Université de Montpellier 2

Rapport de Master 1
Mention : " STPI Electrotechnique Electronique et Automatique "
Spécialité : " Robotique et Automatique "

Préparé au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

présenté et soutenu par

Nicolas CARLESI

le 09 Juillet 2008

Commande pour la stabilisation de mouvements périodiques du robot "Kangourou" : un robot sauteur planaire unijambiste

---

Encadrant : Ahmed CHEMORI

Département : Robotique **Equipe :** DEXTER

Laboratoire d'**I**nformatique de **R**obotique et de **M**icroélectronique de **M**ontpellier
Unité mixte de recherche du **CNRS** - UMR 5506
et de l'**U**niversité de **M**ontpellier **II**

Résumé

La commande des robots sauteurs unijambistes dynamiquements stables a été l'objet de progrès considérable au cours des vingt dernières années. Ce rapport traite de la modélisation et de la commande de ce type de robot. Le robot sauteur en question s'appelle **"Kangourou"**. C'est un robot sauteur planaire unijambiste qui s'inspire du mode de déplacement des kangourous. Un simulateur a été développé et permet de simuler le comportement du robot commandé. Deux approches de commande sont étudiées pour le contrôle du saut vertical. Pour le déplacement planaire, deux approches de commandes ont été proposées, à savoir la commande de Raibert et la commande prédictive non linéaire. La commande de Raibert a été combinée avec la commande prédictive non linéaire, ce qui nous a permis d'améliorer ses performances. Les différentes approches de commandes proposées ont été validées en simulation sur le simulateur du robot développé.

Abstract

The control of dynamically stable legged robots has made great progress in the last twenty years. This report deals with modeling and control of this type of robots. The modeled hopping robot is called **"Kangourou"**. It is a planar hopping robot which imitates the mode of displacement of kangaroos. A simulator is developed and enables to simulate the behavior of the controlled robot. Two control schemes have been proposed to control the vertical hopping of the robot. For planar hopping, two control approaches have been proposed, that is Raibert's control approach and a nonlinear predictive control approach. Raibert's approach has been combined with nonlinear predictive control which has improved significantly its performances. The proposed control approaches have been validated on the robot's simulator.

Remerciements

Avant de présenter mes travaux de recherche, je tenais à remercier tout ceux qui m'ont accueillit au sein du Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier. Ces mois passées au laboratoire ont été une source d'enrichissement considérable.

J'adresse des remerciements particuliers à Monsieur Ahmed Chemori, chercheur CNRS au LIRMM, pour m'avoir confié ce sujet. Qu'il soit assuré de ma gratitude pour sa disponibilité, sa rigueur scientifique, son ouverture ainsi que pour son aide qu'il m'a apporté. Ce fut un réel plaisir de passer ces quelques mois à ses côtés.

Je remercie vivement les stagiaires de $2^{\text{ème}}$ année de master de robotique/automatique pour leur accueil chaleureux. Leurs expériences, leurs conseils avisés et leurs aides pour la rédaction de mon rapport ont été très appréciés.

Bien sûr je n'oublie pas mes collègues de $1^{\text{ère}}$ année pour leurs encouragements ainsi que pour les quelques moments (très peu) de détente à leurs côtés.

Table des matières

Chapitre 1

Introduction générale

La commande des robots sauteurs a attiré ces dernières années de nombreux chercheurs de diverses communautés (conception mécanique, robotique, automatique, ...) qui s'intéressent à plusieurs aspects tels que la conception, la modélisation, l'analyse et la commande.

Le travail réalisé durant ce stage rentre dans le cadre du projet national SHERPA. C'est un projet ANR sur la locomotion dont le but est de concevoir, réaliser et commander un robot marcheur bipède.

1.1 Présentation du LIRMM

Le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) est une Unité Mixte de Recherche de l'Université Montpellier II (UMII) et du Centre National de la Recherche Scientifique (CNRS).

Le LIRMM couvre un large spectre de compétences dans les domaines des **S**ciences et **T**echnologies de l'**I**nformation, de la **C**ommunication et des **S**ystèmes (**STICS**). Ses activités sont réparties au sein de trois départements scientifiques de recherche :

- Informatique (INFO);
- Microélectronique (MIC);
- Robotique (ROB).

Le LIRMM regroupe 292 personnes (dont 152 permanents) :

- 95 enseignants-chercheurs et 29 chercheurs CNRS (et également INRIA,...)
- 28 ingénieurs, techniciens ou administratifs (+6 contractuels)

- 134 doctorants
- 3 chercheurs contractuels (hors doctorants).

En moyenne, la production scientifique annuelle du LIRMM est de 350 publications, dont 25 thèses de doctorat et 170 publications dans des revues ou des congrès d'audience internationale.

1.2 Intérêt des robots sauteurs

Les scientifiques s'attendent à résoudre bien des problèmes de notre civilisation, en perçant le secret du mouvement des êtres vivants, car avec un minimum de matière et d'énergie, les animaux et les plantes obtiennent un effet maximum. La bionique est la sience qui étudie les systèmes biologiques avec l'objectif de comprendre les mécanismes de fonctionnement des organismes vivants et évolutifs afin de pouvoir les appliquer aux créations humaines. La bionique est devenue incontournable en différents domaines, tel que:

- l'intelligence artificielle
- la robotique (les robots marcheurs, les robots sauteurs, les robots nageurs à queue de poisson et les robots "insectes")
- les revêtements autonettoyants basés sur les études sur les feuilles de lotus
- l'aéronautique (les engins volants, les avions à ailes battantes, les micro drones)

Là où l'Homme est contraint de modifier la nature pour exploiter ses véhicules à roues (vélos, voitures, motos,...), les êtres vivants ont su évoluer pour s'adapter à leur environnement. Constamment en fuite ou en chasse, les animaux, ont développés des mouvements complexes qui leurs permettent de se déplacer avec une redoutable efficacité dans des milieux de diverses natures. Ils nagent, volent, rampent, marchent ou encore se déplacent en sautant comme les kangourous (cf. figure 1.1).

FIG. 1.1 – *Les êtres vivants et leurs modes de déplacement*

Le saut du kangourou [1] est particulièrement bien adapté aux terrains accidentés pourvu que ceux-ci soient suffisamment dégagés pour qu'il puisse exercé un mouvement vertical. Les kangourous sont capables de réaliser de véritables prouesses physiques: ils peuvent faire des sauts de plus de 13 m de long et peuvent sauter à plus de 3 m de haut! Lorsqu'ils fuient, leur vitesse de déplacement peut atteindre les 80 km/h! Mais quel est le secret de ce surprenant mammifère? Qu'en est-il du bilan énergétique de ce moyen de locomotion extrêmement performant?

Le saut oblige le corps à reprendre de la vitesse après chaque contact avec le sol. En apparence la nature a fait un calcul énergétique complètement erroné. Mais qu'en est-il réellement? L'énergie d'un saut, au lieu de se perdre, est en fait réutilisée dans le saut suivant. Les puissants tendons d'Achille des pattes arrières fournissent la solution à cette énigme. Ils fonctionnent comme des ressorts, qui emmagasinent à l'atterrissage l'énergie gagnée durant le mouvement balistique, pour la libérer dans le saut suivant. Après étude sur la consommation d'oxygène des kangourous, il s'est avéré que le saut est l'un des modes de locomotion les plus économique du règne animal. Les tendons d'Achille récupèrent jusqu'à 70% de l'énergie d'un saut. Le saut occasionne une dépense d'énergie inférieure de 40% à celle de la course.

Pour toutes ces raisons, il parait intéressant d'étudier ce moyen de locomotion et peut-être tenter de l'imiter. C'est l'objectif du robot sauteur étudié dans le cadre de ce stage, il s'appellera **'Kangourou'**.

1.3 Historique des réalisations

La recherche dans le domaine des robots à pattes a pour objectif d'acquérir des connaissances sur la modélisation, la conception, la commande et la mise au point de machines capables d'accomplir un déplacement avec le maximum de mobilité, de dextérité,de vitesse et d'efficacité. Ce moyen de locomotion implique des robots généralements sous-actionnés à plusieurs degrés de liberté et aux modèles complexes à cause de leurs fortes non linéarités, leurs structures variables et leurs caractères hybrides. Il fallait également surmonter les défits technologiques liés aux actionneurs.

Déjà à la fin du XIXème siècle, des études scientifiques furent menées, principalement aux États-Unis pour trouver des alternatives aux déplacements à cheval (cf. figure 1.2) et aux véhicules à roues. De nombreuses conceptions virent le jour jusqu'au milieu des années 60 mais la performance de ces machines était limité à cause des mouvements de leurs pattes fixes. Elles ne pouvaient pas prendre en compte les perturbations et ne pouvait pas s'adapter à la variation de terrain.

Au milieu des année 60, un robot quadrupède fut développé par Ralph Mosher pour General Electric (cf. figure 1.3). Cette machine de près de 3.40 m de haut et d'un poids de 1.362 tonnes

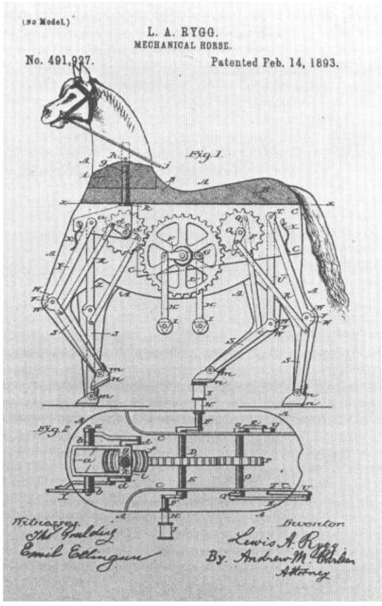

FIG. 1.2 – *Cheval mécanique imaginé par Lewis A. Rygg in 1873, apparemment cette machine n'a jamais été construite.*

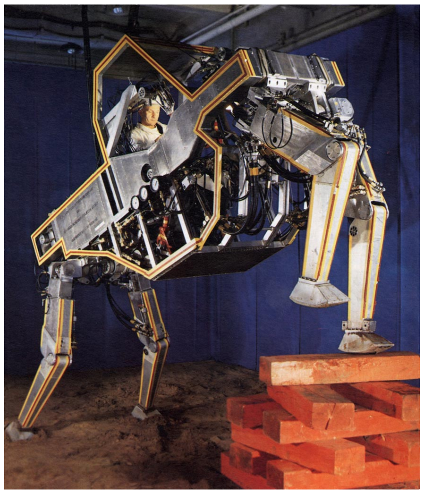

FIG. 1.3 – *Robot quadrupède développé par Ralph Mosher pour General Electric en 1968. L'homme contrôle la machine avec quatre manettes et pédales qui sont connectées à quatre pattes actionnées hydroliquements.*

était entièrement commandé par un homme. Une alternative au contrôle humain fut inventé par l'américain Robert McGhee. Il construit en 1977 un robot insecte à 6 pattes capable de marcher normalement, de tourner, de se déplacer en crabe et de négocier des obstacles simples. Déjà, un calculateur permettait de résoudre les équations cinématiques du système pour coordonner les 18 moteurs électriques des jambes. Des robots aux approches de commandes similaires furent élaborés en même temps en URSS par Gurfinkel et au Japon par S. Hirose. Ces robots adoptaient souvent une marche "statique". Ils gardaient la projection de leur centre de masse à l'intérieur du polygone de substentation formé par les points de contact avec le sol.

Au milieu des années 80, Marc Raibert [10], professeur au Massachusetts Institute of Technologie et créateur Leg Lab, a étudié et conçu les premiers robots unijambistes, bipèdes et quadrupèdes dynamiquements stables (cf. figures 1.4, 1.5 et 1.6).

Le premier robot sauteur réalisé (cf. figures 1.4) était un robot sauteur planaire unijambiste. Constitué d'un actionneur hydraulique pour transmettre la poussé de la jambe, ce robot était tenu latéralement limitant ses mouvements à un déplacement planaire. Ensuite, très rapidement Raibert implémenta sa commande à un robot possédant trois degrés de liberté dans l'espace (cf. figure 1.7).

Les recherches menées par la suite, par de nombreux scientifiques, avaient pour but d'amé-

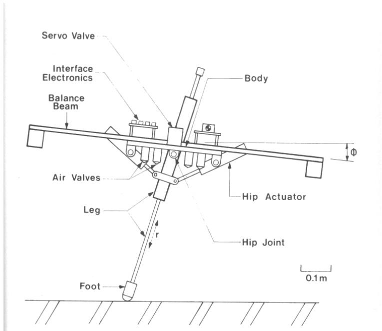

FIG. 1.4 – *Premier robot sauteur unijambiste planaire par M. Raibert (1980-1982)*

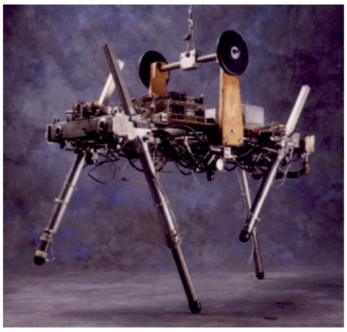

FIG. 1.5 – *Robot quadrupède réalisé par M. Raibert (1984-1987)*

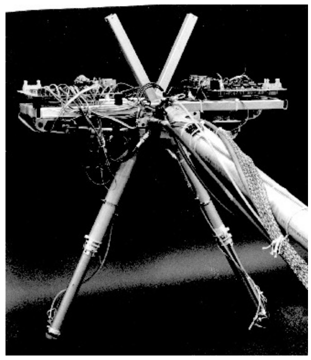

FIG. 1.6 – *Robot bipède planaire réalisé par M. Raibert (1985-1990)*

liorer la commande de Raibert, permettant de mieux suivre les consignes sur la hauteur du saut (mouvement balistique) ou sur la vitesse de déplacement en économisant le maximum d'énergie. D'autres conceptions de robots plus évolués (cf. figure 1.8), actionnés électriquements, virent le jour et d'autres objectifs sont venus raviver l'imagination des scientifiques. En mai 2008, un robot insecte sauteur, s'inspirant de la morphologie de la sauterelle (cf. figure 1.9), fut réalisé par une équipe de l'université de Lausanne [3]. Ce micro-robot de 7g est capable de faire des sauts de 1.4 m de haut, ce qui correspond à 27 fois sa taille. Ce robot ne permet pas encore d'atterrir correctement, pour effectuer le saut suivant, mais cela ne devrait pas tarder!

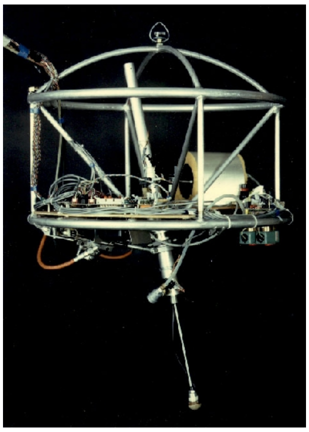

FIG. 1.7 – *Premier robot sauteur unijambiste à 3 degrés de liberté par M. Raibert (1983-1984)*

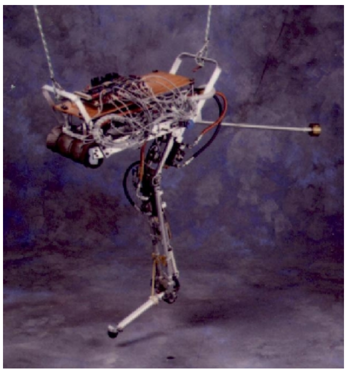

FIG. 1.8 – *Robot sauteur unijambiste planaire "Uniroo", premier robot se rapprochant de la morphologie du kangourou (1991-1993)*

FIG. 1.9 – *Micro-robot sauterelle, il peut sauter 27 fois sa taille, Universté de Lausanne (mai 2008)*

1.4 Contexte et problématiques

La recherche en France, dans le domaine des robots à pattes, s'est beaucoup concentrée sur les robots bipèdes. Il n'y a pas d'études, n'y de réalisations de robots sauteurs unijambistes. Les recherches dans ce domaine, comme nous l'avons vu dans la section précédente 1.3, se sont effectués principalement aux États-Unis, au Canada, au Japon, en Suisse et en Russie.

Dans ce contexte, l'objectif de mon stage est de modéliser, simuler et proposer des lois de commande pour asservir un robot sauteur unijambiste planaire. Les approches de commande proposées seront validées sur un simulateur du robot en question développé sous Matlab.

