

Chapitre 4

Marche bipède

Marcher : Se déplacer par mouvements et appuis successifs des jambes et des pieds sans quitter le sol (*Le Petit Robert*).

Cette définition issue du dictionnaire *Le Petit Robert* est très succincte et nous permet de comprendre qu'au moins un pied reste en contact avec le sol à tout moment de la marche. Les modèles réalisés pour effectuer une marche artificielle s'appuient sur une définition plus étendue qui inclut l'analyse poussée du caractère cyclique ordonné dans les gesticulations du corps. Les études menées en vue d'une expérimentation sur un système humanoïde concernent essentiellement les robots bipèdes : l'analyse se concentre sur les mouvements des jambes. On distingue deux types de marche en robotique : la **marche quasi-statique** et la **marche dynamique**. Dans le cas quasi-statique, la projection du centre de masse sur le sol reste en permanence dans le polygone de sustentation. Le robot restera donc en équilibre s'il s'arrête brutalement. Dans le cas dynamique, la projection du centre de masse sur le sol peut se trouver localement en dehors du polygone de sustentation. Cela traduit le mouvement de chute d'un pied sur l'autre dont nous, humains, sommes familiers. Le robot n'a aucune chance de conserver son équilibre s'il s'arrête brutalement, par exemple juste avant de changer de pied de support.

Marche $\left\{ \begin{array}{l} \text{Marche quasi-statique} \\ \text{Marche dynamique} \end{array} \right.$

Les jouets robotiques se déplacent principalement selon une marche quasi-statique et utilisent des pieds offrant une grande surface de contact avec le sol. Ce cas est peu intéressant du point de vue du contrôle et de la commande, car il est très simple à mettre en œuvre. De plus, les pieds humains présentent une surface au sol trop petite par rapport à la hauteur de leur centre de masse pour permettre une marche statique. Nous utilisons effectivement une marche dynamique dans nos déplacements quotidiens. L'imitation de notre style de marche par un système robotique bipède peut être réalisée moyennant une

habile commande de l'ensemble du corps qui présente la particularité d'être hautement instable. Un robot marcheur bipède se trouve de ce fait en dehors des réalisations de l'ingénierie mécanique conventionnelle. Pour cette raison, de nombreux chercheurs et ingénieurs sont attirés par l'étude des robots marcheurs bipèdes et humanoïdes. L'objet de ce chapitre est d'exposer une méthode de génération de marche dynamique pour ce dernier type de robots.

4.1 Comment réaliser une marche bipède ?

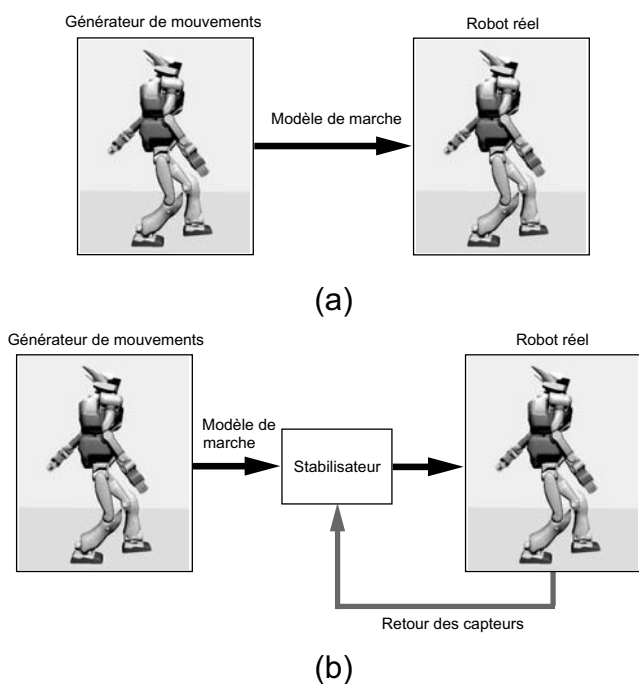


Figure 4.1 – (a) Dans des conditions idéales, un robot bipède peut marcher selon un modèle de marche donné ; (b) dans l'environnement réel, un robot bipède doit utiliser un stabilisateur.

