING et Marco HUTTER. « Dynamic Locomotion Through Online Nonlinear Motion Optimization for Quadrupedal Robots ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2018) (cf. p. 3).
- [Sem11] Claudio SEMINI, Nikos G TSAGARAKIS, Emanuele GUGLIELMINO, Michele FOCCHI, Ferdinando CANNELLA et Darwin G CALDWELL. « Design of HyQ—a hydraulically and electrically actuated quadruped robot ». In : *Proceedings of the Institution of Mechanical Engineers, Part I : Journal of Systems and Control Engineering* (2011) (cf. p. 3).
- [DC18a] Jared DI CARLO, Patrick M WENSING, Benjamin KATZ, Gerardo BLEDT et Sangbae KIM. « Dynamic locomotion in the mit cheetah 3 through convex model-predictive control ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018 (cf. p. 3).
- [Kui20] Scott KUINDERSMA. « Recent progress on atlas, the world’s most dynamic humanoid robot ». In : *This recorded lecture provides precise details about the complete control design behind Boston Dynamics’ world-leading robot videos* (2020) (cf. p. 3).
- [Sta17] Olivier STASSE, Thomas FLAYOLS, Rohan BUDHIRAJA, Kevin GIRAUD-ESCLASSE, Justin CARPENTIER, Joseph MIRABEL, Andrea DEL PRETE, Philippe SOUÈRES, Nicolas MANSARD, Florent LAMIRAUX et al. « TALOS : A new humanoid research platform targeted for industrial applications ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)*. 2017 (cf. p. 3).
- [Gon19a] Yukai GONG, Ross HARTLEY, Xingye DA, Ayonga HEREID, Omar HARIB, Jiunn-Kai HUANG et Jessy GRIZZLE. « Feedback control of a cassie bipedal robot : Walking, standing, and riding a segway ». In : *2019 American Control Conference (ACC)*. 2019 (cf. p. 3).
- [May18] Adrienne MAYOR. « Gods and robots ». In : *Gods and Robots*. Princeton University Press, 2018 (cf. p. 3).
- [Rai08] Marc RAIBERT, Kevin BLANKESPOOR, Gabriel NELSON et Rob PLAYTER. « Big-dog, the rough-terrain quadruped robot ». In : *IFAC Proceedings Volumes* (2008) (cf. p. 3).
- [Oba11] Barack OBAMA. *Remarks by the President at Carnegie Mellon University’s National Robotics Engineering Center*. <https://obamawhitehouse.archives.gov/the-press-office/2011/06/24/remarks-president-carnegie-mellon-university-national-robotics-engineer>. 2011 (cf. p. 3).
- [Lin17] Patrick LIN, Keith ABNEY et Ryan JENKINS. *Robot ethics 2.0 : From autonomous cars to artificial intelligence*. Oxford University Press, 2017 (cf. p. 3).

- [Sip] *Autonomous Weapon Systems and International Humanitarian Law*. <https://www.sipri.org/publications/2021/other-publications/autonomous-weapon-systems-and-international-humanitarian-law-identifying-limits-and-required-type>. SIPRI, Stockholm, 2021 (cf. p. 3).
- [Spr13] Paul J SPRINGER. *Military robots and drones : a reference handbook*. ABC-CLIO, 2013 (cf. p. 3).
- [Liv] *Living under drones*. <https://www-cdn.law.stanford.edu/wp-content/uploads/2015/07/Stanford-NYU-LIVING-UNDER-DRONES.pdf>. Stanford International Human Rights, Conflict Resolution Clinic et Global Justice Clinic (GJC) at NYU School of Law, 2012 (cf. p. 3).
- [Hrw] *Between a Drone and Al-Qaeda*. <https://www.hrw.org/report/2013/10/22/between-drone-and-al-qaeda/civilian-cost-us-targeted-killings-yemen>. Human Rights Watch, 2013 (cf. p. 3).
- [Ple] *Open letter : General Purpose Robots Should Not Be Weaponized*. <https://www.bostondynamics.com/open-letter-opposing-weaponization-general-purpose-robots>. Boston Dynamics, 2022 (cf. p. 3).
- [Sto] *Stop killer robots*. <https://www.stopkillerrobots.org/> (cf. p. 3).
- [Gha21] Moojan GHAFURIAN, Colin ELLARD et Kerstin DAUTENHAHN. « Social companion robots to reduce isolation : a perception change due to covid-19 ». In : *IFIP Conference on Human-Computer Interaction*. Springer. 2021 (cf. p. 3).
- [GA18] Susel GÓNGORA ALONSO, Sofiane HAMRIOUI, Isabel De la TORRE DÍEZ, Eduardo CRUZ, Miguel LOPEZ-CORONADO et Manuel FRANCO. « Social Robots for People with Aging and Dementia : A Systematic Review of Literature ». In : *Telemedicine and e-Health* (2018) (cf. p. 3).
- [Coo20] Sara COOPER, Alessandro DI FAVA, Carlos VIVAS, Luca MARCHIONNI et Francesco FERRO. « ARI : the Social Assistive Robot and Companion ». In : *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 2020 (cf. p. 3).
- [Ack18] Evan ACKERMAN. *Jibo Is Probably Totally Dead Now*. <https://spectrum.ieee.org/jibo-is-probably-totally-dead-now>. 2018 (cf. p. 3).
- [Tul19] Silvia TULLI, Diego Agustin AMBROSSIO, Amro NAJJAR et Francisco Javier Rodriguez LERA. « Great Expectations & Aborted Business Initiatives : The Paradox of Social Robot Between Research and Industry. » In : *BNAIC/BENELEARN*. 2019 (cf. p. 3).
- [Bel18c] Tony BELPAEME, James KENNEDY, Aditi RAMACHANDRAN, Brian SCASSELLATI et Fumihide TANAKA. « Social robots for education : A review ». In : *Science robotics* (2018) (cf. p. 3).
- [Hen21] Anna HENSCHER, Guy LABAN et Emily S CROSS. « What makes a robot social? a review of social robots from science fiction to a home or hospital near you ». In : *Current Robotics Reports* (2021) (cf. p. 3).