Les principales difficultés se situent au niveau de la modélisation pour laquelle il faudra prendre en compte suffisamment de paramètres pour s'approcher au maximum du comportement réel du robot. Ensuite il faudra, en premier lieu, l'implementer sous Matlab dans un simulateur qui modélisera au mieux le comportement du robot. Puis il faudra proposer les lois de commandes pour contrôler le robot en mouvement.

Dans cette phase de commande, plusieurs difficultés et complexités sont à prévoir, telle que la dynamique complexe et non linéaire du modèle du robot, le sous-actionnement, la structure variable et le caractère hybride sur le cycle complet de saut.

Les robots sauteurs sont des systèmes sous-actionnés [14], ce qui signifie que le nombre d'actionneurs est inférieur au nombre de degrés de liberté du robot. Des exemples de ces systèmes sont représentés sur les figures 1.10,1.11 et 1.12.

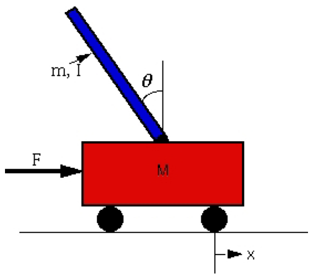

FIG. 1.10 – *Pendule inversé: on commande la position du chariot pour suivre une position du chariot tout en maintenant la barre autour de la verticale*

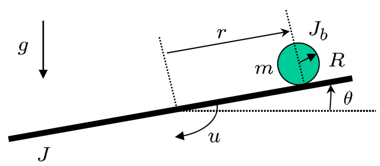

FIG. 1.11 – *Bille sur rail: on commande l'angle de la barre pour ramener la bille à une position désirée*

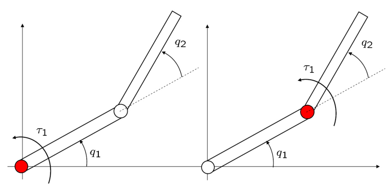

FIG. 1.12 – *Pendubot et acrobot: on commande une articulation pour suivre des angles sur les 2 articulations*

Les robots sauteurs sont des systèmes hybrides à structure variables. En effet, le saut est un mode de déplacement qui implique une modélisation pour les différentes phases du cycle de saut (cf. figure 1.13). Le nombre de degrés de liberté du robot peut varier d'une phase à l'autre (structure variable).

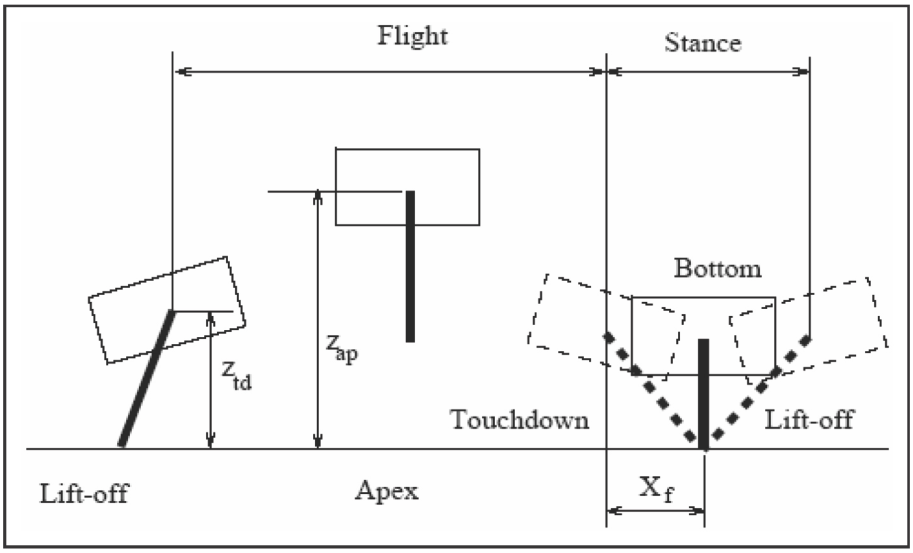

FIG. 1.13 – *Description des différentes phases sur un cycle de saut*

Le cycle de saut peut être décomposé en 2 phases et 2 transitions. Suivons la chronologie de la figure 1.13:

Le décollage (*Lift-off*) **:** transition qui caractérise le passage entre le moment où le robot est en contact avec le sol et celui où il est en l'air. A cet instant, l'impulsion donnée par la jambe pendant la phase de contact donne suffisamment d'énergie au robot pour passer en phase de vol.

La phase de vol (*Flight phase*) **:** phase principale dans laquelle le robot est en l'air et suit un mouvement balistique. Le robot monte dans un premier temps pour redescendre par la suite. L'instant où le robot atteind la hauteur maximale est appelé *Apex*.

L'atterrissage (*Touchdown*) **:** transition qui caractérise le passage entre la phase de vol et la phase de contact.

La phase de contact (*Stance phase*) **:** phase principale dans laquelle le robot est en contact avec le sol et suit un mouvement de balancement. L'instant marqué par une position de la jambe verticale est appelé *Bottom*. Ensuite, le cycle se répète et le robot redécolle.

L'objectif du stage est de générer des mouvements périodiques stables pour converger vers un cycle limite stable. Les consignes données par l'utilisateur, comme la vitesse de déplacement ou la hauteur des sauts, devrons être suivit du mieux que possible.

Chapitre 2

Le robot Kangourou : description et modélisation

Dans ce chapitre, nous présenterons les caractéristiques du robot que l'on va modéliser dans les différentes phases du cycle de saut décrites sur la figure 1.13.

2.1 Description du Robot Kangourou

Le robot sauteur en question est un robot planaire unijambiste. Sa modélisation se fera donc en 2 dimensions. De structure simple comme le montre la figure 2.1, il comporte 5 degrés de liberté:

x,z : 2 degrés pour la position du corps dans le plan XOZ

l : 1 degré pour la longueur de la jambe

ϕ : 1 degré pour l'orientation du corps dans l'espace

υ : 1 degré pour l'orientation de la jambe dans l'espace

Le vecteur de coordonnées généralisées permettant de décrire entièrement le système sera donc : $q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 \end{bmatrix}^T = \begin{bmatrix} x & z & l & \phi & \upsilon \end{bmatrix}^T$.
Les vecteurs de vitesses et des accélérations sont obtenus en dérivant successivement le vecteur de coordonnées généralisées: $\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{l} \\ \dot{\phi} \\ \dot{\upsilon} \end{bmatrix}$; $\ddot{q} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \\ \ddot{q}_4 \\ \ddot{q}_5 \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ \ddot{z} \\ \ddot{l} \\ \ddot{\phi} \\ \ddot{\upsilon} \end{bmatrix}$.

Nous allons décrire plus précisément les éléments pris en compte dans la modélisation. Le système comporte trois masses, celles-ci seront représentées par trois masses ponctuelles localisées à leurs centres de gravité respectifs: m_t pour le corps du robot, m_l pour la jambe et m_u

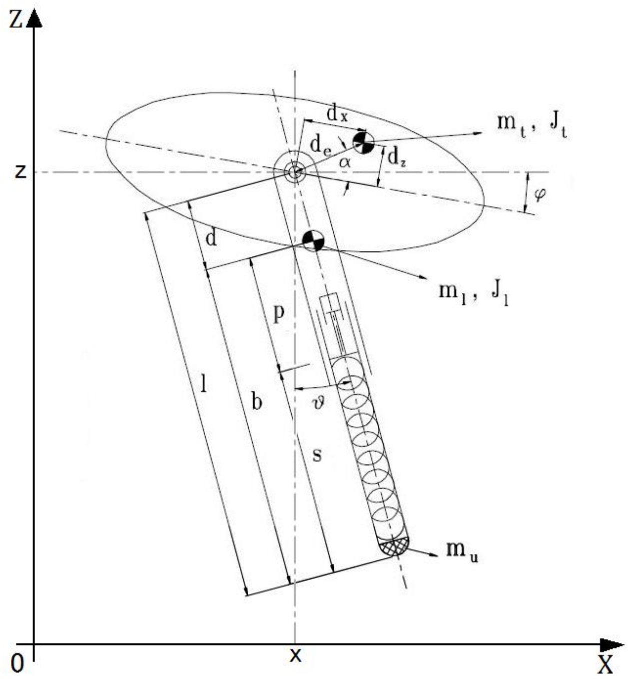

FIG. 2.1 – *Caractéristiques du robot Kangourou*

pour le pied. Le centre de gravité du corps m_t est positionné à une distance d_e de la hanche et forme un angle α avec l'axe longitudinal du corps du robot (cf. figure 2.1). Celui de la jambe m_l se situe, quand à lui, à une distance d de la hanche du robot. Une des extrémités du ressort est fixée à la partie supérieure de la jambe à une distance p du centre de gravité de la jambe. L'autre extrémité est fixée au pied du robot, le tube renfermant le ressort coulisse ainsi dans la partie supérieure de la jambe. Le ressort est caractérisé par une constante de raideur k_l et par sa longueur s, la longueur s_0 symbolise la longueur du ressort au repos. Enfin, la longueur l représente la longueur totale de jambe (l_0 au repos) et définie la distance entre le centre de gravité du pied m_u et la hanche. Les moments d'inertie du corps et de la jambe sont donnés respectivement par J_t et J_l.
Remarque : les différents paramètres géométriques et dynamiques sont illustrés sur la figure 2.1.

Comme le montre la figure 2.2, le robot comporte 2 actionneurs :

- **un actionneur angulaire** au niveau de la hanche qui agit sur l'orientation du corps par rapport à la jambe
- **un actionneur linéaire** au niveau de la jambe constitué d'un vérin électrique compliant qui agit sur l'allongement et la contraction de la jambe

L'actionneur linéaire au niveau de la jambe du robot est associé à un ressort monté en parallèle qui récupère l'énergie du saut à l'image du tendon d'Achille du kangourou. L'actionneur exerce une force entre les parties supérieure et inférieure de la jambe.

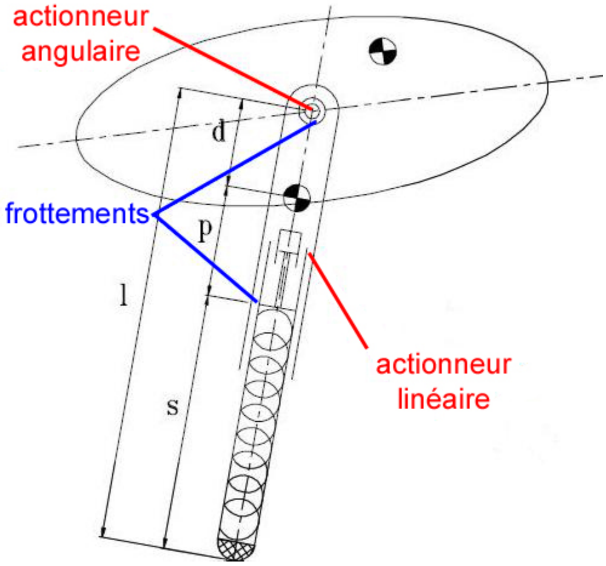

FIG. 2.2 – *Représentation des actionneurs et des frottements*

Les frottements c_l et c_h modélisés se situent respectivements au niveau de la hanche, lorsque le corps se déplace par rapport à la jambe, et au niveau de la jambe à l'endroit où coulisse la partie inférieure de la jambe dans la partie supérieure (cf. figure 2.2) .

2.2 Modélisation dynamique sur un cycle de saut

Maintenant que nous avons décrit les caractéristiques du robot, il va falloir modéliser notre système. La modélisation dynamique permettra de connaître l'évolution du système dans le temps. Pour cela, on doit calculer le modèle dynamique Lagrangien pour chacune des deux phases principales du cycle de saut, à savoir la phase de vol et la phase de contact (appelée aussi phase de support).

2.2.1 Phase de vol

Durant la phase de vol, le robot est libre et ne subit aucune contrainte. Le calcul du modèle dynamique Lagrangien nécessite plusieurs étapes. Tout d'abord, il nous faut calculer les énergies cinétiques et potentielles du système qui nous permettrons de déterminer le Lagrangien $\mathbf{L} = \mathbf{T} - \mathbf{V}$ (T est l'énergie cinétique totale et V est l'énergie potentielle totale du système).

Ensuite, il faut calculer les équations dynamiques qui lient les positions, les vitesses et les accélérations du système:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} + \frac{\partial F}{\partial q_i} = Q_i \tag{2.1}$$

où Q_i $(i = 1...N)$ sont les forces ou les couples généralisés correspondant aux coordonnées généralisées q_i, F est l'énergie due aux frottements.

Enfin, il sera nécessaire de mettre sous forme matricielle ces équations dynamiques, ce qui facilitera l'implémentation du modèle sous Matlab. Le modèle Lagrangien sous forme matricielle s'écrit [15] :

$$M(q).\ddot{q} + N(q,\dot{q}).\dot{q} + F(q) + G(q) = U \tag{2.2}$$

où M est la matrice d'inertie, N est la matrice de Coriolis, F est le vecteur de frottement, G le vecteur de Gravité et U le vecteur de commande. Les vecteurs de positions, vitesses et accélérations sont notés respectivements q,$\dot{q}$ et $\ddot{q}$.

La matrice d'inertie est déduite de l'énergie cinétique que l'on exprime sous la forme:

$$T = \frac{1}{2}\sum_{i,j=1}^{n} M_{ij}(q)\dot{q}_i\dot{q}_j = \frac{1}{2}\dot{q}^T M(q)\dot{q} \tag{2.3}$$

Remarque : la matrice d'inetie M(q) est symétrique définie positive de dimension nxn.

La matrice de Coriolis et des termes centrifuges se déduit de la matrice d'inertie en appliquant la formule suivante [15]:

$$N_{kj} = \sum_{i=1}^{n} N_{ijk}(q)\dot{q}_i = \sum_{i=1}^{n}\frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i \tag{2.4}$$

Le vecteur de frottement se déduit de l'énergie de frottements globale du robot à travers la relation suivante:

$$F(q,\dot{q}) = \begin{bmatrix} \frac{\partial F}{\partial \dot{q}_1} \\ \vdots \\ \frac{\partial F}{\partial \dot{q}_i} \end{bmatrix} \tag{2.5}$$

Le vecteur de gravité ne dépend que des coordonnées généralisées q, il se déduit de l'énergie potentielle globale du robot:

$$G(q) = \begin{bmatrix} \frac{\partial V}{\partial \dot{q}_1} \\ \vdots \\ \frac{\partial V}{\partial \dot{q}_i} \end{bmatrix} \tag{2.6}$$

Calcul des équations dynamiques

Les positions cartésiennes des trois masses des corps du robot (cf. figure 2.1) dans le plan XOZ sont donnée par:

$$\begin{cases} x_t = x + d_e \cos(\alpha + \phi) \\ z_t = z + d_e \sin(\alpha + \phi) \end{cases}$$
$$\begin{cases} x_l = x + d \sin(\upsilon) \\ z_l = z - d \cos(\upsilon) \end{cases} \tag{2.7}$$
$$\begin{cases} x_u = x + l_0 \sin(\upsilon) \\ z_u = z - l_0 \cos(\upsilon) \end{cases}$$

La vitesse de ces masses sont données par:

$$\begin{aligned} v_t &= \dot{x}_t \overrightarrow{x} + \dot{z}_t \overrightarrow{z} = (\dot{x} - d_e \sin(\alpha + \phi)\dot{\phi})\overrightarrow{x} + (\dot{z} + d_e \cos(\alpha + \phi)\dot{\phi})\overrightarrow{z} \\ v_l &= \dot{x}_l \overrightarrow{x} + \dot{z}_l \overrightarrow{z} = (\dot{x} - d \cos(\upsilon)\dot{\upsilon})\overrightarrow{x} + (\dot{z} - d \sin(\upsilon)\dot{\upsilon})\overrightarrow{z} \\ v_u &= \dot{x}_u \overrightarrow{x} + \dot{z}_u \overrightarrow{z} = (\dot{x} + \sin(\upsilon)\dot{l} + l \cos(\upsilon)\dot{\upsilon})\overrightarrow{x} + (\dot{z} - \cos(\upsilon)\dot{l} + l \sin(\upsilon)\dot{\upsilon})\overrightarrow{z} \end{aligned} \tag{2.8}$$

On en déduit l'énergie cinétique totale du robot:

$$\begin{aligned} T =& v_t + v_l + v_u \\ T =& \frac{1}{2}m_t v_t^2 + \frac{1}{2}J_t \phi^2 + \frac{1}{2}m_l v_l^2 + \frac{1}{2}J_l \upsilon^2 + \frac{1}{2}m_u v_u^2 \end{aligned}$$

$$\begin{aligned} T =& \frac{1}{2}m_t\{(\dot{x} - d_e \sin(\alpha + \phi)\dot{\phi})^2 + (\dot{z} + d_e \cos(\alpha + \phi)\dot{\phi})^2\} \\ &+ \frac{1}{2}m_l\{(\dot{x} - d \cos(\upsilon)\dot{\upsilon})^2 + (\dot{z} - d \sin(\upsilon)\dot{\upsilon})^2\} \\ &+ \frac{1}{2}m_u\{(\dot{x} + \sin(\upsilon)\dot{l} + l \cos(\upsilon)\dot{\upsilon})^2 + (\dot{z} - \cos(\upsilon)\dot{l} + l \sin(\upsilon)\dot{\upsilon})^2\} \\ &+ \frac{1}{2}(J_t \phi^2 + J_l \upsilon^2) \end{aligned} \tag{2.9}$$

L'énergie potentielle totale, quant à elle, est donnée par:

$$V = m_t g(z + d_e \sin(\alpha + \phi)) + m_l g(z - d \cos(\upsilon)) + m_u g(z - l \cos(\upsilon)) \tag{2.10}$$

L'énergie due aux frottements vaut:

$$F = f_h + f_l = \tfrac{1}{2}c_h(\dot{\phi} - \dot{\upsilon})^2 + \tfrac{1}{2}c_l(\dot{p} - \dot{l})^2 \tag{2.11}$$

où f_h et f_l sont respectivement l'énergie de frottements de la hanche et de la jambe.