La figure 4.1 illustre le cadre de l'étude sur la commande de la marche de robots bipèdes réalisée dans ce chapitre. L'ensemble des trajectoires articulaires du robot, définies sur des intervalles de temps se succédant pour créer la marche, est appelé *modèle de marche*. Pour créer ce modèle, nous faisons appel à un *générateur de modèle de marche*. Dans une situation idéale, la marche bipède peut être réalisée en fournissant uniquement le modèle de marche au robot (figure 4.1(a)). Afin de réaliser une telle marche dans des conditions idéales, nous devons au préalable disposer d'un modèle précis du robot, sous forme d'un mécanisme rigide se déplaçant exactement selon la commande et sur un

sol parfaitement plat et horizontal (représenté, par exemple, par une grande surface plane).

Dans le cas réel, un robot humanoïde peut facilement tomber à cause d'une irrégularité du sol de quelques millimètres seulement. La distribution de masse et la géométrie des structures humanoïdes créent une amplification rapide des erreurs de posture dues aux irrégularités du support. Il est nécessaire d'implémenter un programme supplémentaire pour éviter cette situation, et qui va modifier le modèle de marche en tenant compte des informations issues du gyroscope, de l'accéléromètre, des capteurs d'efforts et autres dispositifs d'information embarqués sur le robot. Ce programme est appelé *stabilisateur* (figure 4.1(b)).

La suite de ce chapitre est organisée de la manière suivante : dans les parties 4.2, 4.3 et 4.4, nous expliquons le fonctionnement du générateur de modèle. La partie 4.5 est dédiée au fonctionnement du stabilisateur. Nous précisons le contexte historique de la recherche sur les robots bipèdes marcheurs dans la partie 4.6, et décrivons dans la partie 4.7 différentes approches de commande de robots bipèdes utilisées dans un cadre sortant de celui illustré sur la figure 4.1.

4.2 Génération d'une marche en deux dimensions

Ce chapitre expose les principes de base de la marche bipède en deux dimensions et propose un algorithme de génération de modèle.

4.2.1 Le pendule inversé en deux dimensions

Le processus de *coarse-graining* est l'une des méthodes les plus efficaces pour traiter un système complexe. En mécanique céleste, les chercheurs considèrent le Soleil et les planètes comme des masses ponctuelles alors que chacun de ces éléments possède une structure interne propre. Cependant, les orbites suivies dans le système solaire peuvent être calculées avec une grande précision. En thermodynamique, de très grands ensembles de molécules, de l'ordre de 10^{23} éléments, sont soumis au processus de *coarse-graining* pour prédire efficacement les comportements dynamiques en fonction des variations de paramètres comme la température ou l'entropie.

De la même manière, afin de comprendre ce qu'est l'*essence* de la locomotion bipède, nous faisons trois hypothèses sur un robot humanoïde comportant 30 degrés de liberté et constitué de milliers d'éléments mécaniques et électriques. La première hypothèse consiste à considérer le robot comme une masse ponctuelle : l'ensemble des masses des différentes parties qui le constituent se réduisent en son centre de masse. Deuxième hypothèse : nous supposons que le robot possède des jambes sans masse, dont l'extrémité est en contact avec le sol par une liaison rotoïde. Troisièmement, nous ne nous intéressons qu'aux mouvements avant/arrière, haut/bas du robot. Les déplacements latéraux sont

négligés. En d'autres termes, nous supposons que les déplacements du robot sont contraints dans le *plan sagittal* constitué par l'axe de la direction vers l'avant et l'axe vertical. À partir de ces trois hypothèses, nous pouvons représenter le robot comme un pendule inversé en deux dimensions (figure 4.2).