- [Høy22] Ellen HØYER, Arve OPHEIM et Vivien JØRGENSEN. « Implementing the exoskeleton Ekso GTTM for gait rehabilitation in a stroke unit—feasibility, functional benefits and patient experiences ». In : *Disability and Rehabilitation : Assistive Technology* (2022) (cf. p. 3).
- [Gur18a] Thomas GURRIET, Sylvain FINET, Guilhem BOERIS, Alexis DUBURCQ, Ayonga HEREID, Omar HARIB, Matthieu MASSELIN, Jessy GRIZZLE et Aaron D AMES. « Towards restoring locomotion for paraplegics : Realizing dynamically stable walking on exoskeletons ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2018 (cf. p. 3).
- [Cal18] Rocco Salvatore CALABRÒ, Antonino NARO, Margherita RUSSO, Placido BRAMANTI, Luigi CARIOTI, Tina BALLETTA, Antonio BUDA, Alfredo MANULLI, Serena FILONI et Alessia BRAMANTI. « Shaping neuroplasticity by using powered exoskeletons in patients with stroke : a randomized clinical trial ». In : *Journal of neuroengineering and rehabilitation* (2018) (cf. p. 3).
- [Swa19] Chad SWANK, Sharon WANG-PRICE, Fan GAO, Sattam ALMUTAIRI et al. « Walking with a robotic exoskeleton does not mimic natural gait : a within-subjects study ». In : *JMIR rehabilitation and assistive technologies* (2019) (cf. p. 4).
- [Rup16] BS RUPAL, A SINGLA et GS VIRK. « Lower limb exoskeletons : a brief review ». In : *Conference on mechanical engineering and technology (COMET-2016), IIT (BHU), Varanasi, India*. 2016, p. 130-140 (cf. p. 4).
- [FV20] Eduard FOSCH-VILLARONGA et Beste OZCAN. « The progressive intertwinement between design, human needs and the regulation of care technology : the case of lower-limb exoskeletons ». In : *International Journal of Social Robotics* (2020) (cf. p. 4).
- [Edw84] Mary EDWARDS. « Robots in industry : An overview ». In : *Applied ergonomics* (1984) (cf. p. 4).
- [Bay18] Barış and İnce, Gokhan BAYRAM. « Advances in Robotics in the Era of Industry 4.0 ». In : *Industry 4.0 : Managing The Digital Transformation*. Springer, 2018 (cf. p. 4).
- [Büh17] Christian BÜHRER et Christian HAGIST. « The effect of digitalization on the labor market ». In : *The Palgrave Handbook of managing continuous business transformation*. Springer, 2017 (cf. p. 4).
- [Mol22] Ian MOLL. « The Fourth Industrial Revolution : A New Ideology ». In : *tripleC : Communication, Capitalism & Critique. Open Access Journal for a Global Sustainable Information Society* (2022) (cf. p. 4).
- [Buh19] Daniel BUHR. « Why do smart factories need smart welfare states ». In : *Bürkhardt et al.(eds.)(2019)* (2019) (cf. p. 4).
- [Cor22] Enrique CORONADO, Takuya KIYOKAWA, Gustavo A Garcia RICARDEZ, Ixchel G RAMIREZ-ALPIZAR, Gentiane VENTURE et Natsuki YAMANOBE. « Evaluating quality in human-robot interaction : A systematic search and classification of performance and human-centered factors, measures and metrics towards an industry 5.0 ». In : *Journal of Manufacturing Systems* (2022) (cf. p. 4).

- [Arm19] Philip ARM, Radek ZENKL, Patrick BARTON, Lars BEGLINGER, Alex DIETSCHÉ, Luca FERRAZZINI, Elias HAMPP, Jan HINDER, Camille HUBER, David SCHAUFELBERGER et al. « Spacebok : A dynamic legged robot for space exploration ». In : *2019 international conference on robotics and automation (ICRA)*. 2019 (cf. p. 4).
- [DS07] P Gonzalez DE SANTOS, José A COBANO, E GARCIA, J ESTREMERÁ et MA ARMADA. « A six-legged robot-based system for humanitarian demining missions ». In : *Mechatronics* (2007) (cf. p. 4).
- [Hab08] Maki K HABIB. « Humanitarian demining : The problem, difficulties, priorities, demining technology and the challenge for robotics ». In : *Humanitarian Demining*. IntechOpen, 2008 (cf. p. 4).
- [Zha18] Qihao ZHANG, Wei ZHAO, Shengnan CHU, Lei WANG, Jun FU, Jiangrong YANG et Bo GAO. « Research progress of nuclear emergency response robot ». In : *IOP Conference Series : Materials Science and Engineering*. IOP Publishing. 2018 (cf. p. 4).
- [Dar] *DARPA Subterranean Challenge*. <https://www.subtchallenge.com/> (cf. p. 4).
- [Ly15] Olivier LY, Hugo GIMBERT, Grégoire PASSAULT et Gérald BARON. « A fully autonomous robot for putting posts for trellising vineyard with centimetric accuracy ». In : *IEEE International Conference on Autonomous Robot Systems and Competitions*. 2015 (cf. p. 4).
- [Gre] *Empreinte environnementale du numérique mondial*. <https://www.greenit.fr/empreinte-environnementale-du-numerique-mondial/>. GreenIT, 2019 (cf. p. 4).
- [Ade] *Evaluation de l'impact environnemental du numérique en France et analyse prospective*. <https://librairie.ademe.fr/consommer-autrement/5226-evaluation-de-l-impact-environnemental-du-numerique-en-france-et-analyse-prospective.html>. Ademe, 2022 (cf. p. 5).
- [Cap22] Siméon CAPY, Pablo OSORIO, Shohei HAGANE, Corentin AZNAR, Dora GARCIN, Enrique CORONADO, Dominique DEUFF, Ioana OCNARESCU, Isabelle MILLEVILLE et Gentiane VENTURE. « Yōkobo : A Robot to Strengthen Links Amongst Users with Non-Verbal Behaviours ». In : *Machines* (2022) (cf. p. 5).
- [Wan] *Wandercraft*. <https://en.wandercraft.eu/> (cf. p. 7).
- [Car19a] Stephane CARON, Abderrahmane KHEDDAR et Olivier TEMPIER. « Stair climbing stabilization of the HRP-4 humanoid robot using whole-body admittance control ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)* (2019) (cf. p. 7, 13, 14, 86, 88).
- [Gur18b] Thomas GURRIET, Sylvain FINET, Guilhem BOERIS, Alexis DUBURCQ, Ayonga HEREID, Omar HARIB, Matthieu MASSELIN, Jessy GRIZZLE et Aaron D AMES. « Towards restoring locomotion for paraplegics : Realizing dynamically stable walking on exoskeletons ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2018 (cf. p. 7, 85, 87, 88).

- [Ly11] Olivier LY, Matthieu LAPEYRE et Pierre-Yves OUDEYER. « Bio-inspired vertebral column, compliance and semi-passive dynamics in a lightweight humanoid robot ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2011 (cf. p. 10).
- [Fri06] Alain FRIGON et Serge ROSSIGNOL. « Experiments and models of sensorimotor interactions during locomotion ». In : *Biological cybernetics* (2006) (cf. p. 10).
- [Tag94] Gentaro TAGA. « Emergence of bipedal locomotion through entrainment among the neuro-musculo-skeletal system and the environment ». In : *Physica D : Non-linear Phenomena* (1994) (cf. p. 10).
- [Tag95] Gentaro TAGA. « A model of the neuro-musculo-skeletal system for human locomotion - I. Emergence of basic gait ». In : *Biological Cybernetics* (1995) (cf. p. 10).
- [Lew00] M. Anthony LEWIS, Ralph ETIENNE-CUMMINGS, Avis H. COHEN et Mitra HARTMANN. « Toward biomorphic control using custom a VLSI CPG chips ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)* (2000) (cf. p. 10).
- [Tsu01] Katsuyoshi TSUJITA, Kazuo TSUCHIYA et Ahmet ONAT. « Adaptive gait pattern control of a quadruped locomotion robot ». In : *IEEE International Conference on Intelligent Robots and Systems* (2001) (cf. p. 10).
- [Lew02] M. LEWIS et George BEKEY. « Gait Adaptation in a Quadruped Robot ». In : *Autonomous Robots* (2002) (cf. p. 10).
- [Rig08] Ludovic RIGHETTI et Auke Jan IJSPEERT. « Pattern generators with sensory feedback for the control of quadruped locomotion ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)* (2008) (cf. p. 10).
- [Liu11] Chengju LIU, Qijun CHEN et Danwei WANG. « Locomotion control of quadruped robots based on CPG-inspired workspace trajectory generation ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)* (2011) (cf. p. 11).
- [Bar13] Victor BARASUOL, Jonas BUCHLI, Claudio SEMINI, Marco FRIGERIO, Edson R. DE PIERI et Darwin G. CALDWELL. « A reactive controller framework for quadrupedal locomotion on challenging terrain ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)* (2013) (cf. p. 11).
- [Foc13] Michele FOCCHI, Victor BARASUOL, Ioannis HAVOUTIS, Jonas BUCHLI, Claudio SEMINI et Darwin G. CALDWELL. « Local reflex generation for obstacle negotiation in quadrupedal locomotion ». In : *Nature-Inspired Mobile Robotics* (2013) (cf. p. 11).
- [Aja13] Mostafa AJALLOEIAN, Soha POUYA, Alexander SPROEWITZ et Auke J. IJSPEERT. « Central Pattern Generators augmented with virtual model control for quadruped rough terrain locomotion ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)* (2013) (cf. p. 11).
- [Pra01] Jerry PRATT, Chee Meng CHEW, Ann TORRES, Peter DILWORTH et Gill PRATT. « Virtual model control : an intuitive approach for bipedal locomotion ». In : *International Journal of Robotics Research* (2001) (cf. p. 11).