Ensuite on calcule le Lagrangien $L = T - V$. Par application de l'équation de Lagrange (2.1) aux coordonnées généralisées du robot, on obtient:

$$\begin{aligned} F_x =& (m_t + m_l + m_u)\ddot{x} - m_t d_e(\cos(\alpha+\phi)\dot{\phi}^2 + \sin(\alpha+\phi)\ddot{\phi}) \\ &+ (m_l d + m_u l)(\cos(\upsilon)\ddot{\upsilon} - \sin(\upsilon)\dot{\upsilon}^2) + m_u(2\cos(\upsilon)\dot{l}\dot{\upsilon} + \sin(\upsilon)\ddot{l}) \end{aligned} \tag{2.12}$$

$$\begin{aligned} F_z =& (m_t + m_l + m_u)\ddot{z} + m_t de(\cos(\alpha+\phi)\ddot{\phi} - \sin(\alpha+\phi)\dot{\phi}^2) \\ &+ (m_l d + m_u l)(\cos(\upsilon)\dot{\upsilon}^2 + \sin(\upsilon)\ddot{\upsilon}) + m_u(2\sin(\upsilon)\dot{l}\dot{\upsilon} - \cos(\upsilon)\ddot{l}) \\ &+ (m_t + m_l + m_u)g \end{aligned} \tag{2.13}$$

$$\begin{aligned} F_l =& m_u(\sin(\upsilon)\ddot{x} - \cos(\upsilon)\ddot{z} + \ddot{l} - l\dot{\upsilon}^2 - g\cos(\upsilon)) + k_l(l - d - p - s_0) \\ &+ c_l(\dot{l} - \dot{p}) \end{aligned} \tag{2.14}$$

$$\begin{aligned} \Gamma_\phi =& m_t d_e(-\sin(\alpha+\phi)\ddot{x} + \cos(\alpha+\phi)\ddot{z} + d_e\ddot{\phi} + g\cos(\alpha+\phi)) + J_t\ddot{\phi} \\ &+ c_h(\dot{\phi} - \dot{\upsilon}) \end{aligned} \tag{2.15}$$

$$\begin{aligned} \Gamma_\upsilon =& (m_l d^2 + m_u l^2)\ddot{\upsilon} + (m_l d + m_u l)(\cos(\upsilon)\ddot{x} + \sin(\upsilon)(\ddot{z} + g)) + J_l\ddot{\upsilon} \\ &+ 2m_u l\dot{l}\dot{\upsilon} + c_h(\dot{\upsilon} - \dot{\phi}) \end{aligned} \tag{2.16}$$

Ces dernières équations représentes la dynamique du robot sauteur Kangourou en phase de vol. Pour faciliter l'implémentation du modèle sous Matlab, nous avons mis ce système d'équations sous forme matricielle (voir chapitre précédent). Les détails de ces matrices sont données en annexe A.

2.2.2 Phase de contact

Le modèle dynamique durant la phase de support est obtenu à partir du modèle de la phase de vol en rajoutant à ce dernier les contraintes de contact avec le sol. Ces contraintes introduisent un terme supplémentaire qui représente les forces de contact avec le sol (cf. figure 2.3). Sous l'hypothèse que les frottements entre le pied et le sol sont suffisants pour empêcher le glissement. Les contraintes de contact avec le sol peuvent être exprimées comme suit:

$$\begin{cases} x_p = q_1 + q_3\sin(q_5) = cte \\ z_p = q_2 - q_3\cos(q_5) = 0 \end{cases} \tag{2.17}$$

où (x_p,z_p) sont les coordonnées cartésiennes du pied du robot.

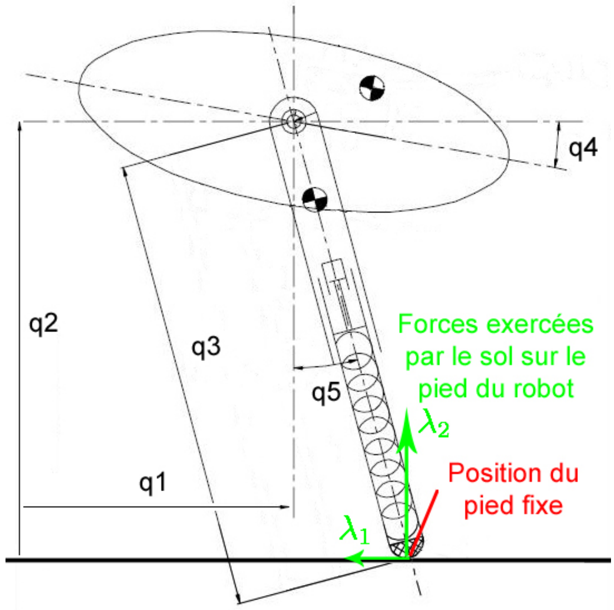

FIG. 2.3 – *Robot Kangourou: phase de contact*

Ces contraintes peuvent être écrites $\Phi(q) = 0$ avec:

$$\Phi(q) = \begin{pmatrix} q_1 + q_3 \sin(q_5) - cte \\ q_2 - q_3 \cos(q_5) \end{pmatrix} \tag{2.18}$$

On applique ensuite la dérivée première puis la dérivée seconde du vecteur $\Phi(q)$ par rapport au temps, ce qui donne:

$$\frac{\partial \Phi(q)}{\partial q_i} \dot{q} = J(q)\dot{q} = 0 \tag{2.19}$$

$$J(q)\ddot{q} + \frac{dJ(q)}{dt}\dot{q} = J(q)\ddot{q} + P(q,\dot{q}) = 0 \tag{2.20}$$

où $J(q)$ est la matrice Jacobienne des contraintes de contact avec le sol, $P(q,\dot{q})$ est un vecteur fonction de la dérivée de la Jacobienne.

Dans notre cas, la Jacobienne $J(q)$ et le vecteur $P(q,\dot{q})$ sont:

$$J(q) = \begin{pmatrix} 1 & 0 & \sin(q_5) & 0 & q_3 \cos(q_5) \\ 0 & 1 & -\cos(q_5) & 0 & q_3 \sin(q_5) \end{pmatrix} \text{ et } P(q,\dot{q}) = \begin{pmatrix} -q_3 \sin(q_5)\dot{q}_5 \\ q_3 \cos(q_5)\dot{q}_5 \end{pmatrix} \tag{2.21}$$

Le modèle résultant de la phase de contact est donné par:

$$\begin{cases} M(q).\ddot{q} + N(q,\dot{q}).\dot{q} + F(q) + G(q) = U + J(q)^T\lambda \\ J(q)\ddot{q} + P(q,\dot{q}) = 0 \end{cases} \tag{2.22}$$

où λ est le vecteur des multiplicateurs de Lagrange relatifs aux forces de contact avec le sol.

Le modèle peut être mis sous forme matricielle:

$$\underbrace{\begin{bmatrix} M(q) & -J(q)^T \\ J(q) & 0 \end{bmatrix}}_{W} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} + \underbrace{\begin{bmatrix} N(q,\dot{q})\dot{q} + F(q,\dot{q}) + G(q) - U \\ P(q,\dot{q}) \end{bmatrix}}_{V} = 0 \tag{2.23}$$

On peut extraire les accélérations et les multiplicateurs de Lagrange à partir de l'équation (2.23). Ils s'expriment alors sous la forme:

$$\begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = -W^{-1}V \tag{2.24}$$

2.2.3 Transitions entre phases

Dans les sections précédentes, nous avons modélisé le robot Kangourou dans ses deux phases principales, à savoir la phase de vol et la phase de support. Maintenant, il suffit d'enchaîner les modèles dynamiques de ces phases lors de la simulation pour voir l'évolution temporelle des coordonnées, des vitesses et des accélérations généralisées du robot. A présent, la difficulté consiste à déterminer les conditions de passage d'une phase à l'autre.

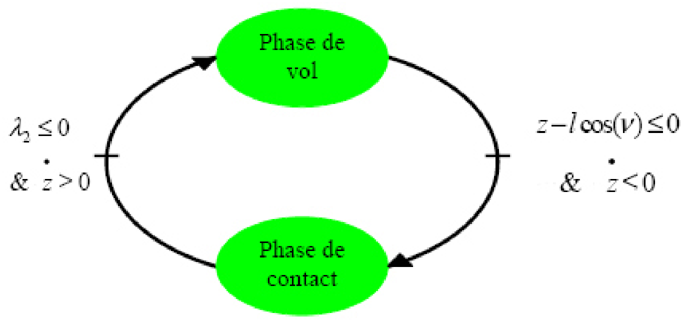

FIG. 2.4 – *Transitions entre phases*

Dans la phase de contact, la composante verticale de la force exercé par le sol λ_2 (cf. figure 2.3) ne peut être que positive. On est dans le cas d'une contrainte unilatérale, à savoir que le sol ne peut pas retenir le pied si celui-ci se soulève. Il faut que la soit force soit dirigé vers le haut pour que le contact soit persistant. Cependant, dans notre cas, les contraintes ajoutées pour garder le contact avec le sol peuvent engendrer des forces aussi bien positives que négatives. On détecte l'instant de passage d'une phase à l'autre en analysant à chaque instant la composante verticale de la force générée par les contraintes. Si celle-ci est nulle ou négative alors le robot passe à la phase de vol et peut ainsi décoller.

La détection du passage de la phase de vol à la phase de contact est assez intuitive. Il suffit de détecter l'instant où le pied du robot touche le sol. Ceci est réalisable car l'on connaît à chaque instant la position du corps dans l'espace et la longueur de la jambe.

Les transitions décrites ci-dessus doivent être complétées par d'autres conditions. En effet, dans cette configuration, le système peut basculer indéfiniment d'une phase à l'autre. Prenons l'exemple de l'atterrissage $z - l\cos(\upsilon) \leq 0$, le système étant échantillonné, la position verticale du pied du robot sera certainement légèrement inférieure à zéro. Le pied est alors théoriquement légèrement enfoncé dans le sol est le reste durant la phase de contact. Au moment du décollage, il se peut que la position du pied soit toujours inférieure au sol, ceci satisferait la condition $z - l\cos(\upsilon) \leq 0$ et le robot se retrouverait de nouveau dans la phase de contact. Le système enchaînera ces transitions indéfiniment. Pour éviter cela il faut prendre en compte les vitesses verticales, c'est ainsi qu'il faut vérifier que le robot monte pour pouvoir décoller et descende lors de l'atterrissage, ce qui donne les transitions suivantes:

$$T_{contact \longrightarrow vol} \quad : \quad \lambda_2 \leq 0 \quad \& \quad \dot{z} > 0 \tag{2.25}$$

$$T_{vol \longrightarrow contact} \quad : \quad z - l\cos(\upsilon) \leq 0 \quad \& \quad \dot{z} \leq 0 \tag{2.26}$$

Un dernier problème subsiste lors de l'atterrissage. A cet instant, on applique les contraintes sur le pied pour garder le contact avec le sol (voir section 2.2.2). Ces contraintes permettent d'annuler les accélérations au niveau du pied (voir équation (2.23)) mais pas les vitesses. Ainsi, lorsque le pied arrive avec une certaine vitesse dans la phase de vol, on applique les contraintes, le pied n'accélère plus mais conserve sa vitesse initiale. Le pied ne s'immobilise donc pas au moment de l'atterrissage. Pour palier ce problème numérique, il serait intéressant d'annuler la vitesse du pied, avant de commencer la phase de contact.

A partir des coordonnées du pied (voir équation (2.17)), on en déduit les vitesses cartésiennes du pied:

$$\left\{ \begin{array}{l} \dot{x}_p = \dot{q}_1 + \dot{q}_3 \sin(q_5) + q_3 \dot{q}_5 \cos q_5 = 0 \\ \dot{z}_p = \dot{q}_2 + \dot{q}_3 \cos(q_5) - q_3 \dot{q}_5 \sin q_5 = 0 \end{array} \right. \tag{2.27}$$

On peut ensuite extraire les vitesses généralisées que doit satisfaire le robot au moment de l'atterrissage et qui permettent d'annuler la vitesse du pied:

$$\begin{cases} \dot{q}_3 = \dfrac{\dot{q}_2 + q_3\dot{q}_5 \sin(q_5)}{\cos(q_5)} \\ \dot{q}_5 = -\dfrac{\dot{q}_1 + \dot{q}_2 \tan(q_5)}{q_3(\cos(q_5) + \sin(q_5)\tan(q_5))} \end{cases} \tag{2.28}$$

Au moment de l'atterrissage, avant de passer à la phase de contact, il est nécessaire de modifier les vitesses généralisées du système pour introduire les vitesses telles qu'elles sont décrites par les équations (2.28).

2.2.4 Implémentation sous Matlab

Le logiciel Matlab permet, à l'aide des équations dynamiques du système, de pouvoir simuler le modèle dynamique du robot sauteur et de suivre son évolution temporelle. Les équations dynamiques calculées dans les sections précédentes permettent de déduire les accélérations du système à partir des forces et des couples de commande, des coordonnées et des vitesses généralisées.

La fonction **ODE45** de Matlab permet d'intégrer la dérivée du vecteur d'état à un instant donné en se basant sur la méthode de Runge-Kutta afin d'obtenir le vecteur d'état au prochain instant d'échantillonnage. En connaissant les coordonnées et les vitesses initiales du robot Kangourou, on peut ainsi simuler les mouvements du robot. Le nombre de paramètres pris en compte dans la modélisation déterminera l'exactitude du comportement du robot en mode simulation par rapport à son comportement réel.

Chapitre 3

Le simulateur développé

La modélisation du robot Kangourou terminée, la prochaine étape consiste à l'implémenter sous Matlab afin de pouvoir simuler le comportement du robot et les lois de commandes proposées. Dans le but d'optimiser cette étape essentielle, nous avons développé un simulateur qui sera détaillé dans la section suivante.

3.1 Interface graphique

En plus du calcul numérique et symbolique, Matlab donne la possibilité de réaliser des interfaces graphiques. Une interface graphique peut comporter des fenêtres, des boutons poussoirs, des menus déroulants, des curseurs, des éditeurs de texte, etc.

Les interfaces graphiques donne également possibilité à l'utilisateur de communiquer avec les programmes (choix entre différentes alternatives, choix de certains paramètres, choix des conditions initiales, etc). Dans ce contexte et afin de réaliser un simulateur de notre robot "Kangourou", on a adopté cette solution. Le simulateur développé à l'aide de **GUI** (Graphical User Interface an anglais) de Matlab sera présenter dans la section suivante.

3.1.1 Description du simulateur développé

L'interface graphique de notre simulateur est représenté sur la figure 3.1.C'est un ensemble d'objets graphiques (boutons, menus, curseurs, ...) permettant d'interagir entre des (parties de) programmes et l'utilisateur en affichant des informations (textes, graphiques, images, ...) et en déclenchant des actions (callbacks) à la suite d'événements : clic de souris, déplacement de souris, entrée au clavier, ...).