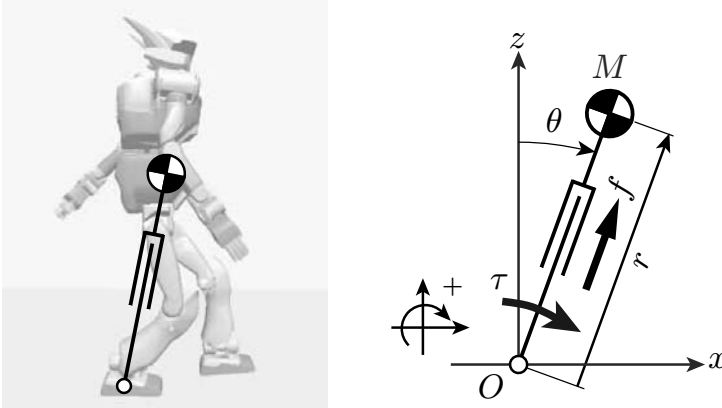


Figure 4.2 – Pendule inversé en deux dimensions : le modèle le plus simple de la marche humaine ou robotique bipède. Il est constitué du centre de masse (CoM) et d'une jambe télescopique sans masse. Nous mesurons l'inclinaison du pendule θ à partir de la verticale, orienté positivement dans le sens trigonométrique indirect.

Dans le modèle du pendule inversé, les entrées sont le couple τ au pivot et la force impulsive f de la liaison prismatique sur la jambe. La dynamique du pendule est décrite par le système de deux équations différentielles suivant¹ :

$$\begin{aligned} r^2\ddot{\theta} + 2r\dot{r}\dot{\theta} - gr \sin \theta &= \tau/M \\ \ddot{r} - r\dot{\theta}^2 + g \cos \theta &= f/M \end{aligned}$$

Le comportement du pendule inversé peut être simulé par intégration numérique de ces équations avec un couple donné en entrée.

Une limite importante à ce modèle est l'impossibilité de donner en entrée un couple τ trop important car la surface de contact entre le pied du robot et le sol est très petite. Si un robot marcheur possède une jambe comme une échasse, il faudra utiliser

$$\tau = 0 \tag{4.1}$$

Dans ce cas, le pendule tombera presque toujours, à moins que la position du CoM soit à la verticale du pivot. Même dans ce cas, il pourra tomber de multiples manières, cela en fonction de la façon dont est contrôlée la force impulsive f . Observons les quelques exemples de chutes illustrés par la figure 4.3.

¹Ces équations peuvent être dérivées en utilisant la méthode de Lagrange. Joseph-Louis Lagrange (1736-1813) était un mathématicien français né en Italie. Il est célèbre notamment pour ses travaux sur les *points de Lagrange*, qui permettent de définir des orbites stables pour les satellites planétaires.

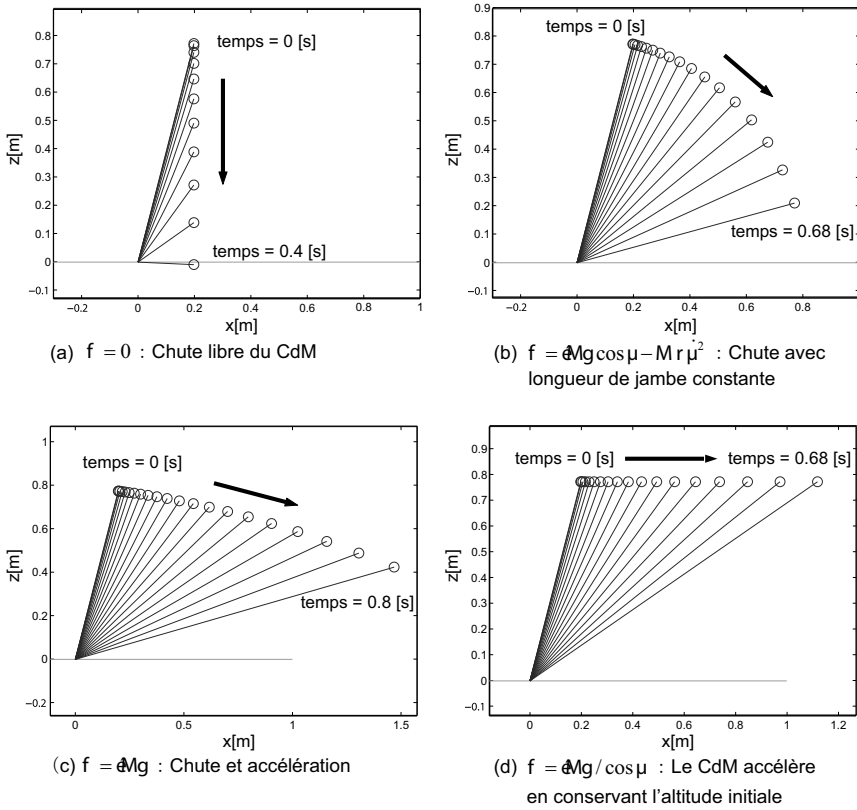


Figure 4.3 – Chute du pendule inversé en fonction de la force impulsive f . Le couple du pivot est maintenu à zéro ($\tau = 0$) durant l'ensemble du mouvement.