- [Mas19] Elisa MASSI, Lorenzo VANNUCCI, Ugo ALBANESE, Marie Claire CAPOLEI, Alexander VANDESOMPELE, Gabriel URBAIN, Angelo Maria SABATINI, Joni DAMBRE, Cecilia LASCHI, Silvia TOLU et Egidio FALOTICO. « Combining evolutionary and adaptive control strategies for quadruped robotic locomotion ». In : *Frontiers in Neurorobotics* (2019) (cf. p. 11).
- [Are21] Paolo ARENA, Pierfrancesco SUERI, Salvatore TAFFARA et Luca PATANE. « MPC-based control strategy of a neuro-inspired quadruped robot ». In : *Proceedings of the International Joint Conference on Neural Networks* (2021) (cf. p. 11).
- [Zha21] Dong ZHANG, Wenjie ZHENG, Yansen YANG et Tianyu FANG. « CPG-Based Gait Control Method for Quadruped Robot Jumping Movement ». In : *7th International Conference on Control, Automation and Robotics, ICCAR 2021* (2021) (cf. p. 11).
- [Gep] *Mouvement des Systèmes Anthropomorphes - GEPETTO (LAAS-CNRS)*. <https://www.laas.fr/public/fr/gepetto> (cf. p. 11).
- [Cam13] Eduardo F CAMACHO et Carlos Bordons ALBA. *Model predictive control*. Springer science & business media, 2013 (cf. p. 11, 13).
- [Wie02] Pierre-Brice WIEBER. « On the stability of walking systems ». In : *Proceedings of the international workshop on humanoid and human friendly robotics*. 2002 (cf. p. 12, 88).
- [Wie16] Pierre-Brice WIEBER, Russ TEDRAKE et Scott KUINDERSMA. « Modeling and Control of Legged Robots. Springer Handbook of Robotics ». In : (2016) (cf. p. 12, 13).
- [Pra06b] Jerry E PRATT et Russ TEDRAKE. « Velocity-based stability margins for fast bipedal walking ». In : *Fast motions in biomechanics and robotics*. Springer, 2006 (cf. p. 12).
- [Vuk04] Miomir VUKOBRATOVIC et Branislav BOROVAC. « Zero-Moment Point - Thirty Five Years of its Life. » In : *International journal of Humanoid Robotics* (2004) (cf. p. 12).
- [Kaj03a] Shuuji KAJITA, Fumio KANEHIRO, Kenji KANEKO, Kiyoshi FUJIWARA, Kensuke HARADA, Kazuhito YOKOI et Hirohisa HIRUKAWA. « Biped walking pattern generation by using preview control of zero-moment point ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)* (2003) (cf. p. 12, 13, 86).
- [DP16] Andrea DEL PRETE, Steve TONNEAU et Nicolas MANSARD. « Fast algorithms to test robust static equilibrium for legged robots ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2016 (cf. p. 12, 20).
- [Wie06a] Pierre Brice WIEBER. « Trajectory free linear model predictive control for stable walking in the presence of strong perturbations ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)* (2006) (cf. p. 13).
- [Her10a] Andrei HERDT, Holger DIEDAM, Pierre Brice WIEBER, Dimitar DIMITROV, Katja MOMBAUR et Moritz DIEHL. « Online walking motion generation with automatic footstep placement ». In : *Advanced Robotics* (2010) (cf. p. 13, 53).

- [Dim09] Dimitar DIMITROV, Pierre-Brice WIEBER, Olivier STASSE, Hans Joachim FERREAU et Holger DIEDAM. « An optimized linear model predictive control solver for online walking motion generation ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2009 (cf. p. 13).
- [Die08] Holger DIEDAM, Dimitar DIMITROV, Pierre-Brice WIEBER, Katja MOMBAUR et Moritz DIEHL. « Online walking gait generation with adaptive foot positioning through Linear Model Predictive control ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2008 (cf. p. 13).
- [Her10b] Andrei HERDT, Nicolas PERRIN et Pierre-Brice WIEBER. « Walking without thinking about it ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2010) (cf. p. 13).
- [Nav17] M. NAVEAU, M. KUDRUSS, O. STASSE, C. KIRCHES, K. MOMBAUR et P. SOUÈRES. « A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2017) (cf. p. 13).
- [Kaj03b] Shuuji KAJITA, Fumio KANEHIRO, Kenji KANEKO, Kiyoshi FUJIWARA, Kensuke HARADA, Kazuhito YOKOI et Hirohisa HIRUKAWA. « Resolved momentum control : Humanoid motion planning based on the linear and angular momentum ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2003 (cf. p. 13).
- [Fuj98] Yasutaka FUJIMOTO et Atsuo KAWAMURA. « Simulation of an autonomous biped walking robot including environmental force interaction ». In : *IEEE Robotics & Automation Magazine* (1998) (cf. p. 13).
- [Esc10] Adrien ESCANDE, Nicolas MANSARD et Pierre-Brice WIEBER. « Fast resolution of hierarchized inverse kinematics with inequality constraints ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2010 (cf. p. 13).
- [Esc14] Adrien ESCANDE, Nicolas MANSARD et Pierre-Brice WIEBER. « Hierarchical quadratic programming : Fast online humanoid-robot motion generation ». In : *The International Journal of Robotics Research* (2014) (cf. p. 13, 38).
- [Koo12] Twan KOOLEN, Tomas DE BOER, John REBULA, Ambarish GOSWAMI et Jerry PRATT. « Capturability-based analysis and control of legged locomotion, Part 1 : Theory and application to three simple gait models ». In : *The international journal of robotics research* (2012) (cf. p. 13).
- [Eng11] Johannes ENGLSBERGER, Christian OTT, Máximo A ROA, Alin ALBU-SCHÄFFER et Gerhard HIRZINGER. « Bipedal walking control based on capture point dynamics ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2011 (cf. p. 13).
- [Bel18b] C Dario BELLICOSO, Fabian JENELTEN, Christian GEHRING et Marco HUTTER. « Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2018) (cf. p. 13, 22).