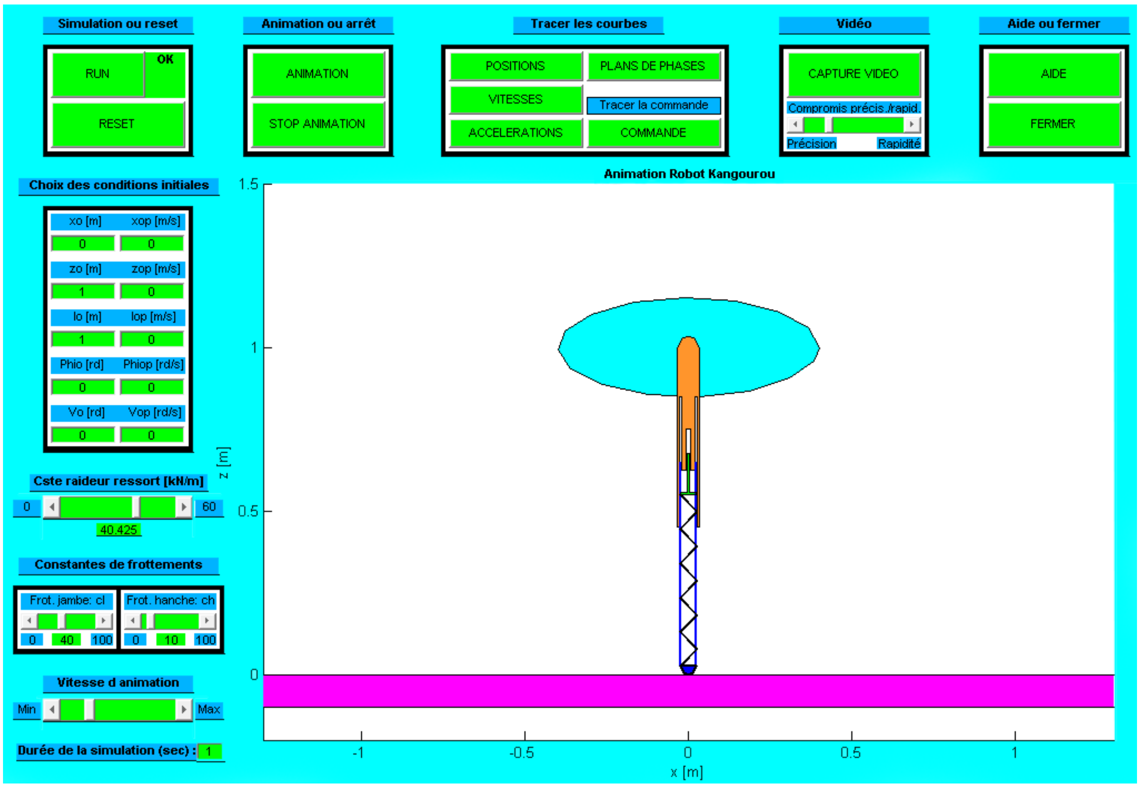

FIG. 3.1 – *Interface graphique du robot Kangourou*

Elle est composée de 11 rubriques (voir figure 3.1):

- ✓ Simulation ou reset
- ✓ Animation ou arrêt
- ✓ Tracer les courbes
- ✓ Vidéo
- ✓ Aide ou fermer
- ✓ Choix des conditions initiales
- ✓ Constante de raideur du ressort
- ✓ Constantes de frottements
- ✓ Vitesse d'animation
- ✓ Durée de la simulation
- ✓ Affichage

Cette interface graphique nous permet de choisir sans difficultés, les conditions initiales et les paramètres du robot, ainsi que les paramètres de simulation. Elle permet de visualiser les mouvements du robot, d'afficher les courbes de simulation et d'acquérir des vidéos.

3.1.2 Fonctions du simulateur

Dans cette partie, nous allons décrire les principales fonctions du simulateur. Elles sont regroupées dans la bande supérieure, au dessus de la fenêtre graphique:

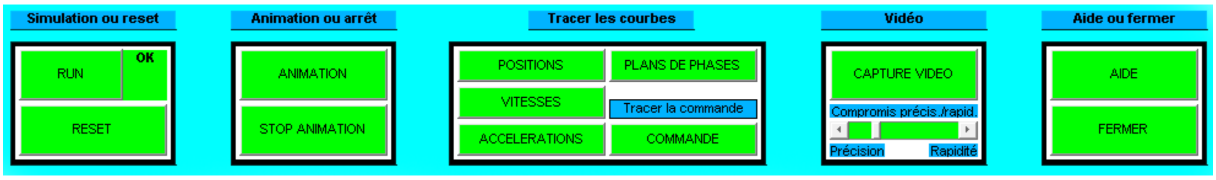

FIG. 3.2 – *Fonctions principales du simulateur*

Simulation ou reset : Permet de lancer la simulation du modèle dynamique avec les paramètres de simulation choisis. Elle permet également de réinitialiser le simulateur.

Animation ou arrêt : Permet de lancer l'animation graphique du robot Kangourou à partir de la dernière simulation exécutée. Cette animation sera visible sur la fenêtre graphique.

Tracer les courbes : Permet de tracer les courbes de simulation. On peut ainsi visualiser l'évolution des coordonnées, des vitesses et des accélération généralisées, et afficher les plans de phases et les courbes de commandes.

Vidéo : Permet de générer une séquence d'image qui sera utilisé pour créer une vidéo sous format .avi à l'aide du logiciel "Fast Movie Processor 1.44". Un curseur permet de fixer la cadence avec laquelle on veut générer la séquence d'image.

Aide ou fermer : Permet de sortir du simulateur en fermant la fenêtre d'interface graphique ou d'appeler une fenêtre d'aide qui explique l'utilisation de ce simulateur.

3.1.3 Choix des conditions initiales et paramètres du robot

Avant de lancer la simulation, il faut préalablement choisir les conditions initiales du robot Kangourou: coordonnées et vitesses généralisées (celles-ci sont décrites à la section 2.1). On peut également ajuster certains paramètres du robot tels que la raideur du ressort ou les coefficients de frottement et voir aisément leurs influences sur la simulation. Ces paramètres sont initialisés à l'ouverture du simulateur et sont regroupés dans la bande à gauche de l'affichage du simulateur:

Choix des conditions initiales :

xo et xop : Choix de la position initiale x_0 et la vitesse initiale $\dot{x}_0$ du centre du corps du robot.

zo et zop : Choix de la position initiale z_0 et la vitesse initiale $\dot{z}_0$ du centre du corps du robot.

lo et lop : Choix de l'état initial de la jambe (longueur l_0 et sa variation $\dot{l}_0$).

Phio et Phiop : Choix de la position angulaire initiale ϕ_0 et de sa vitesse initiale $\dot{\phi}_0$ du corps du robot par rapport à l'horizontale.

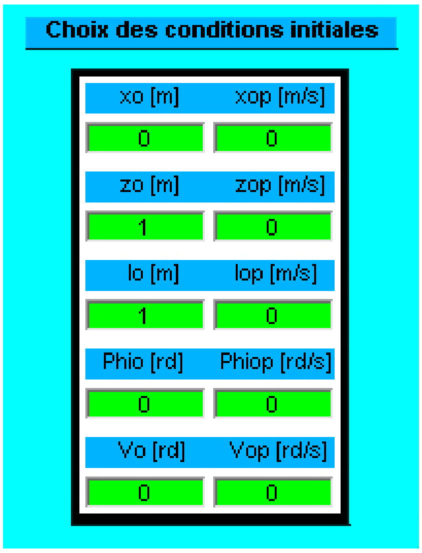

FIG. 3.3 – *Choix des conditions initiales*

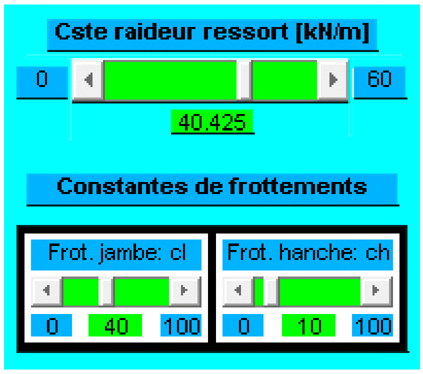

FIG. 3.4 – *Choix des paramètres du robot*

Vo et Vop : Choix de la position angulaire initiale v_0 et de sa vitesse initiale $\dot{v}_0$ de la jambe du robot par rapport à la verticale.

Choix des paramètres du robot :

Constante de raideur du ressort : Permet d'ajuster la constante de raideur k_l du ressort.

Constantes de frottements : Permet d'ajuster les constantes de frottements entre la partie supérieure et inférieure de la jambe c_l, et entre le corps et la jambe du robot c_h, au niveau de la hanche.

3.1.4 Réglages de simulation et d'animation

Cette partie du simulateur regroupe un éditeur de texte, une barre glissante et bien sûr la fenêtre graphique (cf. figures 3.5 et 3.6).

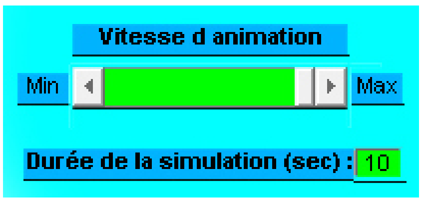

FIG. 3.5 – *Réglages de simulation et d'animation*

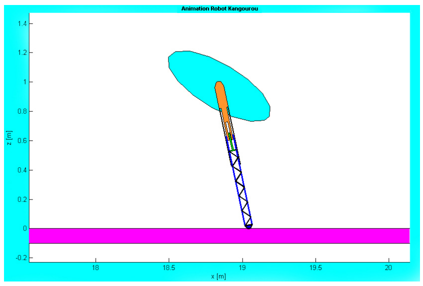

FIG. 3.6 – *Fenêtre d'animation*

Vitesse d'animation : Permet de régler la vitesse d'animation.

Durée de la simulation : Permet à l'utilisateur de saisir au clavier la durée de simulation désirée.

Affichage : Fenêtre graphique permettant de visualiser le robot sauteur Kangourou pendant son déplacement.

3.2 Première validation du simulateur

Le simulateur est opérationnel, nous pouvons désormais le tester et vérifier si le comportement du robot est cohérent. Nous réalisons une première simulation en inclinant légèrement le corps de façon à ce que son centre de gravité ne soit pas d'en alignement de celui de la jambe et du pied. Le système est simulé en boucle ouverte (commande nulle), de cette façon on simule le **modèle passif** du robot Kangourou. Les paramètres du robot sont présentés dans l'annexe B.

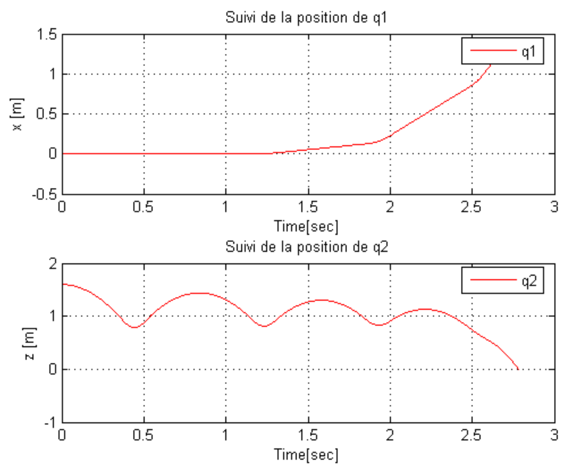

Fig. 3.7 – *Variation de x et z en fonction du temps*

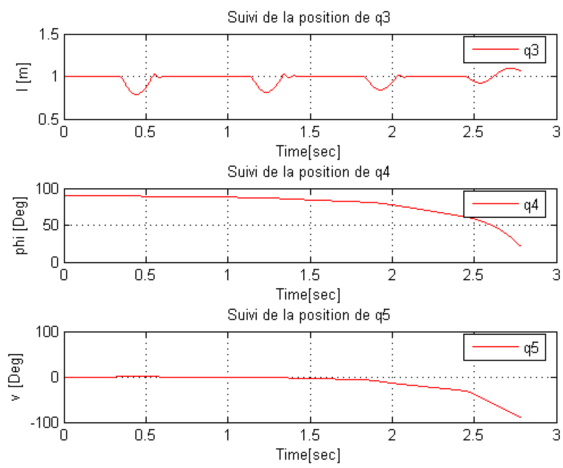

Fig. 3.8 – *Variation de l, ϕ et v en fonction du temps*

Ces premiers résultats sont assez concluants. Le robot lâché en l'air, tombe sous l'effet de la gravité, touche le sol, le ressort se comprime puis se détend pour pousser le robot vers le haut. Ce dernier saute et retombe à nouveau et le cycle se répète jusqu'à ce qu'il soit en déséquilibre, ce qui provoque sa chute. Les courbes qui retranscrivent les positions $q_2 = z$ et $q_3 = l$ permettent d'identifier les déplacements verticaux du robot et les contractions et décontractions de la jambe. L'angle donné au corps dans les conditions initiales entraîne le robot vers le déséquilibre. Le robot en sautant, avance puis tombe (cf. figure 3.7). On peut remarquer que le système perd de l'énergie, ceci est dû aux frottements pris en compte dans la modélisation.

Cette simulation préliminaire a permit avec d'autres, de valider aussi bien le modèle dynamique du robot décrit dans le chapitre 2 que le simulateur développé. A présent, nous pouvons tester les différentes lois de commandes qui seront proposées dans le prochain chapitre pour contrôler les déplacements du robot Kangourou. L'objectif de ces lois de commande sera de réaliser des mouvements périodiques stables.

Chapitre 4

Approches de commandes proposées

Dans ce chapitre, les différentes approches de commande proposées pour le contrôle du robot sauteur Kangourou seront présentées. Le chapitre s'articule autour de deux parties principales. Dans la première, deux approches de commande sont proposées pour le contrôle du saut vertical du robot et dans la deuxième, deux approches sont proposées pour la commande du saut pour le déplacement planaire du robot.

4.1 Commande pour le saut vertical

Le saut vertical (cf. figure 4.1) est le mouvement le plus simple que peut réaliser le robot. Il se déplace en s'éjectant vers le haut grâce à l'actionneur et le ressort se trouvant au niveau de la jambe. Les mouvements du robot, dans ce cas, se limitent à la seule direction verticale.

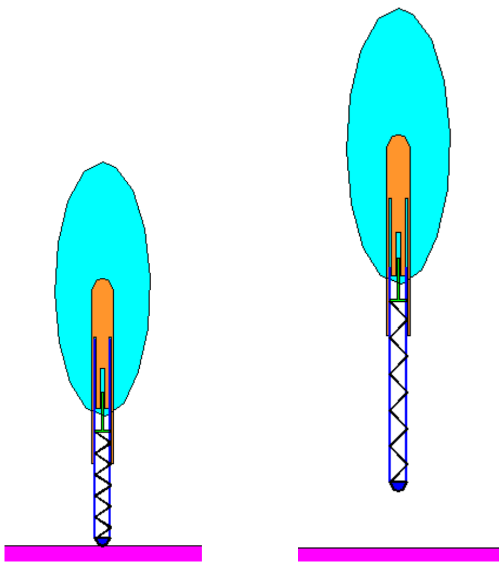

FIG. 4.1 – *Saut vertical du robot Kangourou (à gauche: phase de contact, à droite: phase de vol).*

Le fait de se limiter aux seuls mouvements verticaux permet de se concentrer uniquement sur la commande de la poussée de la jambe. Les études réalisées sur ce type de robot ont parmis d'améliorer les connaissances de l'influence de la poussée de la jambe sur le comportement des robots sauteurs [7][16].

Dans le cas de notre robot, il suffit d'aligner verticalement les centres de gravité du corps, de la jambe et du pied. Si aucune perturbation horizontale ne vient déstabiliser le robot, il ne pourra évoluer que dans la direction verticale.

4.1.1 Approche 1: Commande en boucle ouverte

Le principe de cette approche de commande consiste à appliquer, par l'intermédiaire de l'actionneur linéaire, une force constante sur la jambe pendant la décompression du ressort. Pendant la phase de contact, on analyse les variations de vitesse subies par la jambe, ce qui va permettre de détecter l'instant approprié pour appliquer la commande. Au moment où la jambe commence à s'allonger, annonçant la décontraction du ressort, on applique une commande sous forme d'une force constante:

$$F_l = cte \quad , \quad \text{si} \quad \dot{l} > 0 \tag{4.1}$$

Cette commande permet compenser les pertes dues aux frottements en injectant de l'énergie durant la phase de contact. Les paramètres du robot sont identiques à ceux de la simulation du modèle passif et sont donnés dans l'annexe B. La force appliquée à la jambe est de 200 N, les résultats de simulations sont présentées sur les figures 4.2 et 4.3.