Le cas 4.3(d) est le plus intéressant car il permet de voir que le CoM se déplace horizontalement sous la force impulsive

$$f = \frac{Mg}{\cos \theta} \quad (4.2)$$

La raison de cette horizontalité est schématisée sur la figure 4.4 : nous pouvons dire intuitivement que le pendule conserve l'altitude du CoM en étendant sa jambe aussi vite qu'il tombe. Le cas particulier où le pendule génère des entrées qui lui permettent de suivre une ligne droite est appelé **pendule inversé linéaire** (LIP : *Linear Inverted Pendulum* [33])².

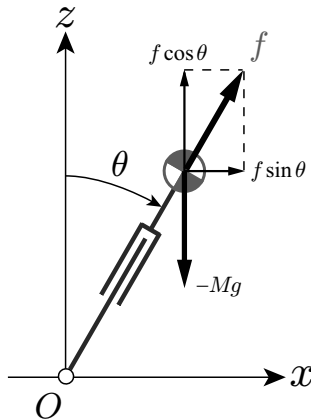


Figure 4.4 – Conservation de l’horizontalité des déplacements du CoM au cours de la chute du pendule inversé. La force impulsive $f = Mg / \cos \theta$ équilibre en permanence les actions de la gravité sur le centre de masse.

4.2.2 Comportement du pendule inversé linéaire (LIP)

Le LIP fournit un outil mathématique très pratique pour le traitement de la dynamique. Commençons par analyser ses déplacements horizontaux, afin de pouvoir en exploiter toutes les caractéristiques pour la génération des modèles de locomotion bipède.

Dynamique horizontale

Un second regard sur la figure 4.4 permet d’observer que seule la composante horizontale de la force d’impulsion est active, alors que la composante verticale est équilibrée par les forces de gravité. La composante horizontale crée une accélération horizontale \ddot{x} du CoM définie par

$$M\ddot{x} = f \sin \theta \quad (4.3)$$

En remplaçant la valeur de f par celle définie dans l’équation 4.2, nous obtenons

$$M\ddot{x} = \frac{Mg}{\cos \theta} \sin \theta = Mg \tan \theta = Mg \frac{x}{z}$$

où x et z définissent la position du CoM dans le plan sagittal. L’équation ci-dessus fournit une équation différentielle caractérisant la dynamique horizontale du CoM :

$$\ddot{x} = \frac{g}{z} x \quad (4.4)$$

²La définition que nous en donnons ici représente la forme la plus simple du pendule inversé linéaire. Le concept qu’il représente sera développé tout au long de ce chapitre.

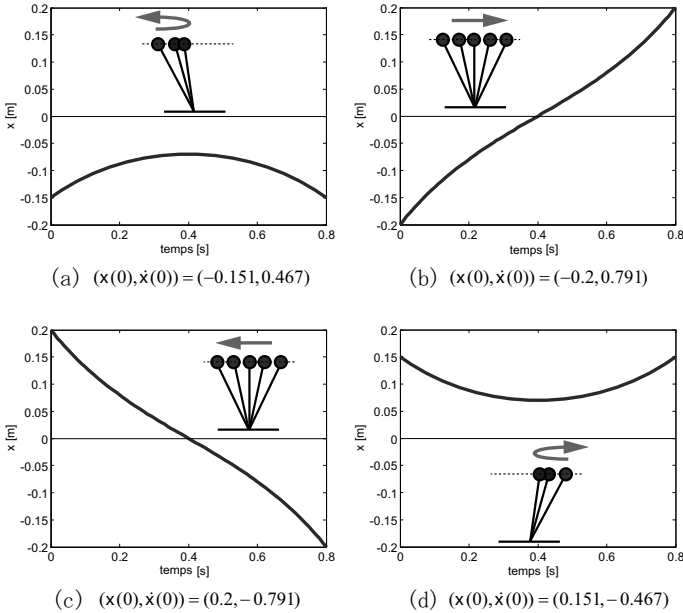


Figure 4.5 – LIP utilisant différentes conditions initiales ($z = 0.8$ m).