- [Mas17] Carlos MASTALLI, Michele FOCCHI, Ioannis HAVOUTIS, Andreea RADULESCU, Sylvain CALINON, Jonas BUCHLI, Darwin G CALDWELL et Claudio SEMINI. « Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2017 (cf. p. 13).
- [DC18b] Jared DI CARLO, Patrick M WENSING, Benjamin KATZ, Gerardo BLEDT et Sangbae KIM. « Dynamic locomotion in the mit cheetah 3 through convex model-predictive control ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018 (cf. p. 13).
- [Liu12] Mingxing LIU, Alain MICAELLI, Paul EVRARD, Adrien ESCANDE et Claude ANDRIOT. « Interactive virtual humans : A two-level prioritized control framework with wrench bounds ». In : *IEEE Transactions on Robotics (T-RO)* (2012) (cf. p. 13).
- [Car15a] Stéphane CARON, Quang-Cuong PHAM et Yoshihiko NAKAMURA. « Stability of surface contacts for humanoid robots : Closed-form formulae of the contact wrench cone for rectangular support areas ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2015 (cf. p. 13).
- [Dai16] Hongkai DAI et Russ TEDRAKE. « Planning robust walking motion on uneven terrain via convex optimization ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)* (2016) (cf. p. 13).
- [Kaj10] Shuuji KAJITA, Mitsuharu MORISAWA, Kanako MIURA, Shin'ichiro NAKAOKA, Kensuke HARADA, Kenji KANEKO, Fumio KANEHIRO et Kazuhito YOKOI. « Biped walking stabilization based on linear inverted pendulum tracking ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2010 (cf. p. 14).
- [Kim09] Jung-Yup KIM, Ill-Woo PARK et Jun-Ho OH. « Realization of dynamic stair climbing for biped humanoid robot using force/torque sensors ». In : *Journal of Intelligent and Robotic Systems* (2009) (cf. p. 14).
- [Bou18] Karim BOUYARMANE, Kevin CHAPPELLET, Joris VAILLANT et Abderrahmane KHEDDAR. « Quadratic programming for multirobot and task-space force control ». In : *IEEE Transactions on Robotics (T-RO)* (2018) (cf. p. 14).
- [Sug02] Tomomichi SUGIHARA et Yoshihiko NAKAMURA. « Whole-body cooperative balancing of humanoid robot using cog jacobian ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2002 (cf. p. 14).
- [Eng12] Johannes ENGLSBERGER et Christian OTT. « Integration of vertical com motion and angular momentum in an extended capture point tracking controller for bipedal walking ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)*. 2012 (cf. p. 14).
- [Kim19] Donghyun KIM, Jared DI CARLO, Benjamin KATZ, Gerardo BLEDT et Sangbae KIM. « Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control ». In : (2019) (cf. p. 14).

- [Léz21] Pierre-Alexandre LÉZIART, Thomas FLAYOLS, Felix GRIMMINGER, Nicolas MANSARD et Philippe SOUÈRES. « Implementation of a Reactive Walking Controller for the New Open-Hardware Quadruped Solo-12 ». In : (2021) (cf. p. 14, 19, 25, 36, 72).
- [Ble20] Gerardo BLEDT et Sangbae KIM. « Extracting legged locomotion heuristics with regularized predictive control ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2020 (cf. p. 14).
- [Cor21] Thomas CORBÈRES, Thomas FLAYOLS, Pierre-Alexandre LÉZIART, Rohan BUDHIRAJA, Philippe SOUÈRES, Guilhem SAUREL et Nicolas MANSARD. « Comparison of predictive controllers for locomotion and balance recovery of quadruped robots ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2021 (cf. p. 14, 15, 58).
- [Ble19] Gerardo BLEDT et Sangbae KIM. « Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019 (cf. p. 15).
- [Léz22] Pierre-Alexandre LÉZIART, Thomas CORBÈRES, Thomas FLAYOLS, Steve TONNEAU, Nicolas MANSARD et Philippe SOUÈRES. « Improved Control Scheme for the Solo Quadruped and Experimental Comparison of Model Predictive Controllers ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2022) (cf. p. 15, 36).
- [Wie06b] Pierre-Brice WIEBER et Christine CHEVALLEREAU. « Online adaptation of reference trajectories for the control of walking systems ». In : *Robotics and Autonomous Systems* (2006) (cf. p. 15).
- [Guo21] Yijie GUO, Mingwei ZHANG, Hao DONG et Mingguo ZHAO. « Fast online planning for bipedal locomotion via centroidal model predictive gait synthesis ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2021) (cf. p. 15, 16).
- [Atk92] Christopher G ATKESON. « Memory-based approaches to approximating continuous functions ». In : *SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-*. ADDISON-WESLEY PUBLISHING CO. 1992 (cf. p. 15).
- [Liu13] Chenggang LIU, Christopher G ATKESON et Jianbo SU. « Biped walking control using a trajectory library ». In : *Robotica* (2013) (cf. p. 15, 16).
- [McG90] Tad MCGEER et al. « Passive dynamic walking ». In : *International Journal Robotics Research* (1990) (cf. p. 15).
- [Wes18a] Eric R WESTERVELT, Jessy W GRIZZLE, Christine CHEVALLEREAU, Jun Ho CHOI et Benjamin MORRIS. *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018 (cf. p. 15).
- [Wes18b] Eric R WESTERVELT, Jessy W GRIZZLE, Christine CHEVALLEREAU, Jun Ho CHOI et Benjamin MORRIS. *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018 (cf. p. 15).

- [Liu09] Chenggang LIU et Christopher G ATKESON. « Standing balance control using a trajectory library ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2009 (cf. p. 16).
- [Da16] Xingye DA, Omar HARIB, Ross HARTLEY, Brent GRIFFIN et Jessy W GRIZZLE. « From 2D design of underactuated bipedal gaits to 3D implementation : Walking with speed tracking ». In : *IEEE Access* (2016) (cf. p. 16).
- [Gon19b] Yukai GONG, Ross HARTLEY, Xingye DA, Ayonga HEREID, Omar HARIB, Jiunn-Kai HUANG et Jessy GRIZZLE. « Feedback control of a cassie bipedal robot : Walking, standing, and riding a segway ». In : *American Control Conference (ACC)*. IEEE. 2019 (cf. p. 16).
- [Ngu20] Quan NGUYEN, Xingye DA, JW GRIZZLE et Koushil SREENATH. « Dynamic walking on stepping stones with gait library and control barrier functions ». In : *Algorithmic Foundations of Robotics XII*. Springer, 2020 (cf. p. 16).
- [Ma19] Wen Loong MA, Kaveh Akbari HAMED et Aaron D. AMES. « First Steps Towards Full Model Based Motion Planning and Control of Quadrupeds : A Hybrid Zero Dynamics Approach ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019) (cf. p. 16).
- [Liu15] Xin LIU, Claudio SEMINI et Ioannis POULAKAKIS. « Active compliance hybrid zero dynamics control of bounding on HyQ ». In : *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2015 (cf. p. 16).
- [Ere15] Tom EREZ, Kendall LOWREY, Yuval TASSA, Vikash KUMAR, Svetoslav KOLEV et Emanuel TODOROV. « An integrated system for real-time model predictive control of humanoid robots ». In : *IEEE-RAS International Conference on Humanoid Robots* (2015) (cf. p. 17).
- [Koe15] Jonas KOENEMANN, Andrea DEL PRETE, Yuval TASSA, Emanuel TODOROV, Olivier STASSE, Maren BENNEWITZ et Nicolas MANSARD. « Whole-body model-predictive control applied to the HRP-2 humanoid ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015 (cf. p. 17).
- [Neu18] Michael NEUNERT, Markus STÄUBLE, Markus GIFTTHALER, Carmine D BELLICOSO, Jan CARIUS, Christian GEHRING, Marco HUTTER et Jonas BUCHLI. « Whole-body nonlinear model predictive control through contacts for quadrupeds ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2018) (cf. p. 17).
- [Mas20] Carlos MASTALLI, Rohan BUDHIRAJA, Wolfgang MERKT, Guilhem SAUREL, Bilal HAMMOUD, Maximilien NAVEAU, Justin CARPENTIER, Ludovic RIGHETTI, Sethu VIJAYAKUMAR et Nicolas MANSARD. « Crocoddyl : An efficient and versatile framework for multi-contact optimal control ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2020 (cf. p. 17, 33, 72, 74).
- [Dan21] Ewen DANTEC, Rohan BUDHIRAJA, Adria ROIG, Teguh LEMBONO, Guilhem SAUREL, Olivier STASSE, Pierre FERNBACH, Steve TONNEAU, Sethu VIJAYAKUMAR, Sylvain CALINON et al. « Whole body model predictive control with a memory of motion : Experiments on a torque-controlled talos ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2021 (cf. p. 17, 72).