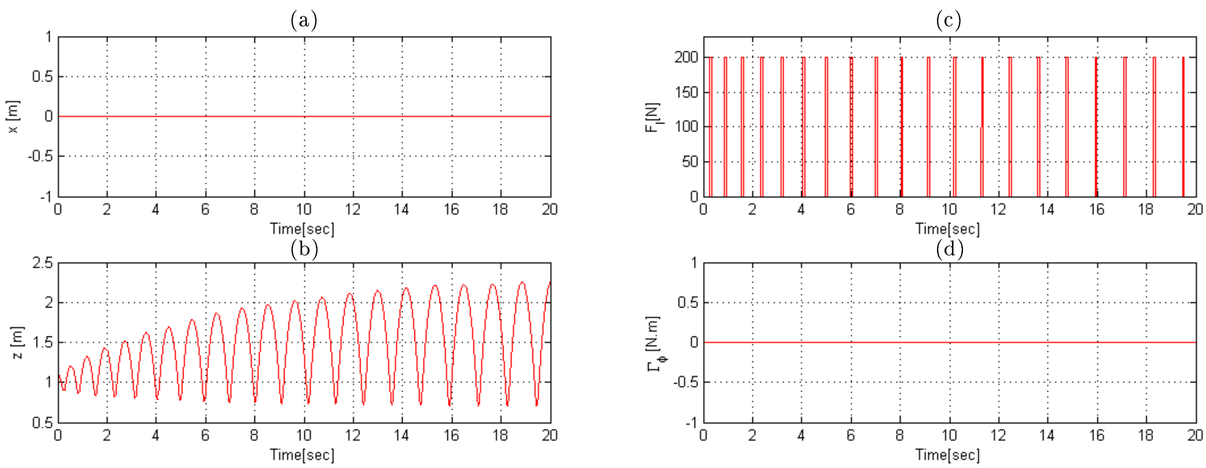

FIG. 4.2 – *Commande pour le saut vertical en boucle ouverte: variables d'états et commandes*

La commande F_l (cf. figure 4.2.(c)) permet de stabiliser la hauteur du saut au bout d'environ 14 s (cf. figure 4.2.(b)). Cette convergence se voit sur la figure 4.3 qui représente les plans de

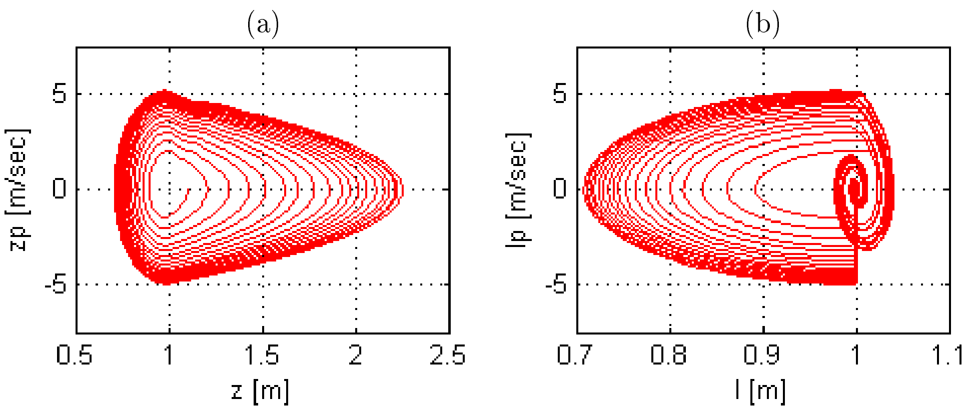

FIG. 4.3 – *Commande pour le saut vertical en boucle ouverte: plans de phases*

phase $(z,\dot{z})$ et $(l,\dot{l})$ où on remarque clairement la convergence vers des cycles limites stables. Etant donné que cette commande est en boucle ouverte, elle ne permet pas de suivre une consigne sur la hauteur du saut. L'inconvénient de cette commande est sa sensibilité vis-à-vis des incertitudes ou des variations dans les paramètres du système. En effet, si un paramètre est inconnu ou varie au cours du temps, il est impossible de contrôler correctement le robot avec ce type de commande, vu qu'il n'y a pas de bouclage.

4.1.2 Approche 2: Commande en boucle fermée

D'après les résultats de la commande en boucle ouverte, il s'avert indispensable de boucler le système afin d'améliorer les performances de la commande. L'objectif de la commande en boucle fermée est de suivre une consigne sur la hauteur du saut que l'on désire atteindre. Dans la phase de vol, le robot suit une trajectoire balistique décrite par:

$$z(t) = z_{lo} + \dot{z}_{lo}t - \frac{gt^2}{2} \quad \Rightarrow \quad z_{ap} = \frac{\dot{z}_{lo}^2}{2g} + z_{lo} \quad , \quad T_f = \frac{2\dot{z}_{lo}}{g} \tag{4.2}$$

où z_{ap} est la hauteur maximale atteinte par le corps appelé "apex" et T_f la durée de la phase de vol qui sont tous deux, fonctions de la hauteur du corps au décollage z_{lo} et de la vitesse verticale au décollage $\dot{z}_{lo}$.

A partir des équations (4.2), nous pouvons déterminer la vitesse verticale que doit avoir le robot pour atteindre une hauteur de saut donnée:

$$\dot{z}_{lo} = \sqrt{2g(z_{apd} - z_{lo})} \tag{4.3}$$

Désormais, nous connaissons la vitesse que le robot doit avoir au décollage pour atteindre une hauteur de saut désirée. Il suffit de suivre cette vitesse lorsque le ressort se décontracte dans la phase de contact (voir section 4.1.1). La force que doit exercer l'actionneur est donnée par:

$$F_l = \beta(\dot{z}_{lo} - \dot{z}) \qquad , \text{si} \quad \dot{l} > 0 \quad \& \quad l < z_{lo} \tag{4.4}$$

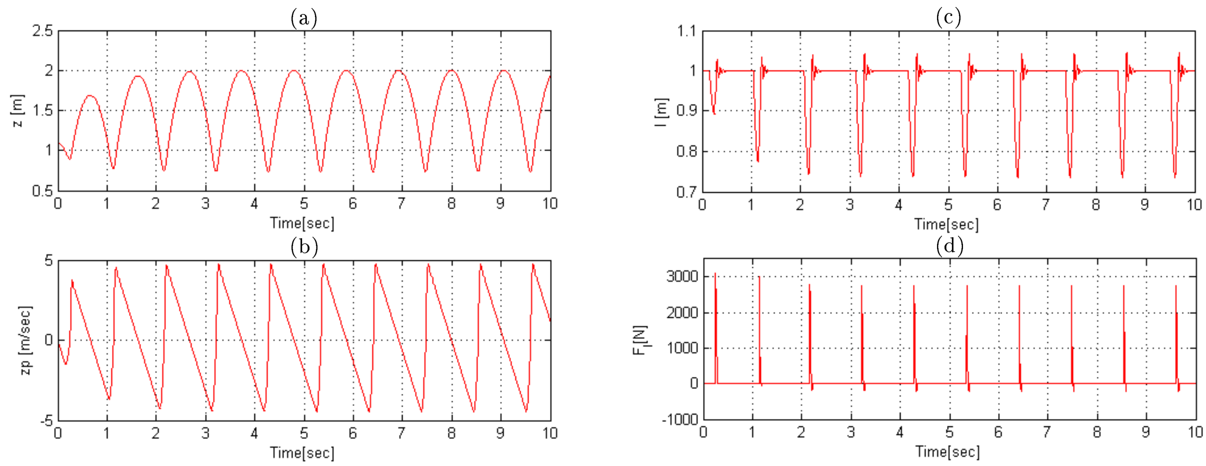

FIG. 4.4 – *Commande pour le saut vertical en boucle fermée: évolution du signal de commande et quelques états du système*

Les paramètres du robot sont identiques à ceux de la simulation du précédent et sont donnés dans l'annexe B. Le gain β utilisé pour la commande (voir équation (4.4)) est de 700 et la hauteur de saut désirée est de 2 m. les résultats de simulations sont présentées sur les figures 4.4 et 4.5.

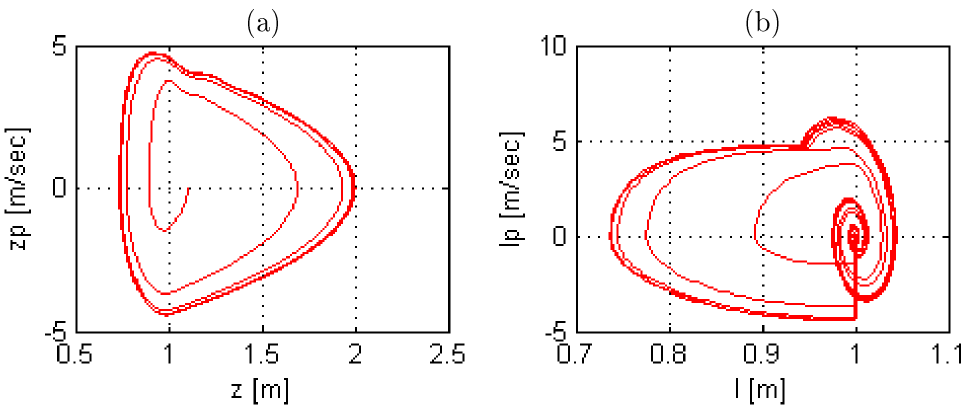

FIG. 4.5 – *Commande pour le saut vertical en boucle fermée: plans de phase*

La figure 4.4.(d) représente le signal de commande (force générée par l'actionneur linéaire). L'évolution de la longueur l de la jambe est représentée sur la figure 4.4.(c) où on remarque de petits mouvements oscillatoires à l'instant de décollage, ceci est dû à la compliance du ressort. Les figures 4.4.(a) et 4.4.(b) représentent la position et la vitesse verticales de la hanche où on remarque la convergence vers des trajectoires cycliques. Cette convergence est confirmée sur la figure 4.5 qui représente les plans de phase $(z,\dot{z})$ et $(l,\dot{l})$. D'après ces résultats, la consigne est parfaitement suivie mais la force de commande permettant d'exercer la poussée au niveau de la jambe est assez importante. On peut diminuer celle-ci en diminuant le gain de commande β, cependant le système convergera moins vite vers la hauteur de saut désirée. On peut également réduire les frottements au niveau de la jambe pour diminuer l'énergie perdue durant la phase de contact et par conséquent la commande aura moins d'énergie à fournir. Vu que tout actionneur est physiquement limité, il faudra insérer des saturations au niveau de la commande. La valeur de la saturation dépendra de l'actionneur utilisé.

En comparant cette commande avec celle en boucle ouverte, on constate que la convergence est beaucoup plus rapide (ceci se voit sur les figures 4.4.(a) et 4.5), et la précision est beaucoup plus élevée (la hauteur de saut désirée de 2 m est atteinte).

4.2 Commande du saut pour le déplacement planaire

4.2.1 Principe de base

Le déplacement planaire introduit de nouvelles difficultés par rapport au saut vertical pur. Alors que nous nous préoccupions que de la poussée exercée par l'actionneur linéaire au niveau de la jambe. Nous devons, à présent, prendre en considérations les mouvements dans le plan sagittal XOZ pour effectuer le saut vers l'avant tout en se déplaçant horizontalement.

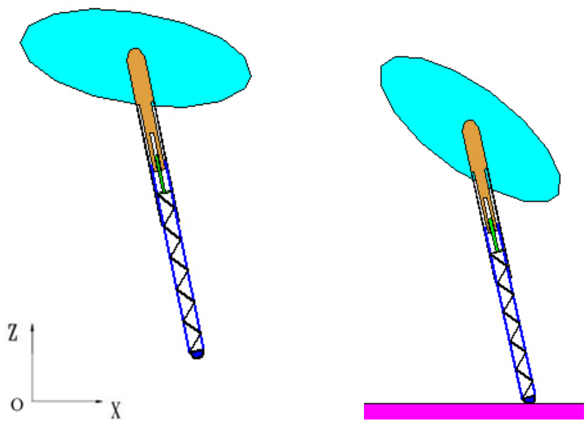

FIG. 4.6 – *Saut planaire du robot Kangourou (à gauche: phase de vol, à droite: au moment de l'atterrissage)*

Analysons le comportement de notre robot pendant un cycle de saut. Dans la phase de vol (cf. figure 4.6), le robot est en mouvement balistique, il n'a aucun support lui permettant de modifier sa trajectoire. Le corps a une masse plus importante que celles de la jambe et le pied réunis, il servira d'appui pour positionner correctement le pied au moment de l'atterrissage.

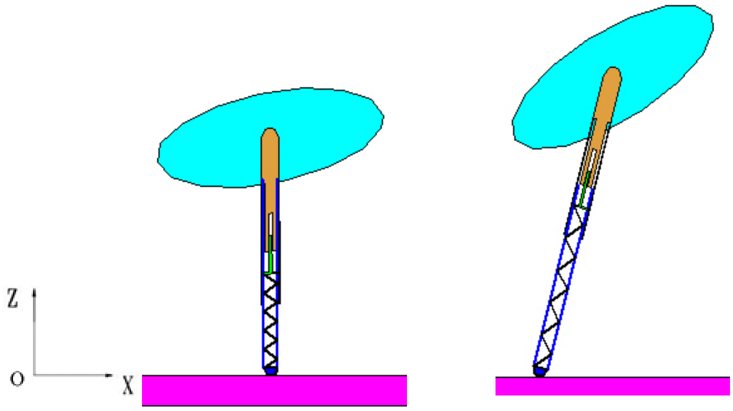

FIG. 4.7 – *Saut planaire du robot Kangourou (à gauche: phase de contact, à droite: moment du décollage)*

Après l'atterrissage, le pied du robot est en contact avec le sol, nous avons un point d'appui ce qui nous permettra de modifier l'inclinaison du corps. Pendant la phase de contact (cf. figure 4.7), le ressort accumule l'énergie du saut précédent, puis la libère en se décompressant. La poussée exercée par l'actionneur avant le décollage permet de contrôler l'action du ressort en agissant sur l'énergie du système.

4.2.2 Etat de l'art

Depuis une vingtaine d'années, de nombreux scientifiques se sont intéressés à la commande des robots sauteurs. Les robots sauteurs à 2 dimensions ont été très étudiés puisque leur principe de commande est facilement généralisable aux cas de robots plus complexes tels que les robots sauteurs 3D, les robots sauteurs bipèdes et quadrupèdes.

L'un des pionniers de la commande des robots sauteurs est Marc Raibert, fondateur du Laboratoire Leg au MIT en 1980. Il est à l'origine de la plupart des commandes actuelles dans le domaine des robots sauteurs. Au début des années 80, il imagina une commande en trois parties permettant de contrôler un robot sauteur unijambiste planaire [10].

La première partie de la commande consiste à exercer une poussée constante au niveau de la jambe pendant l'extension de la jambe (phase de contact) pour commander la hauteur du saut.

La seconde partie permet de contrôler la vitesse de déplacement. En effet, durant la phase de vol, il commande la position de la jambe du robot de sorte que la position du pied au moment de l'atterrissage, permettra au robot d'accélérer ou de ralentir. Enfin, la dernière partie consiste à suivre une trajectoire de référence pour commander le corps du robot pendant la phase de contact. Cette commande sera développée dans la section 4.2.3, elle a été étendue par Raibert au cas d'un robot sauteur en 3 dimensions.

Dans la literature, on trouve aussi les robots passifs. Ce sont des robots qui dispose d'une compliance au niveau de la jambe et de la hanche. Ceci leur permet de se déplacer sans actionneurs. Cependant, leur mobilité est très réduite car les frottements mécaniques engendrent une perte d'énergie du système. En ajoutant des actionneurs sur ce type de robot, certaines commandes furent spécialement conçus pour compenser les effets des frottements sur ce type de systèmes [16][6].

La commande de la vitesse de déplacement de Raibert a été améliorée à plusieurs reprises. Les améliorations consistaient à calculer plus précisément l'angle de la jambe au moment de l'atterrissage [5] [13]. Certaines recherches ont abouti à une nouvelle expression de cette commande de vitesse dans un autre référentiel que le temps. La variable introduite était mieux adaptée pour décrire les mouvements des robots sauteurs [2].