Comme la hauteur z du pendule inversé est constante au cours de ce mouvement, il est très facile de résoudre l'équation 4.4.

$$x(t) = x(0) \cosh(t/T_c) + T_c \dot{x}(0) \sinh(t/T_c) \quad (4.5)$$

$$\dot{x}(t) = \dot{x}(0) \cosh(t/T_c) + x(0)/T_c \sinh(t/T_c) \quad (4.6)$$

où T_c est une constante de temps qui ne dépend que de la hauteur du centre de masse et de l'accélération de la gravité :

$$T_c \equiv \sqrt{z/g}$$

$x(0)$ et $\dot{x}(0)$ représentent respectivement une position et une vitesse initiales. Ces deux valeurs constituent l'état initial, ou encore les conditions initiales. La figure 4.5 montre les mouvements du pendule répondant à différentes conditions initiales.

Temps de transfert

Dans beaucoup de cas, nous aurons besoin de connaître le temps mis par le centre de masse pour évoluer d'un point à un autre. Si des conditions initiales (x_0, \dot{x}_0) et un état (x_1, \dot{x}_1) à l'instant t sont donnés, les différentes variables sont liées par les équations suivantes, basées sur les résultats des équations 4.5

et 4.6 :

$$x_1 = x_0 \cosh(\tau/T_c) + T_c \dot{x}_0 \sinh(\tau/T_c) \quad (4.7)$$

$$\dot{x}_1 = x_0/T_c \sinh(\tau/T_c) + \dot{x}_0 \cosh(\tau/T_c) \quad (4.8)$$

où τ est le temps mis par le centre de masse pour se déplacer de (x_0, \dot{x}_0) à (x_1, \dot{x}_1) . En utilisant les relations $\cosh(x) = \frac{e^x + e^{-x}}{2}$ et $\sinh(x) = \frac{e^x - e^{-x}}{2}$, nous pouvons récrire les équations précédentes sous la forme

$$x_1 = \frac{x_0 + T_c \dot{x}_0}{2} e^{\tau/T_c} + \frac{x_0 - T_c \dot{x}_0}{2} e^{-\tau/T_c} \quad (4.9)$$

$$\dot{x}_1 = \frac{x_0 + T_c \dot{x}_0}{2T_c} e^{\tau/T_c} - \frac{x_0 - T_c \dot{x}_0}{2T_c} e^{-\tau/T_c} \quad (4.10)$$

Pour simplifier cette écriture et extraire une relation exploitable, nous effectuons l'opération $4.9 + T_c \times 4.10$. Nous obtenons alors

$$x_1 + T_c \dot{x}_1 = (x_0 + T_c \dot{x}_0) e^{\tau/T_c}$$

Nous pouvons à présent exprimer la valeur de τ en fonction de l'état initial, de l'état à l'instant t et de la constante de temps T_c :

$$\tau = T_c \ln \frac{x_1 + T_c \dot{x}_1}{x_0 + T_c \dot{x}_0}. \quad (4.11)$$

Le temps de transfert τ peut donc être calculé, sous la condition $x_0 + T_c \dot{x}_0 \neq 0$. Sinon, nous procédons de la même manière que précédemment pour extraire une seconde équation, à partir de l'opération $4.9 - T_c \times 4.10$, et nous obtenons :

$$\tau = T_c \ln \frac{x_0 - T_c \dot{x}_0}{x_1 - T_c \dot{x}_1}. \quad (4.12)$$

4.2.3 Énergie orbitale

Pour comprendre intuitivement les notions physiques impliquées par le LIP, imaginons un champ de potentiel imaginaire ayant la forme d'une colline, comme illustré sur la figure 4.6. Le pendule se déplace sur la pente de la colline, pour la monter ou en descendre. La figure 4.6(a) montre le cas où les déplacements du COM changent de direction car la vitesse initiale est insuffisante pour atteindre et passer le sommet de la colline. Dans le cas de la figure 4.6(b), la direction de déplacement initiale du COM est conservée : la vitesse initiale est suffisante pour vaincre les forces de pesanteur, et sur l'ensemble du mouvement la vitesse est minimale au sommet du potentiel. Cette position correspond au passage à la verticale du pivot fixant le pendule au support.