- [Dan22b] Ewen Louis DANTEC, Maximilien NAVEAU, Pierre FERNBACH, Nahuel VILLA, Guilhem SAUREL, Olivier STASSE, Michel TAIX et Nicolas MANSARD. « Whole-Body Model Predictive Control for Biped Locomotion on a Torque-Controlled Humanoid Robot ». In : (2022) (cf. p. 17).
- [Bud19] Rohan BUDHIRAJA, Justin CARPENTIER et Nicolas MANSARD. « Dynamics consensus between centroidal and whole-body models for locomotion of legged robots ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2019 (cf. p. 17, 72).
- [Med22] Avadesh MEDURI, Paarth SHAH, Julian VIERECK, Majid KHADIV, Ioannis HAVOUTIS et Ludovic RIGHETTI. « BiConMP : A Nonlinear Model Predictive Control Framework for Whole Body Motion Planning ». In : *arXiv preprint arXiv :2201.07601* (2022) (cf. p. 17, 72).
- [Gri19b] Felix GRIMMINGER, Avadesh MEDURI, Majid KHADIV, Julian VIERECK, Manuel WÜTHRICH, Maximilien NAVEAU, Vincent BERENZ, Steve HEIM, Felix WIDMAIER, Thomas FLAYOLS, Jonathan FIENE, Alexander BADRI-SPRÖWITZ et Ludovic RIGHETTI. « An Open Torque-Controlled Modular Robot Architecture for Legged Locomotion Research ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2019) (cf. p. 17, 35).
- [Kat22] Sotaro KATAYAMA et Toshiyuki OHTSUKA. « Whole-body model predictive control with rigid contacts via online switching time optimization ». In : *arXiv preprint arXiv :2203.00997* (2022) (cf. p. 17, 54, 72).
- [Dan22a] Ewen DANTEC, Michel TAIX et Nicolas MANSARD. « First Order Approximation of Model Predictive Control Solutions for High Frequency Feedback ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2022) (cf. p. 17, 72).
- [Lee20] Joonho LEE, Jemin HWANGBO, Lorenz WELLHAUSEN, Vladlen KOLTUN et Marco HUTTER. « Learning quadrupedal locomotion over challenging terrain ». In : *Science robotics* (2020) (cf. p. 18, 19).
- [Yan22] Tsung-Yen YANG, Tingnan ZHANG, Linda LUU, Sehoon HA, Jie TAN et Wenhao YU. « Safe Reinforcement Learning for Legged Locomotion ». In : *arXiv preprint arXiv :2203.02638* (2022) (cf. p. 18).
- [Iba21] Julian IBARZ, Jie TAN, Chelsea FINN, Mrinal KALAKRISHNAN, Peter PASTOR et Sergey LEVINE. « How to train your robot with deep reinforcement learning : lessons we have learned ». In : *The International Journal of Robotics Research* (2021) (cf. p. 18).
- [Pen18] Xue Bin PENG, Marcin ANDRYCHOWICZ, Wojciech ZAREMBA et Pieter ABBEEL. « Sim-to-real transfer of robotic control with dynamics randomization ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2018 (cf. p. 18).
- [Tan18] Jie TAN, Tingnan ZHANG, Erwin COUMANS, Atil ISCEN, Yunfei BAI, Danijar HAFNER, Steven BOHEZ et Vincent VANHOUCHE. « Sim-to-real : Learning agile locomotion for quadruped robots ». In : *arXiv preprint arXiv :1804.10332* (2018) (cf. p. 18).

- [Geh16] Christian GEHRING, Stelian COROS, Marco HUTTER, Carmine Dario BELLICOSO, Huub HEIJNEN, Remo DIETHELM, Michael BLOESCH, Péter FANKHAUSER, Jemin HWANGBO, Mark HOEPFLINGER et al. « Practice makes perfect : An optimization-based approach to controlling agile motions for a quadruped robot ». In : *IEEE Robotics & Automation Magazine* (2016) (cf. p. 18, 19).
- [Man18] Nicolas MANSARD, Andrea DELPRETE, Mathieu GEISERT, Steve TONNEAU et Olivier STASSE. « Using a memory of motion to efficiently warm-start a nonlinear predictive controller ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2018 (cf. p. 19, 62).
- [Pon14] Brahayam PONTÓN, Farbod FARSHIDIAN et Jonas BUCHLI. « Learning compliant locomotion on a quadruped robot ». In : *IROS Workshop (Ed.), Compliant manipulation : Challenges in learning and control*. Citeseer. 2014 (cf. p. 19).
- [Xie21] Zhaoming XIE, Xingye DA, Buck BABICH, Animesh GARG et Michiel van de PANNE. « Glide : Generalizable quadrupedal locomotion in diverse environments with a centroidal model ». In : *arXiv preprint arXiv :2104.09771* (2021) (cf. p. 19).
- [Tso20] Vassilios TSOUNIS, Mitja ALGE, Joonho LEE, Farbod FARSHIDIAN et Marco HUTTER. « DeepGait : Planning and Control of Quadrupedal Gaits Using Deep Reinforcement Learning ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2020) (cf. p. 19).
- [Gan21] Siddhant GANGAPURWALA, Mathieu GEISERT, Romeo ORSOLINO, Maurice FALLON et Ioannis HAVOUTIS. « Real-Time Trajectory Adaptation for Quadrupedal Locomotion using Deep Reinforcement Learning ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)* (2021) (cf. p. 19).
- [Sch17] John SCHULMAN, Filip WOLSKI, Prafulla DHARIWAL, Alec RADFORD et Oleg KLIMOV. « Proximal policy optimization algorithms ». In : *arXiv preprint arXiv :1707.06347* (2017) (cf. p. 19).
- [Fuj18] Scott FUJIMOTO, Herke HOOF et David MEGER. « Addressing function approximation error in actor-critic methods ». In : *International conference on machine learning*. PMLR. 2018 (cf. p. 19).
- [Haa18] Tuomas HAARNOJA, Aurick ZHOU, Pieter ABBEEL et Sergey LEVINE. « Soft actor-critic : Off-policy maximum entropy deep reinforcement learning with a stochastic actor ». In : *International conference on machine learning*. PMLR. 2018 (cf. p. 19).
- [Dav07] Andrew J DAVISON, Ian D REID, Nicholas D MOLTON et Olivier STASSE. « MonoSLAM : Real-time single camera SLAM ». In : *IEEE transactions on pattern analysis and machine intelligence* (2007) (cf. p. 19).
- [Doh17] Kevin DOHERTY, Jinkun WANG et Brendan ENGLLOT. « Bayesian generalized kernel inference for occupancy map prediction ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2017 (cf. p. 19).
- [Del99] Frank DELLAERT, Dieter FOX, Wolfram BURGARD et Sebastian THRUN. « Monte carlo localization for mobile robots ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 1999 (cf. p. 19).