De nouvelles approches pour commander la hauteur du saut du robot ont été développées. Au décollage, un calculateur calcule les équations dynamiques du système jusqu'au prochain décollage. Il calcule ensuite la longueur de la jambe appropriée pour réaliser le saut désiré [8][9].

Récemment Sato et ses collaborateurs ont élaboré une nouvelle commande sur un modèle simplifié d'un robot sauteur comportant un actionneur unique au niveau de la hanche. Cette approche a permis la génération de mouvements périodiques stables en simulation [11] et en expérimentation [12].

4.2.3 Approche 1: Commande de Raibert

Le système de contrôle de Raibert traite le saut, la vitesse de déplacement et le comportement du corps comme trois problèmes séparés, ils sont présentés dans les paragraphes suivants:

Contrôle de la hauteur du saut

Le principe de cette approche de commande consiste à appliquer, par l'intermédiaire de l'actionneur linéaire, une force constante sur la jambe pendant la décompression du ressort.

Cette commande est détaillée dans la section 4.1.1.

Contrôle de la vitesse de déplacement

Le système étant sous-actionné, il n'est pas possible de commander directement le déplacement horizontal du robot. La position du pied quand il touche le sol, à la fin de la phase de vol, a une importance cruciale sur l'évolution de la partie passive du robot pendant la phase de contact suivante (cf. figure 4.8).

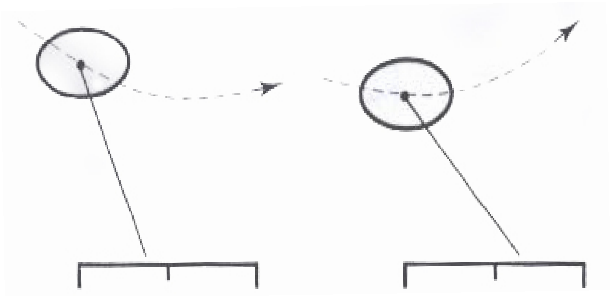

FIG. 4.8 – *Trajectoires asymétriques: le déplacement du pied permettra au robot d'accélérer ou de décélérer*

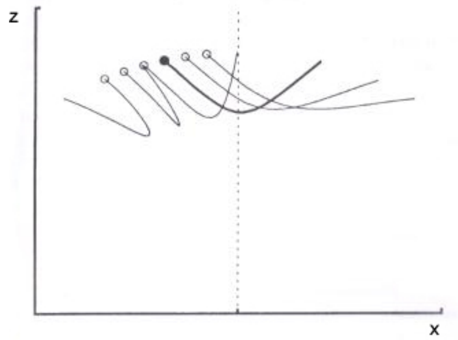

FIG. 4.9 – *Trajectoire du corps pendant la phase de contact pour différentes positions du pied.*

Différentes trajectoires du corps sont représentés sur la figure 4.9 pour différentes positions du pied. Initialement, les vitesses du robots étaient les mêmes pour chaque trajectoire, c'est la position du pied qui détermine la trajectoire du robot pendant la phase de contact. On remarque que pour une vitesse donnée, il existe une seule position du pied à l'atterrissage qui permette une trajectoire symétrique pendant la phase de contact (courbe en gras sur la figure 4.9). Cette position du pied est appelé "*la position neutre*". La trajectoire ainsi obtenue permet de conserver la vitesse horizontale du robot pendant toute la durée de la phase de contact (cf. figure 4.10).

La phase de vol ne modifie pas la vitesse horizontale du robot car la force gravitationelle n'a d'influence que sur l'accélération verticale du robot. Par conséquent, si l'on parvient à positionner le pied correctement au moment de l'atterrissage, la vitesse du robot restera constante pendant toute la durée du cycle de saut.

La vitesse moyenne de déplacement du robot est identique pendant toute la phase de contact. Dès lors, la position du pied pour avoir une trajectoire symétrique doit se trouver a mi-chemin de la distance parcourue dans la phase de contact (cf. figure 4.10). Cette position peut être caractérisée par l'équation (4.5).

$$x_{f0} = \frac{\dot{x}T_s}{2} \tag{4.5}$$

où

x_{f0} est la position du pied par rapport au centre de gravité global du robot

$\dot{x}$ est la vitesse du déplacement horizontal du robot

T_s est la durée de la phase de contact

La phase de contact peut être décrite comme une contraction suivie d'une décontraction du ressort. Le système se comporte ainsi comme un système masse-ressort. La période d'oscillation de ce type de système est indépendante de l'amplitude des oscillations. La durée de la phase de contact T_s est donc pratiquement constante pour une raideur du ressort donnée. En positionnant correctement le pied, au moment de l'atterrissage, on parvient ainsi à maintenir une vitesse moyenne de déplacement du robot constante.

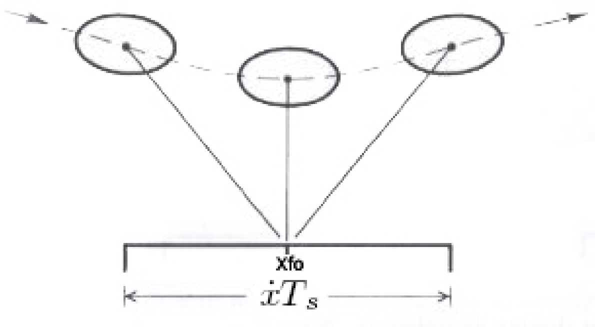

FIG. 4.10 – *Trajectoire symétrique du robot pendant la phase de contact*

A présent, il serait intéressant de pouvoir contrôler la vitesse de déplacement du robot dans le but de pouvoir accélérer ou décélérer le système. Ces accélérations, positives ou négatives, se font par l'intermédiaire d'un système de contrôle qui introduit des trajectoires asymétriques (cf. figure 4.8). Cette commande permet de déplacer le pied proportionnellement à l'erreur entre la vitesse horizontale du robot et la vitesse de déplacement désirée. Ce déplacement s'effectue par rapport à la "position neutre" qui permettait d'engendrer une trajectoire symétrique du robot pendant la phase de contact (voir équation (4.6)).

$$x_f = \frac{\dot{x}T_s}{2} + k_{\dot{x}}(\dot{x} - \dot{x}_d) \tag{4.6}$$

où

x_f est la position du pied par rapport au centre de gravité global du robot

$\dot{x}_d$ est la vitesse désirée du déplacement du robot

$k_{\dot{x}}$ est le gain de rétroaction

Ce système de contrôle devrait être assez robuste par rapport aux perturbations puisqu'on prend en compte l'erreur entre la vitesse de déplacement actuelle et la vitesse désirée dans la commande. Pour contrôler la position du pied au moment de l'atterrissage, il suffit de positionner la jambe par rapport au corps pendant la phase de vol, ce qui peut être décrit par :

$$\gamma_d = \phi - \arcsin\left(\frac{\dot{x}T_s}{2l} + \frac{k_{\dot{x}}(\dot{x} - \dot{x}_d)}{l}\right) \tag{4.7}$$

γ représente l'angle entre la jambe et l'axe longitudinal du corps du robot. On utilisera une commande PD (Proportionnelle Dérivée) classique pour contrôler la position de la jambe dans la phase de vol.

Contrôle du comportement du corps

Durant le mouvement balistique, il est impossible de contrôler la position du corps dans l'espace. Celui-ci a une masse beaucoup plus importante que celle de la jambe et du pied réuni. Il sert ainsi de support pour positionner la jambe correctement avant l'atterrissage (voir paragraphe précédant). La phase de contact est donc le seul moment où l'on peut repositionner le corps du robot. Le contact entre le sol et le pied du robot dans cette phase permet au couple fourni par l'actionneur angulaire de la hanche de piloter l'angle du corps sans engendrer de fortes accélérations au niveau de la jambe. Le système de contrôle permet de commander ce couple en utilisant un contrôleur PD classique et permet au corps de suivre la trajectoire désirée :

$$\Gamma = k_p(\phi_d - \phi) - k_v(\dot{\phi}) \tag{4.8}$$

où

Γ est le couple exercé par l'actionneur angulaire de la hanche

ϕ est l'angle du corps par rapport à l'horizontale

k_p,k_v sont respectivement les gains propotionnel et dérivé du contrôleur

Résultats de simulation

Nous avons intégré au simulateur de notre robot la commande de Raibert. Les paramètres de simulation sont donnés en annexe C. Les résultats de simulation obtenus sont présentées sur les figures 4.11, 4.12 et 4.13.

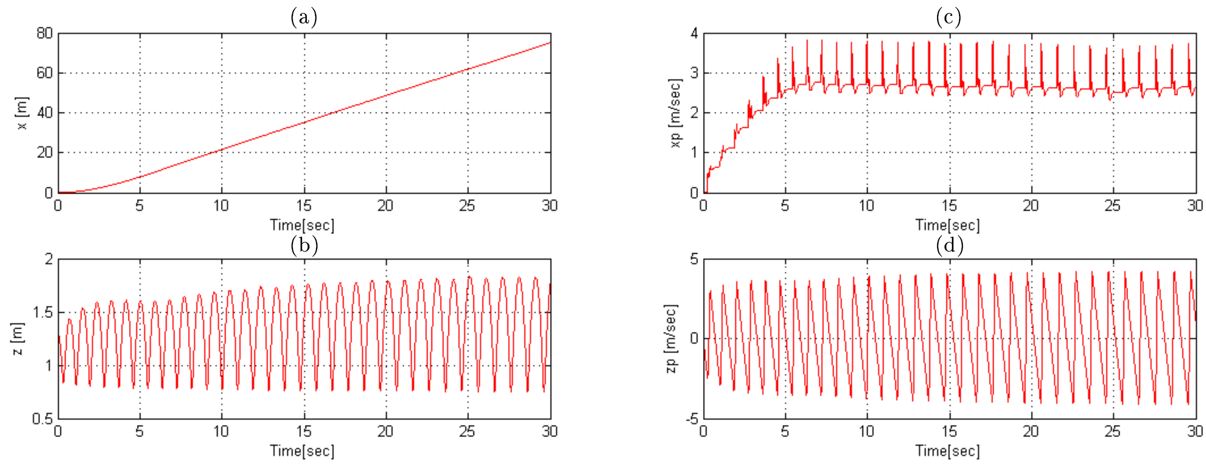

Fig. 4.11 – *Commande de Raibert: évolution des positions et vitesses cartésiennes*

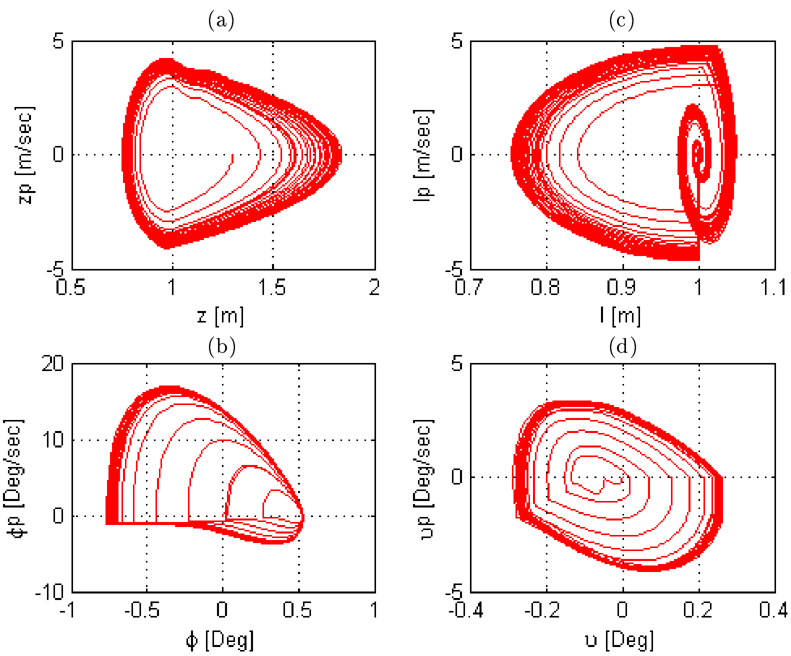

Fig. 4.12 – *Commande de Raibert: plans de phases*

Les déplacements verticaux (cf. figure 4.11.(b)) et horizontaux (cf. figure 4.11.(a)) nous montre que la commande proposée est susceptible de contrôler les sauts du robot pour converger vers des trajectoires cycliques. La hauteur du saut (cf. figure 4.11.(c)) augmente progressivement avant de se stabiliser vers 1.8 m au bout de 15 s. La vitesse initiale du déplacement du robot (cf. figure 4.11.(c)) est nulle puis elle converge rapidement vers 2.6 m/s. La commande de la vitesse de déplacement de Raibert permet une accélération satisfaisante du robot mais engendre une importante erreur de vitesse puisque la consigne sur la vitesse de déplacement est de 1 m/s (voir annexe C).

Les plans de phase $(z,\dot{z})$, $(\phi,\dot{\phi})$, $(l,\dot{l})$ et $(v,\dot{v})$, représentés sur la figure 4.12, mettent en évidence la convergence vers des cycles limites stables. La trajectoire de la hanche du robot Kangourou dans le plan sagittal est représentée sur la figure 4.13.(a).

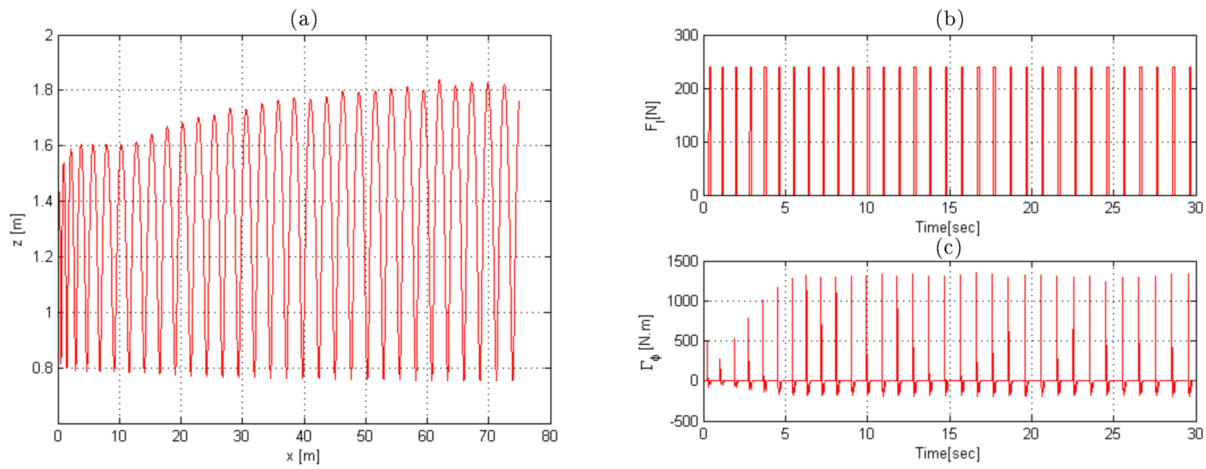

Fig. 4.13 – *Commande de Raibert: trajectoire du robot dans le plan et signaux de commande*

Le robot parcourt une distance d'environ 74 m pendant une durée de 30 s, ce qui donne une vitesse moyenne de déplacement de 2.46 m/s. La commande au niveau de l'actionneur linéaire (cf. figure 4.13.(b)) engendre des forces qui ne sont pas trop importantes, elles ne dépassent pas 250 N. Par contre, le couple généré par l'actionneur angulaire de la hanche est assez élevé pendant la phase de contact. Selon les caractéristiques de l'actionneur utilisé, il est envisageable diminuer le gain proportionnel k_p de la commande du comportement du corps afin d'éviter la saturation de l'actionneur.

La commande en trois parties de Raibert a permis de générer des mouvements périodiques stables. C'est un premier pas important dans la commande pour le déplacement planaire des robots sauteurs. Cependant, de nombreux points restent à améliorer. La consigne de vitesse de déplacement n'est pas suivie avec précision, et le contrôle de la hauteur du saut n'est pas satisfaisant. La poussée de la jambe pendant la phase de contact reste constante à chaque saut. En revanche, cette commande ne prend pas en compte les perturbations extérieures et la vitesse de déplacement du robot. Il serait intéressant que celle-ci force le système afin de poursuivre la consigne de hauteur du saut, quel que soit les conditions.

Afin d'améliorer l'approche de commande de Raibert, on propose d'utiliser une technique de commande prédictive non linéaire qui sera présentée dans la section suivante.