- [Las21] Thibaud LASGUIGNES, Isabelle MAROGER, Maurice FALLON, Milad RAMEZANI, Luca MARCHIONNI, Olivier STASSE, Nicolas MANSARD et Bruno WATIER. « ICP Localization and Walking Experiments on a TALOS Humanoid Robot ». In : *International Conference on Advanced Robotics (ICAR)*. 2021 (cf. p. 19).
- [Zhe21] Kaiyu ZHENG. « Ros navigation tuning guide ». In : *Robot Operating System (ROS)*. Springer, 2021 (cf. p. 19).
- [Fer20] Pierre FERNBACH, Steve TONNEAU, Olivier STASSE, Justin CARPENTIER et Michel TAÏX. « C-croc : Continuous and convex resolution of centroidal dynamic trajectories for legged robots in multicontact scenarios ». In : *IEEE Transactions on Robotics (T-RO)* (2020) (cf. p. 20, 40).
- [Dei14] Robin DEITS et Russ TEDRAKE. « Footstep planning on uneven terrain with mixed-integer convex optimization ». In : *2014 IEEE-RAS international conference on humanoid robots*. 2014 (cf. p. 20, 40-42).
- [Pon16] Brahayam PONTON, Alexander HERZOG, Stefan SCHAAL et Ludovic RIGHETTI. « A convex model of humanoid momentum dynamics for multi-contact motion generation ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)*. 2016 (cf. p. 20).
- [AC17] Bernardo ACEITUNO-CABEZAS, Carlos MASTALLI, Hongkai DAI, Michele FOCCHI, Andreea RADULESCU, Darwin G CALDWELL, José CAPPELLETTO, Juan C GRIECO, Gerardo FERNÁNDEZ-LÓPEZ et Claudio SEMINI. « Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2017) (cf. p. 20).
- [Ton20] Steve TONNEAU, Daeun SONG, Pierre FERNBACH, Nicolas MANSARD, Michel TAÏX et Andrea DEL PRETE. « SL1M : Sparse L1-norm Minimization for contact planning on uneven terrain ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2020 (cf. p. 20, 40, 42, 44).
- [Esc13] Adrien ESCANDE, Abderrahmane KHEDDAR et Sylvain MIOSSEC. « Planning contact points for humanoid robots ». In : *Robotics and Autonomous Systems* (2013) (cf. p. 20, 40).
- [Che07] Joel CHESTNUTT, Koichi NISHIWAKI, James KUFFNER et Satoshi KAGAMI. « An adaptive action model for legged navigation planning ». In : *2007 7th IEEE-RAS International Conference on Humanoid Robots*. 2007 (cf. p. 20).
- [Lin18] Yu-Chi LIN et Dmitry BERENSON. « Humanoid navigation planning in large unstructured environments using traversability-based segmentation ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018 (cf. p. 20).
- [Gri19a] Robert J GRIFFIN, Georg WIEDEBACH, Stephen MCCRORY, Sylvain BERTRAND, Inho LEE et Jerry PRATT. « Footstep planning for autonomous walking over rough terrain ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)*. 2019 (cf. p. 20, 40).

- [Lin19] Yu-Chi LIN, Brahayam PONTON, Ludovic RIGHETTI et Dmitry BERENSON. « Efficient humanoid contact planning using learned centroidal dynamics prediction ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2019 (cf. p. 20).
- [Car16b] Justin CARPENTIER, Steve TONNEAU, Maximilien NAVEAU, Olivier STASSE et Nicolas MANSARD. « A versatile and efficient pattern generator for generalized legged locomotion ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2016 (cf. p. 20, 21).
- [Bou09] Karim BOUYARMANE, Adrien ESCANDE, Florent LAMIRAUX et Abderrahmane KHEDDAR. « Potential field guide for humanoid multicontacts acyclic motion planning ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2009 (cf. p. 20).
- [Ton18a] Steve TONNEAU, Andrea DEL PRETE, Julien PETTRÉ, Chonhyon PARK, Dinesh MANOCHA et Nicolas MANSARD. « An efficient acyclic contact planner for multi-ped robots ». In : *IEEE Transactions on Robotics (T-RO)* (2018) (cf. p. 20).
- [Son21] Daeun SONG, Pierre FERNBACH, Thomas FLAYOLS, Andrea DEL PRETE, Nicolas MANSARD, Steve TONNEAU et Young J KIM. « Solving footstep planning as a feasibility problem using l1-norm minimization ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2021) (cf. p. 20, 42).
- [Che21] Jason CHEMIN, Pierre FERNBACH, Daeun SONG, Guilhem SAUREL, Nicolas MANSARD et Steve TONNEAU. « Learning to steer a locomotion contact planner ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2021 (cf. p. 20).
- [Wel21] Lorenz WELLHAUSEN et Marco HUTTER. « Rough terrain navigation for legged robots using reachability planning and template learning ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021 (cf. p. 20).
- [Pet22] Matej PETRLIK, Pavel PETRACEK, Vit KRATKY, Tomas MUSIL, Yurii STASINCHUK, Matous VRBA, Tomas BACA, Daniel HERT, Martin PECKA, Tomas SVOBODA et al. « UAVs Beneath the Surface : Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT ». In : *arXiv preprint arXiv :2206.08185* (2022) (cf. p. 21).
- [Ciz16] Petr CIZEK et Jan FAIGL. « On localization and mapping with RGB-D sensor and hexapod walking robot in rough terrains ». In : *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2016 (cf. p. 21).
- [Tam21] Benjamin TAM, Thomas MOLNAR, Fletcher TALBOT, Brett WOOD et Ryan STEINDL. « Deploying COTS Legged Robot Platforms into a Heterogeneous Robot Team ». In : *arXiv preprint arXiv :2106.07182* (2021) (cf. p. 21).
- [Tra22] Marco TRANZATTO, Frank MASCARICH, Lukas BERNREITER, Carolina GODINHO, Marco CAMURRI, Shehryar KHATTAK, Tung DANG, Victor REIJGWART, Johannes LOEJE, David WISTH et al. « Cerberus : Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge ». In : *arXiv preprint arXiv :2201.07067* (2022) (cf. p. 21).

- [Mel21] Olivier MELON, Romeo ORSOLINO, David SUROVIK, Mathieu GEISERT, Ioannis HAVOUTIS et Maurice FALLON. « Receding-Horizon Perceptive Trajectory Optimization for Dynamic Legged Locomotion with Learned Initialization ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)* (2021) (cf. p. 21).
- [Win18] Alexander W WINKLER, C Dario BELLICOSO, Marco HUTTER et Jonas BUCHLI. « Gait and trajectory optimization for legged systems through phase-based end-effector parameterization ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2018) (cf. p. 21, 54, 72).
- [Gan22] Siddhant GANGAPURWALA, Mathieu GEISERT, Romeo ORSOLINO, Maurice FALLON et Ioannis HAVOUTIS. « Rloc : Terrain-aware legged locomotion using reinforcement learning and optimal control ». In : *IEEE Transactions on Robotics (T-RO)* (2022) (cf. p. 21).
- [Jen22] Fabian JENELTEN, Ruben GRANDIA, Farbod FARSHIDIAN et Marco HUTTER. « TAMOLS : Terrain-aware motion optimization for legged systems ». In : *IEEE Transactions on Robotics (T-RO)* (2022) (cf. p. 22).
- [Gra22] Ruben GRANDIA, Fabian JENELTEN, Shaohui YANG, Farbod FARSHIDIAN et Marco HUTTER. « Perceptive locomotion through nonlinear model predictive control ». In : *arXiv preprint arXiv :2208.08373* (2022) (cf. p. 22).
- [Rud22] Nikita RUDIN, David HOELLER, Marko BJELONIC et Marco HUTTER. « Advanced Skills by Learning Locomotion and Local Navigation End-to-End ». In : *arXiv preprint arXiv :2209.12827* (2022) (cf. p. 22).
- [Che13] Christine CHEVALLEREAU, Guy BESSONNET, Gabriel ABBA et Yannick Aoustin. *Bipedal robots : modeling, design and walking synthesis*. John Wiley & Sons, 2013 (cf. p. 24).
- [Bul] *Bullet physics library*. <https://pybullet.org/wordpress/> (cf. p. 28).
- [Ode] *Open dynamics engine*. <http://ode.org> (cf. p. 28).
- [Muj] *MuJoCo physics engine*. www.mujooco.org (cf. p. 28).
- [Häm15] Perttu HÄMÄLÄINEN, Joose RAJAMÄKI et C Karen LIU. « Online control of simulated humanoids using particle belief propagation ». In : *ACM Transactions on Graphics (TOG)* (2015) (cf. p. 28).
- [Raj16] Joose RAJAMÄKI, Kouros NADERI, Ville KYRKI et Perttu HÄMÄLÄINEN. « Sampled differential dynamic programming ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016 (cf. p. 28).
- [Tas12] Yuval TASSA, Tom EREZ et Emanuel TODOROV. « Synthesis and stabilization of complex behaviors through online trajectory optimization ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012 (cf. p. 28, 54).
- [Lid22] Quentin Le LIDEC, Louis MONTAUT, Cordelia SCHMID, Ivan LAPTEV et Justin CARPENTIER. « Augmenting differentiable physics with randomized smoothing ». In : *arXiv preprint arXiv :2206.11884* (2022) (cf. p. 28).

- [Tré05] Emmanuel TRÉLAT. *Contrôle optimal : théorie & applications*. Vuibert Paris, 2005 (cf. p. 29).
- [Pos14] Michael POSA, Cecilia CANTU et Russ TEDRAKE. « A direct method for trajectory optimization of rigid bodies through contact ». In : *The International Journal of Robotics Research* (2014) (cf. p. 30).
- [Bet10] John T BETTS. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010 (cf. p. 30).
- [Bir16] Francesco BIRAL, Enrico BERTOLAZZI et Paolo BOSETTI. « Notes on numerical methods for solving optimal control problems ». In : *IEEJ Journal of Industry Applications* (2016) (cf. p. 30).
- [Bel54] Richard BELLMAN. « The theory of dynamic programming ». In : *Bulletin of the American Mathematical Society* (1954) (cf. p. 31).
- [Jal22c] Wilson JALLET, Nicolas MANSARD et Justin CARPENTIER. « Implicit differential dynamic programming ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2022 (cf. p. 34).
- [Jal22a] Wilson JALLET, Antoine BAMBADE, Nicolas MANSARD et Justin CARPENTIER. « Constrained Differential Dynamic Programming : A primal-dual augmented Lagrangian approach ». In : (2022) (cf. p. 34).
- [Jal22b] Wilson JALLET, Antoine BAMBADE, Nicolas MANSARD et Justin CARPENTIER. « ProxNLP : a primal-dual augmented Lagrangian solver for nonlinear programming in Robotics and beyond ». In : *6th Legged Robots Workshop*. 2022 (cf. p. 34, 84).
- [Pas16] Grégoire PASSAULT, Quentin ROUXEL, François PETIT et Olivier LY. « Metabot : a low-cost legged robotics platform for education ». In : *International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE. 2016 (cf. p. 34).
- [Odr] *Open Dynamic Robot Initiative*. <https://open-dynamic-robot-initiative.github.io/> (cf. p. 35).
- [Ste20] Bartolomeo STELLATO, Goran BANJAC, Paul GOULART, Alberto BEMPORAD et Stephen BOYD. « OSQP : An operator splitting solver for quadratic programs ». In : *Mathematical Programming Computation* (2020) (cf. p. 38).
- [Car19b] Justin CARPENTIER, Guilhem SAUREL, Gabriele BUONDONNO, Joseph MIRABEL, Florent LAMIRAUX, Olivier STASSE et Nicolas MANSARD. « The Pinocchio C++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives ». In : *IEEE/SICE International Symposium on System Integration (SII)*. 2019 (cf. p. 38, 74, 91).
- [Ble18] Gerardo BLEDT, Matthew J POWELL, Benjamin KATZ, Jared DI CARLO, Patrick M WENSING et Sangbae KIM. « MIT Cheetah 3 : Design and control of a robust, dynamic quadruped robot ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018 (cf. p. 39).

- [Sie21] Jonah SIEKMANN, Kevin GREEN, John WARILA, Alan FERN et Jonathan HURST. « Blind bipedal stair traversal via sim-to-real reinforcement learning ». In : *arXiv preprint arXiv :2105.08328* (2021) (cf. p. 39).
- [Hau08] Kris HAUSER, Timothy BRETLE, Jean-Claude LATOMBE, Kensuke HARADA et Brian WILCOX. « Motion planning for legged robots on varied terrain ». In : *The International Journal of Robotics Research* (2008) (cf. p. 40).
- [Bre06] Timothy BRETLE. « Motion planning of multi-limbed robots subject to equilibrium constraints : The free-climbing robot problem ». In : *The International Journal of Robotics Research* (2006) (cf. p. 40).
- [Car18] Justin CARPENTIER et Nicolas MANSARD. « Multicontact locomotion of legged robots ». In : *IEEE Transactions on Robotics (T-RO)* (2018) (cf. p. 40, 54).
- [Ton18b] Steve TONNEAU, Pierre FERNBACH, Andrea Del PRETE, Julien PETTRÉ et Nicolas MANSARD. « 2pac : Two-point attractors for center of mass trajectories in multi-contact scenarios ». In : *ACM Transactions on Graphics (TOG)* (2018) (cf. p. 44).
- [Fla17] Thomas FLAYOLS, Andrea DEL PRETE, Patrick WENSING, Alexis MIFSUD, Mehdi BENALLEGUE et Olivier STASSE. « Experimental evaluation of simple estimators for humanoid robots ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)*. 2017 (cf. p. 47).
- [Car15b] Justin CARPENTIER, Mehdi BENALLEGUE, Nicolas MANSARD et Jean-Paul LAUMOND. « A kinematics-dynamics based estimator of the center of mass position for anthropomorphic system—A complementary filtering approach ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)*. 2015 (cf. p. 47).
- [Gei19] Mathieu GEISERT, Thomas YATES, Asil ORGEN, Pierre FERNBACH et Ioannis HAVOUTIS. « Contact planning for the anymal quadruped robot using an acyclic reachability-based planner ». In : *Annual Conference Towards Autonomous Robotic Systems*. Springer. 2019 (cf. p. 47).
- [Sol] *Real time footstep planning and control of the Solo quadruped robot in 3D environments*. https://gepettoweb.laas.fr/articles/risbourg_corberes_2022.html (cf. p. 48, 49).
- [Aft12] Zohaib AFTAB, Thomas ROBERT et Pierre Brice WIEBER. « Ankle, hip and stepping strategies for humanoid balance recovery with a single Model Predictive Control scheme ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)* (2012) (cf. p. 53).
- [Kry15] Przemyslaw KRYCZKA, Petar KORMUSHEV, Nikos G. TSAGARAKIS et Darwin G. CALDWELL. « Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015) (cf. p. 53, 54).
- [Kha16] Majid KHADIV, Alexander HERZOG, S. Ali A. MOOSAVIAN et Ludovic RIGHETTI. « Step timing adjustment : A step toward generating robust gaits ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)*. 2016 (cf. p. 53, 54).