4.2.4 Approche 2: Commande prédictive non linéaire

Dans le but de suivre plus efficacement les consignes de vitesse et de hauteur de saut, nous avons décidé d'améliorer la commande de Raibert. L'amélioration concerne le contrôle de la poussée exercée par la jambe avant le décollage. En effet, si l'on commande correctement l'angle de la jambe et la vitesse du robot au moment du décollage, on contrôlera sa trajectoire balistique (cf. figure 4.14) dans la phase de vol. Cette phase étant la plus longue sur un cycle complet de saut, il est essentiel de pouvoir contrôler la vitesse et la position du robot dans cette période. Le contrôle du placement du pied à l'atterrissage et du comportement du corps pendant la phase de contact restent identiques à ceux de la commande de Raibert (voir 4.2.3).

Trajectoire balistique

Dans la phase de vol, il est impossible de commander la trajectoire du robot. Cette trajectoire est déterminée au moment du décollage. La force de gravité exerce une action sur l'accélération verticale du robot. Par conséquent, la composante horizontale de la vitesse du robot $\dot{x}$ ne varie pas pendant la durée de la phase de vol. La vitesse horizontale étant conservée, il faut qu'au décollage, cette vitesse corresponde à la vitesse de déplacement désirée du robot $\dot{x}_{lo} = \dot{x}_d$. La vitesse verticale du robot au décollage $\dot{z}_{lo}$ nécessaire pour sauter à une hauteur z_{ap} désirée est donnée par l'équation 4.3.

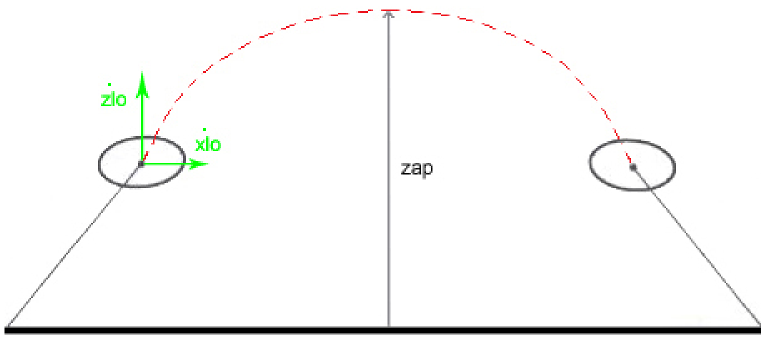

FIG. 4.14 – *Trajectoire du robot pendant la phase de vol*

Nous connaissons désormais, les composantes verticales et horizontales des vitesses du robot au décollage qui permettent de suivre la vitesse de déplacement et la hauteur de saut désirées. Pour suivre ces vitesses, il faut déterminer l'angle et la vitesse de la jambe correspondants. En dérivant les coordonnées du centre du corps du robot au décollage, on obtient le système d'équations non linéaires suivant:

$$\begin{cases} \dot{x}_{lo} = -\dot{l}_d \sin(\upsilon_d) - l\dot{\upsilon}\cos(\upsilon_d) \\ \dot{z}_{lo} = \dot{l}_d \cos(\upsilon_d) - l\dot{\upsilon}\sin(\upsilon_d) \end{cases} \tag{4.9}$$

d'où on peut extraire, l'angle υ_d et la vitesse $\dot{l}_d$ de la jambe que l'on désire suivre au décollage.

Principe de base de la commande prédictive non linéaire

La commande prédictive (appelée aussi commande prédictive à horizon glissant) [4] consiste à trouver une séquence de commande permettant de minimiser un critère de performance J (fonction de coût) :

$$\begin{aligned} &\underset{u}{min} \quad J = \sum_{k=1}^{Np}(x_d - x_k)^T Q(x_d - x_k) + U^T RU \\ &sous\ x_k \in C \end{aligned} \tag{4.10}$$

où l'ensemble C regroupe les contraintes sur l'état le long de l'horizon de prédiction Np. Q et R sont respectivement les pondérations sur les états et les commandes.

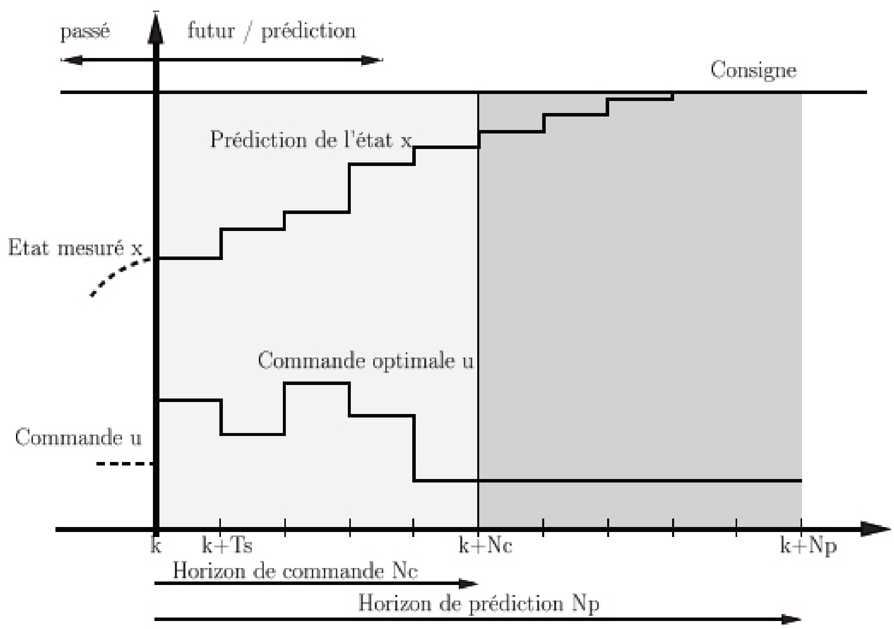

FIG. 4.15 – *Commande prédictive: principe de base*

Dans le cas de problèmes d'optimisation non linéaire sous contraintes (ce qui est notre cas), il n'est pas possible de trouver une solution analytique. Par conséquent, le calculateur doit résoudre numériquement à chaque instant d'échantillonnage ce problème d'optimisation. Il doit déterminer une séquence de commande de taille Nc qui permet de minimiser le critère J sur un horizon de prédiction Np. Cette séquence de commande devra prendre en compte les contraintes fixées au préalable.

L'horizon de commande est généralement inférieur à l'horizon de prédiction [4]. Au terme de l'horizon de commande, celle-ci est maintenu à zéro jusqu'à la fin de l'horizon de prédiction. Quelle que soit la valeur de l'horizon de commande, la séquence de commande optimale est toujours optimisée sur la totalité de l'horizon de prédiction, mais les changements de valeurs sont limités à un horizon plus court.

De la séquence de commande calculée le long de l'horizon de prédiction, seul le premier échantillon sera effectivement appliqué au système. En effet, l'application de cette commande mène le système à un état qui peut être différent de l'état prédit du fait des approximations de modélisation et des inévitables perturbations qui peuvent affecter le système. Pour surmonter ce problème, un bouclage du système est nécessaire (cf. figure 4.16). Cela revient à appliquer la technique de glissement de l'horizon. En effet, après application du premier échantillon de la commande, il faut mesurer l'état courant du système, glisser l'horizon d'une période d'échantillonnage et reposer le même problème d'optimisation en considérant l'état actuel comme étant l'état initial. Cette opération est répétée à chaque instant d'échantillonnage, ce qui va améliorer significativement les performances de l'approche de commande et augmenter sa robustesse.

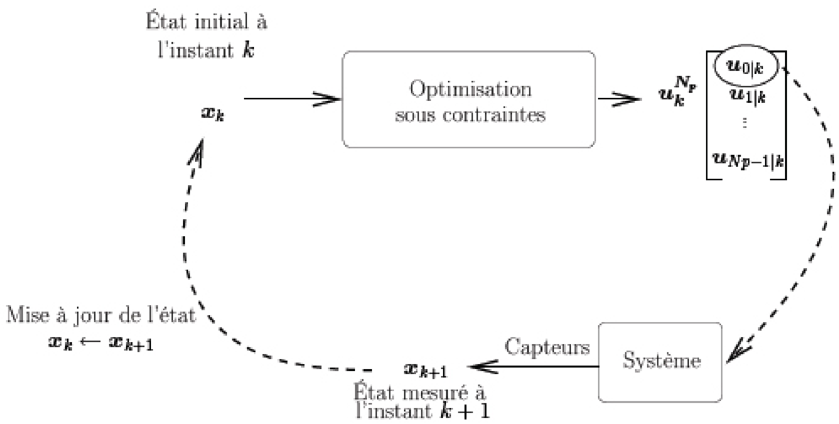

FIG. 4.16 – *Principe de la commande prédictive*

La commande prédictive non linéaire est une commande particulièrement efficace pour les systèmes sous-actionnés contrairement à d'autres approches de commande classiques qui présentes des limitations pour la commande de ce genre de système. En outre, la commande prédictive a montré ses hautes performances en terme de robustesse vis-à-vis des incertitudes et du rejet de perturbation. Elle nécessite cependant, une modélisation particulièrement pointue. L'inconvénient de ce type de commande est le temps de calcul nécessaire à la résolution du problème d'optimisation associé. Celui-ci dépend essentiellement de l'horizon de commande à déterminer et, dans une moindre mesure, de l'horizon de prédiction.

Résultats de simulation

Dans notre cas, on propose de modifier la commande de Raibert en rajoutant la commande prédictive non linéaire pour contrôler la poussée de la jambe. L'approche obtenue est donc constituée des deux contrôleurs de Raibert du positionnement du pied à l'atterrissage et du comportement du corps, puis d'une commande prédictive non linéaire pour contrôler la poussée exercée par la jambe avant le décollage.

La commande obtenue a été intégrée au simulateur du robot sauteur présenté au chapitre 3. Lorsque le robot est en contact avec le sol et que la jambe commence à se détendre, on exerce une force au niveau de l'actionneur de la jambe jusqu'au décollage. Pour contrôler cette poussée on utilise la commande prédictive non linéaire et on minimise le critère de performance suivant:

$$\begin{aligned} &\underset{u}{min} \quad J = \sum_{k=1}^{Np} Q_1(l_0(k) - l(k))^2 + Q_2(\dot{x}_{lo}(k) - \dot{x}(k))^2 + Q_3(\dot{z}_{lo}(k) - \dot{z}(k))^2 \\ &sous \quad |u| < u_{max} \end{aligned} \tag{4.11}$$

Ce critère d'optimisation permet de suivre les vitesses horizontale $\dot{x}_{lo}$ et verticale $\dot{z}_{lo}$ correspondantes à la trajectoire balistique désirée (voir paragraphe 4.2.4). On peut ainsi contrôler la hauteur du saut et la vitesse de déplacement du robot pendant la phase de vol.

Remarque 1 *Pour la simulation de cette commande, nous avons utilisé les mêmes paramètres de modélisation que pour la commande de Raibert. Ce qui va nous permettre de comparer facilement les deux approches de commandes. Les paramètres de la commande prédictive utilisée sont présentés en annexe* C.

Les résultats de simulations obtenus sont présentées sur les figures 4.17, 4.18 et 4.19. Après un régime transitoire d'environ 2 s, la hauteur du saut se stabilise autour de 1.7 m (cf. figure 4.17.(b)). Celle-ci converge vers sa valeur finale beaucoup plus rapidement qu'avec la commande de Raibert (cf. figure 4.11.(b)). La consigne sur la hauteur du saut est relativement mieux suivi puisque l'on constate une erreur statique d'environ 6 %.

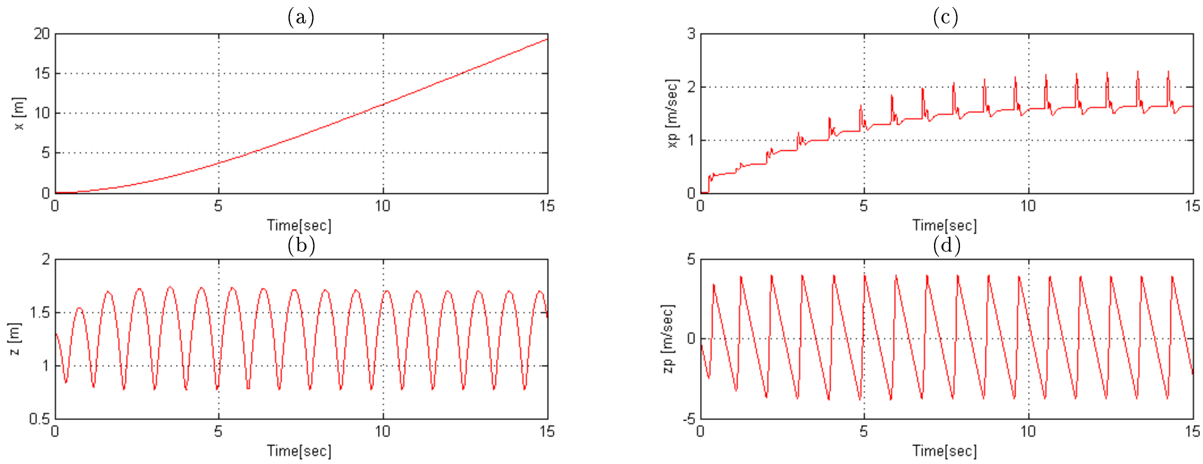

Fig. 4.17 – *Commande predictive non linéaire: évolution des positions et vitesses cartésiennes*

D'autre part, la consigne de la vitesse de déplacement (1 m/s) est beaucoup mieux suivie qu'avec la commande de Raibert (cf. figure 4.11.(c)) puisque l'erreur de consigne sur la vitesse est ramener à 50% alors qu'elle était de 150% avec la commande précédente. Cela se paye par une convergence un peu moins rapide vers le régime permanent puisque l'on passe d'un régime transitoire de 6 secondes avec la commande de Raibert (cf. figure 4.11.(c)) à un régime transitoire de 9 secondes avec la commande prédictive (cf. figure 4.17.(c)).

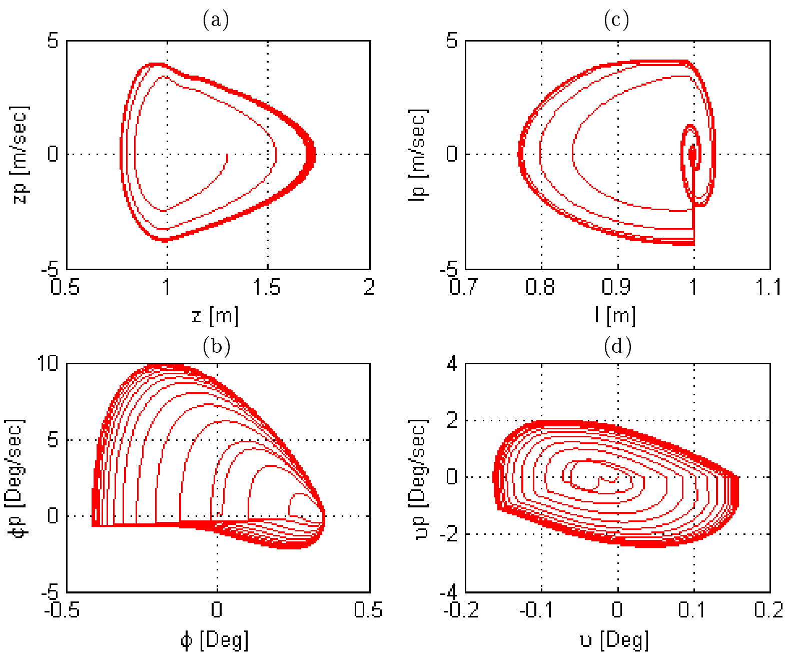

Fig. 4.18 – *Commande predictive non linéaire: plans de phases*

Les plans de phase représentés sur la figure 4.18 mettent en évidence la convergence vers des cycles limites stables. La trajectoire du robot est représentée sur la figure 4.19.(a).