- [Gri17] Robert J GRIFFIN, Georg WIEDEBACH, Sylvain BERTRAND, Alexander LEONESSA et Jerry PRATT. « Walking stabilization using step timing and location adjustment on the humanoid robot, Atlas ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017 (cf. p. 53).
- [Pha14] Quang Cuong PHAM. « A general, fast, and robust implementation of the time-optimal path parameterization algorithm ». In : *IEEE Transactions on Robotics (T-RO)* (2014) (cf. p. 54).
- [Car16a] Stéphane CARON et Quang-Cuong PHAM. « When to make a step? Tackling the timing problem in multi-contact locomotion by TOPP-MPC ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)* (2016) (cf. p. 54).
- [Far16] Farbod FARSHIDIAN, Michael NEUNERT, Alexander W. WINKLER, Gonzalo REY et Jonas BUCHLI. « An Efficient Optimal Planning and Control Framework For Quadrupedal Locomotion ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)* (2016) (cf. p. 54).
- [Pon18] Brahayam PONTON, Alexander HERZOG, Andrea DEL PRETE, Stefan SCHAAL et Ludovic RIGHETTI. « On Time Optimization of Centroidal Momentum Dynamics ». In : *IEEE/RAS International Conference on Robotics and Automation (ICRA)*. 2018 (cf. p. 54).
- [Neu17] Michael NEUNERT, Markus STÄUBLE, Markus GIFTTHALER, Carmine D. BELLICOSO, Jan CARIUS, Christian GEHRING, Marco HUTTER et Jonas BUCHLI. « Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2017) (cf. p. 54).
- [Wäc06] Andreas WÄCHTER et Lorenz T BIEGLER. « On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming ». In : *Mathematical programming* (2006) (cf. p. 59).
- [Wri97] Stephen J WRIGHT. *Primal-dual interior-point methods*. SIAM, 1997 (cf. p. 59).
- [Mel20] Oliwier MELON, Mathieu GEISERT, David SUROVIK, Ioannis HAVOUTIS et Maurice FALLON. « Reliable Trajectories for Dynamic Quadrupeds using Analytical Costs and Learned Initializations ». In : 2020 (cf. p. 72).
- [Wen17] Patrick WENSING, Albert WANG, Sangok SEOK, David OTTEN, Jeffrey LANG et Sangbae KIM. « Proprioceptive Actuator Design in the MIT Cheetah : Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots ». In : *IEEE Transactions on Robotics (T-RO)* (2017) (cf. p. 72).
- [Fea83] Roy FEATHERSTONE. « The calculation of robot dynamics using articulated-body inertias ». In : *The international journal of robotics research* (1983) (cf. p. 74).
- [Kro70] Fred T. KROGH. « Efficient Algorithms for Polynomial Interpolation and Numerical Differentiation ». In : *Mathematics of Computation* (1970) (cf. p. 74).
- [Wbc] *Whole-Body MPC without Foot References for the Locomotion of an Impedance-Controlled Robot*. https://gepettoweb.laas.fr/articles/assirelli_2022.html (cf. p. 78, 80).

- [Ara22] Michel ARACTINGI, Pierre-Alexandre LÉZIART, Thomas FLAYOLS, Julien PEREZ, Tomi SILANDER et Philippe SOUÈRES. « Controlling the Solo12 Quadruped Robot with Deep Reinforcement Learning ». In : (2022) (cf. p. 80).
- [Mas22] Carlos MASTALLI, Saroj Prasad CHHATOI, Thomas CORBÈRES, Steve TONNEAU et Sethu VIJAYAKUMAR. « Inverse-Dynamics MPC via Nullspace Resolution ». In : *arXiv preprint arXiv :2209.05375* (2022) (cf. p. 84).
- [Mar22] Giorgos MARINOÛ, Lizeth SLOOT et Katja MOMBAUR. « Towards efficient lower-limb exoskeleton evaluation : defining biomechanical metrics to quantify assisted gait familiarization ». In : *IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. 2022 (cf. p. 85).
- [Vig18] Matthieu VIGNE, Antonio EL KHOURY, Matthieu MASSELIN, Florent DI MEGLIO et Nicolas PETIT. « Estimation of multiple flexibilities of an articulated system using inertial measurements ». In : *IEEE Conference on Decision and Control (CDC)*. 2018 (cf. p. 86, 90).
- [Her18] Ayonga HEREID, Christian M HUBICKI, Eric A COUSINEAU et Aaron D AMES. « Dynamic humanoid locomotion : A scalable formulation for HZD gait optimization ». In : *IEEE Transactions on Robotics (T-RO)* (2018) (cf. p. 87).
- [Pra06a] Jerry PRATT, John CARFF, Sergey DRAKUNOV et Ambarish GOSWAMI. « Capture point : A step toward humanoid push recovery ». In : *IEEE/RAS International Conference on Humanoid Robotics (Humanoids)*. 2006 (cf. p. 88).
- [Kaj01] Shuuji KAJITA, Fumio KANEHIRO, Kenji KANEKO, Kazuhito YOKOI et Hirohisa HIRUKAWA. « The 3D linear inverted pendulum mode : A simple modeling for a biped walking pattern generation ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2001 (cf. p. 88).
- [Vig19] Matthieu VIGNE, Antonio EL KHOURY, Florent DI MEGLIO et Nicolas PETIT. « State Estimation for a Legged Robot With Multiple Flexibilities Using IMUs : A Kinematic Approach ». In : *IEEE/RAS Robotics and Automation Letters (RA-L)* (2019) (cf. p. 90).
- [Vig20] Matthieu VIGNE, Antonio EL KHOURY, Florent DI MEGLIO et Nicolas PETIT. « Improving low-level control of the exoskeleton Atalante in single support by compensating joint flexibility ». In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020 (cf. p. 90, 91).
- [Jim] *Jiminy*. <https://duburcqa.github.io/jiminy/> (cf. p. 91).
- [Vid] *Vidéo d'expériences avec l'exosquelette Atalante*. <https://peertube.laas.fr/w/irCFRKSz8YtVD9N3cA9zYP> (cf. p. 93).

Résumé —

La commande prédictive permet de choisir un contrôle optimal sur un horizon glissant pour la locomotion des robots à pattes. Le choix du modèle et du problème de contrôle optimal est crucial pour que la résolution numérique soit possible dans les temps impartis. Cette thèse apporte des contributions pour améliorer la locomotion de quadrupèdes sur des terrains 3D. Elle étudie comment l'optimisation de certains paramètres de la marche peut améliorer la robustesse du contrôleur. Une première partie utilise un modèle réduit de la dynamique en complément de planification réactive des contacts pour permettre la navigation dans un terrain semé d'obstacles. Une seconde partie étudie comment la durée des pas peut-être optimisée dans un schéma de contrôle avec commande prédictive sur modèle réduit. Enfin une dernière partie s'intéresse à la commande prédictive avec modèle complet de la dynamique pour optimiser sans guide la trajectoire des pieds de vol. L'accent est mis sur la validation expérimentale de ces différentes contributions sur un robot quadrupède.

Mots clés : Robot, Commande, Equilibre, Marche, Predictif, Réactif

Contributions to the predictive control for the locomotion of quadruped robots over 3D terrains

Abstract —

Predictive control is used to choose an optimal control over a sliding horizon for the locomotion of legged robots. The choice of the model and optimal control problem is crucial to find a numerical solution in the given time. This thesis contributes to the improvement of the locomotion of quadrupeds over 3D terrains. It studies how the optimization of some parameters of the walk can improve the robustness of the controller. A first part uses a reduced dynamics model as a complement to reactive contact planning to enable navigation in an obstacle-ridden terrain. A second part studies how the step duration can be optimized in a control scheme with predictive control on a reduced model. Finally, the last part focuses on predictive control with a complete model of the dynamics to optimize the trajectory of the flight foot without a guide. The emphasis is put on the experimental validation of these different contributions on a quadruped robot.

Keywords : Robot, Control, Balance, Walking, Predictive, Reactive

LAAS-CNRS
7 Av. du Colonel Roche,
31400 Toulouse