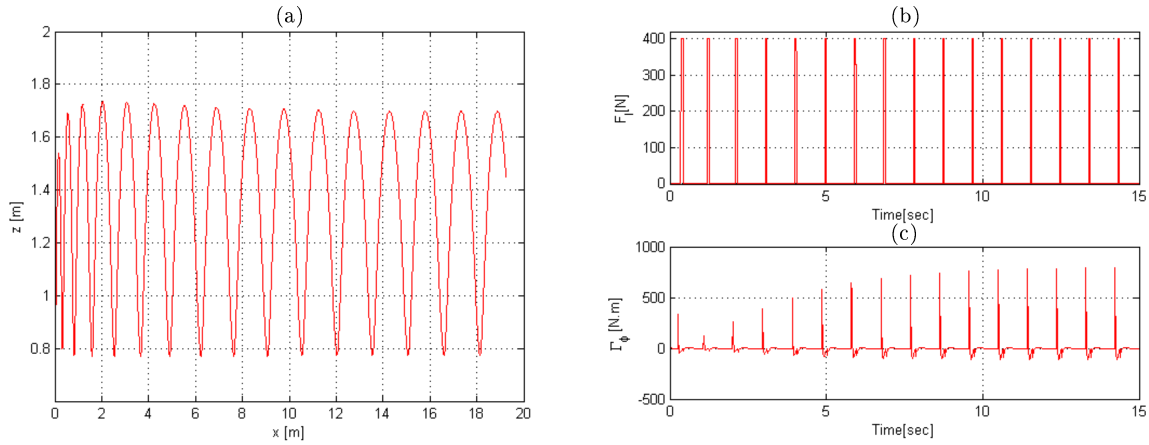

FIG. 4.19 – *Commande predictive non linéaire: trajectoire du robot dans le plan et commandes*

A présent, examinons la commande (cf. figures 4.19.(a) et (b)). Un des avantages de la commande prédictive est de pouvoir prendre en compte les contraintes du système. Ici, nous avons volontairement saturé la commande ($u_{max} = 400\ N$ dans (4.11)). En réalité, la saturation sera déterminée en fonction de l'actionneur utilisé. Cependant, plus celle-ci sera élevée, plus le système convergera rapidement vers son régime permanent (cf. figure 4.19.(a)). Dans notre cas, trois à quatre sauts sont nécessaires pour atteindre la hauteur de saut et la vitesse de déplacement désirées. Afin de montrer la saturation de la commande à u_{max} et la vérification de cette contrainte (imposée dans le critère d'optimisation 4.11)) un zoom effectué sur la figure 4.19.(b), est montré sur la figure 4.20.

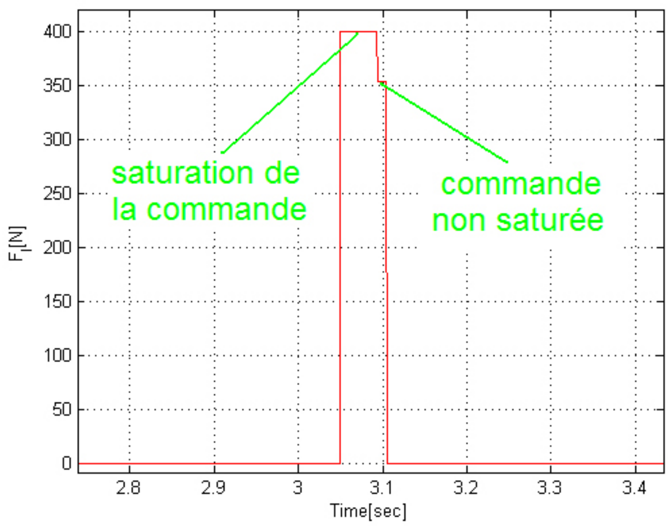

FIG. 4.20 – *Commande predictive non linéaire: zoom sur le signal de commande F_l*

L'utilisation de la commande prédictive pour contrôler la poussée avant le décollage à considérablement amélioré le contrôle global du robot Kangourou. On peut désormais contrôler la hauteur de saut du robot, ce qui était impossible avec la commande de Raibert. Le robot suit avec plus de précision les consignes et converge plus rapidement vers un mouvement cyclique stable. Les erreurs par rapport aux consignes pourraient être sans doute minimisées si l'on annulait les accélérations verticales et horizontales du robot au moment du décollage. Ce point pourrait être pris en considération dans le perfectionnement de cette approche de commande.

Chapitre 5

Conclusions et perspectives

5.1 Conclusion

L'objectif de ce stage était de concevoir des approches de commande pour un robot sauteur planaire unijambiste dans le but générer des mouvements périodiques stables. Pour répondre à cet objectif, le travail réalisé durant ce stage comportait trois volets :

Modélisation : Calcul du modèle dynamique lagrangien du robot sauteur sur les différentes phases du cycle de mouvement.

Commande : Proposition des lois de commande permettant la génération de mouvements périodiques stables.

Simulation : Test et validation en mode simulation des approches de commande proposées pour la commande du robot sauteur.

Plusieurs lois de commandes ont été proposées puis testées à l'aide d'un simulateur développé sous Matlab. Deux approches de commandes ont été proposées pour contrôler le saut vertical du robot: une loi de commande en boucle ouverte et une autre en boucle fermée.

En ce qui concerne la commande du déplacement planaire du robot Kangourou, deux approches de commande ont également été proposées, à savoir la commande de Raibert et la commande prédictive non linéaire.

La commande de Raibert est une référence dans le domaine des robots sauteurs unijambistes. Elle permet un déplacement périodique stable du robot sauteur, cependant elle ne permet pas de contrôler efficacement la vitesse de déplacement et la hauteur de saut du robot.

Cette commande a été améliorée pour mieux répondre à ces nouveaux objectifs. Pour cela, une commande prédictive non linéaire a permis de remplacer une des trois parties du contrôleur de Raibert. Après simulation, cette commande s'est avérée beaucoup plus performante.

Les consignes sont suivies avec plus de précision et la convergence vers un cycle limite stable se fait plus rapidement.

Ce stage a été très enrichissant, tant sur le point technique que sur le plan personnel. Intéressé par la recherche, ce stage m'a permis de découvrir ce milieu et de mieux cerner mon avenir professionnel. Ce fut particulièrement captivant de travailler à la recherche de nouvelles approches de commandes et d'élaborer de nouvelles idées.

5.2 Perspectives

La commande prédictive non linéaire telle qu'elle a été présentée dans la section 4.2.4 pourrait bénéficier de plusieurs améliorations. Dans le but de mieux suivre les consignes sur la hauteur de saut et la vitesse de déplacement, il serait intéressant d'annuler les accélérations du robot au moment du décollage. On pourrait également ajuster les pondérations Q_i associés aux variables d'états dans le critère d'optimisation afin de trouver le meilleur comportement du robot.

Cette loi de commande peut être améliorée vis-à-vis des contraintes liées au temps réel sur ce type de systèmes particulièrement rapides. Dans ce cas, il serait sans doute nécessaire d'adapter certains paramètres de cette commande pour minimiser le temps de calcul du processeur.

Afin de mettre en valeur le travail réalisé dans le cadre de ce stage, un article de conférence est en cours de préparation et qui sera soumis (probablement) à IEEE ICRA2009.

Bibliographie

[1] ARTE 3i Traductions. http://www.3itraductions.fr/3iEN/archive/arte/bio2.htm. *Bionique 2 - Le secret du mouvement*, 2008.

[2] M. Ahmadi and M. Buehler. Stable control of a simulated one-legged running robot with hip and leg compliance. In *IEEE Trans. Robotics and Automation*, 1997.

[3] F. BOISDRON. http://robotimpact.free.fr. *Un micro-robot capable de sauter jusqu'à 27 fois sa taille*, 2008.

[4] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer (2nd Edition), 2007.

[5] N. Cherouvim and E. Papadopoulos. Single actuator control analysis of a planar 3 dof hopping robot. In *Robotics: Science and Systems 2005*, Cambridge, Massachusetts, USA, 2005.

[6] P. Gregorio H. Rad and M.Buehler. Design, modeling and control of a hopping robot. In *IEEE/RSJ Conf. Intelligent Systems and Robots*, pages 1778–1785, Yokohama, Japan, 1993.

[7] K. Harbick and G. Sukhatme. *Height control for a one-legged hopping robot using a one-dimensional model*. Tech. Rep. IRIS-01-406, Institute for Robotics and Intelligent Systems, University of Southern California, USA, 2001.

[8] K. Harbick and G. Sukhatme. Controlling hopping height of a pneumatic monopod. In *IEEE Intl. Conf. on Robotics and Automation*, 2002.

[9] K. Harbick and G. Sukhatme. Robustness experiments for a planar hopping control system. In *International Conference on Climbing and Walking Robots*, 2002.

[10] M. H. Raibert. *Legged Robots That Balance*. MIT Press, 1986.

[11] A. Sato. Simulation of one-legged hopping robot with phase plane stability. In *Proceedings of the 18th conference on Proceedings of the 18th IASTED International Conference: modelling and simulation*, Montreal, Quebec, Canada, 2007.

[12] A. Sato and M. Buehler. A planar hopping robot with one actuator. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Sendai, Japan, 2004.

[13] W. Schwind and D. Koditschek. Control of forward velocity for a simplified planar hopping robot. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, 1995.

[14] M. Spong. Underactuated mechanical systems. *Control Problems in Robotics and Automation, B. Siciliano and K. P. Valavanis Eds., LNCIS, Springer Verlag*, 230:135–150, 1998.

[15] M. Spong and M. Vidyasagar. *Robot Dynamics and Control.* John Wiley & Sons, New York, 1989.

[16] H. D. Taghirad. *Analysis, Design, and Control of Hopping Robot.* Master report, McGill University, Quebec, Canada, 1993.

Annexe A

Détails du modèle dynamique

Description des matrices du modèle dynamique du robot Kangourou écrit sous le formalisme de Lagrange:

$$M(q).\ddot{q} + N(q,\dot{q}).\dot{q} + F(q) + G(q) = U \tag{A.1}$$

Matrice d'inertie $\mathbf{M}(q) =$

$$\begin{pmatrix} m_t + m_l + m_u & 0 & m_u \sin(q_5) & -m_t d_e \sin(\alpha + q_4) & (m_u q_3 + m_l d)\cos(q_5) \\ 0 & m_t + m_l + m_u & -m_u \cos(q_5) & m_t d_e \cos(\alpha + q_4) & (m_u q_3 + m_l d)\sin(q_5) \\ m_u \sin(q_5) & -m_u \cos(q_5) & m_u & 0 & 0 \\ -m_t d_e \sin(\alpha + q_4) & m_t d_e \cos(\alpha + q_4) & 0 & m_t d_e^2 + J_t & 0 \\ (m_u q_3 + m_l d)\cos(q_5) & (m_u q_3 + m_l d)\sin(q_5) & 0 & 0 & J_l + m_l d^2 + m_u {q_3}^2 \end{pmatrix} \tag{A.2}$$

Matrice de Coriolis et des termes centrifuges $\mathbf{N}(q,\dot{q}) =$

$$\begin{pmatrix} 0 & 0 & m_u \cos(q_5)\dot{q}_5 & -m_t d_e \cos(\alpha + q_4)\dot{q}_4 & m_u \cos(q_5)\dot{q}_3 - (m_l d + m_u q_3)\sin(q_5)\dot{q}_5 \\ 0 & 0 & m_u \sin(q_5)\dot{q}_5 & -m_t d_e \sin(\alpha + q_4)\dot{q}_4 & m_u \sin(q_5)\dot{q}_3 + (m_l d + m_u q_3)\cos(q_5)\dot{q}_5 \\ 0 & 0 & 0 & 0 & -m_u q_3 \dot{q}_5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_u q_3 \dot{q}_5 & 0 & m_u q_3 \dot{q}_3 \end{pmatrix} \tag{A.3}$$

Vecteur de friction $\mathbf{F}(q,\dot{q}) =$

$$\begin{pmatrix} 0 \\ 0 \\ -c_l(\dot{p} - \dot{q}_3) \\ c_h(\dot{q}_4 - \dot{q}_5) \\ -c_h(\dot{q}_4 - \dot{q}_5) \end{pmatrix} \tag{A.4}$$

Vecteur de gravité $\mathbf{G}(q) =$

$$\begin{pmatrix} 0 \\ (m_t + m_l + m_u)g \\ -m_u g \cos(q_5) - k_l(s_0 + p + d - q_3) \\ m_t g d_e \cos(\alpha + q_4) \\ (m_l d + m_u q_3) g \sin(q_5) \end{pmatrix} \tag{A.5}$$

Annexe B

Paramètres du modèle passif

Les paramètres du modèle passif du robot sont utilisés dans les simulations du modèle passif seul 3.2, de la commande pour le saut vertical en boucle ouverte 4.1.1 et de celle en boucle fermée 4.1.2. Ces paramètres dynamiques et géométriques sont donnés par:

Symbole	**Description**	**Valeur**
m_t	Masse du corps du robot	12.5 kg
m_l	Masse de la jambe du robot	3 kg
m_u	Masse du pied du robot	0.5 kg
J_t	Moment d'inertie du corps du robot	1 kg/m^2
J_l	Moment d'inertie de la jambe du robot	0.2 kg/m^2
d_e	Distance entre la hanche et le centre de gravité du corps du robot	0.1 m
α	Angle entre l'axe longitudinal du corps du robot et le centre de gravité du corps du robot	0 rad
d	Distance entre la hanche et le centre de gravité de la jambe du robot	0.2 m
p	Distance entre le centre de gravité de la jambe du robot et le point de fixation du ressort	0.25 m
$\dot{p}$	Vitesse de déplacement du point de fixation du ressort par rapport à la hanche (p est fixe)	0 m/s
s_0	Longueur du ressort au repos	1 m
g	Accélération de la gravité	9.81 m/s^2
k_l	Constante de raideur du ressort	5.05 kN/m
c_l	Constante de frottement de la jambe	18
c_h	Constante de frottement de la hanche	6

TAB. B.1 – *Paramètres du modèle passif du robot kangourou*

Annexe C

Simulation: commande Raibert et commande prédictive

Les paramètres de simulation de la commande de Raibert et de la commande prédictive sont:

Symbole	**Description**	**Valeur**
m_t	Masse du corps du robot	12.5 kg
m_l	Masse de la jambe du robot	3 kg
m_u	Masse du pied du robot	0.5 kg
J_t	Moment d'inertie du corps du robot	1 kg/m^2
J_l	Moment d'inertie de la jambe du robot	0.2 kg/m^2
d_e	Distance entre la hanche et le centre de gravité du corps du robot	0 m
α	Angle entre l'axe longitudinal du corps du robot et le centre de gravité du corps du robot	0 rad
d	Distance entre la hanche et le centre de gravité de la jambe du robot	0.2 m
p	Distance entre le centre de gravité de la jambe du robot et le point de fixation du ressort	0.25 m
$\dot{p}$	Vitesse de déplacement du point de fixation du ressort par rapport à la hanche (p est fixe)	0 m/s
s_0	Longueur du ressort au repos	1 m
g	Accélération de la gravité	9.81 m/s^2
k_l	Constante de raideur du ressort	5.05 kN/m
c_l	Constante de frottement de la jambe	18
c_h	Constante de frottement de la hanche	2
k_p	Gain de position du contrôle du comportement du corps	1000
k_v	Gain de vitesse du contrôle du comportement du corps	50
k_P	Gain de position du contrôle de la vitesse de déplacement	200
k_V	Gain de vitesse du contrôle de la vitesse de déplacement	10
$k_{\dot{x}}$	Gain de vitesse du contrôle du déplacement de la position du pied	0.05
$\dot{x}_d$	Vitesse de déplacement du robot désirée	1 m/s
ϕ_d	Position angulaire désirée du corps du robot par rapport à l'horizontale	30 deg

TAB. C.1 – *Paramètres du robot kangourou: commande de Raibert et commande prédictive*

Certains paramètres sont différents d'une commande à l'autre. Ceci est lié aux différences structurelles entre les commandes. Cependant, cela ne nuit pas à leurs comparaisons puisque les paramètres du robot restent identiques, ainsi que les consignes à suivre. Les différences entre les simulations de ces deux commandes sont spécifiées dans les tableaux suivants:

Symbole	**Description**	**Valeur**
F_l	Force exercé par l'actionneur linéaire	240 N

TAB. C.2 – *Spécifications liées à la commande de Raibert*

Symbole	**Description**	**Valeur**
Nu	Horizon de commande	3
Np	Horizon de prédiction	6
T_e	Pas d'échantillonnage de la commande prédictive	0.01 s
$Q1$	Pondération liée à la poursuite de la consigne l_0	1
$Q2$	Pondération liée à la poursuite de la consigne $\dot{x_{lo}}$	1
$Q3$	Pondération liée à la poursuite de la consigne $\dot{z_{lo}}$	1
$haut_d$	Hauteur de saut désirée	1.8 m

TAB. C.3 – *Spécifications liées à la commande prédictive*