Pour extraire une relation entre le potentiel imaginaire et les déplacements du centre de masse, nous multiplions les deux côtés de l'équation du mouvement

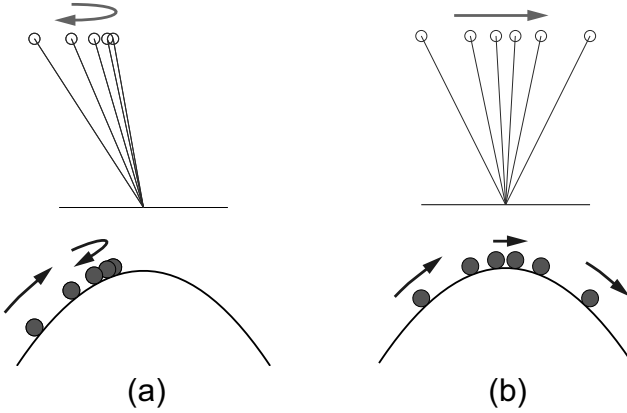


Figure 4.6 – LIP et potentiel imaginaire.

(équation 4.4) par \dot{x} , puis l'intégrons :

$$\begin{aligned} \dot{x}(\ddot{x} - \frac{g}{z}x) &= 0 \\ \int \{\dot{x}\ddot{x} - \frac{g}{z}x\dot{x}\} dt &= \text{Constante} \end{aligned}$$

Le résultat fournit

$$\frac{1}{2}\dot{x}^2 - \frac{g}{2z}x^2 = \text{Constante} \equiv E \tag{4.13}$$

Le premier terme du membre de gauche représente l'énergie cinétique et le second est l'énergie du potentiel imaginaire illustré par la figure 4.6. Ce calcul considère les énergies par unités de masse, de ce fait la masse du CoM n'apparaît pas dans les expressions.

Nous appelons *énergie orbitale*, notée E , la somme de l'énergie cinétique et de l'énergie du potentiel imaginaire³. L'équation 4.13 montre que **l'énergie orbitale est conservée lors des mouvements du LIP**.

Dans le cas de la figure 4.6(a) où le CoM ne parvient pas à passer le sommet du potentiel, l'énergie orbitale vaut

$$E = -\frac{g}{2z}x_{apex}^2 \tag{4.14}$$

où x_{apex} est la position horizontale atteinte par le CoM au moment où sa vitesse devient nulle. Dans ce cas, l'énergie orbitale est négative ou nulle.

Dans le cas de la figure 4.6(b) où le CoM parvient à passer le sommet du potentiel, l'énergie orbitale vaut

$$E = \frac{1}{2}\dot{x}_{apex}^2 \tag{4.15}$$

³En mécanique, cette valeur est connue comme une constante du mouvement [17].

où $\dot{x}_{apex} (> 0)$ est la vitesse à laquelle le CoM passe le sommet du potentiel. L'énergie orbitale est alors positive.

Si nous relevons la position et la vitesse de la masse du pendule inversé à un instant donné, et calculons l'énergie orbitale E grâce à l'équation 4.13, nous pouvons immédiatement savoir si le CoM va passer le sommet du potentiel ou non en regardant le signe de E . Si $E > 0$, la vitesse absolue au moment du passage vaut :

$$|\dot{x}_{apex}| = \sqrt{2E}$$

Si $E < 0$, nous pouvons prédire la position absolue à laquelle la vitesse du CoM devient nulle :

$$|x_{apex}| = \sqrt{-\frac{2zE}{g}}$$

4.2.4 Changement de la jambe de support

Les mouvements du LIP sont essentiellement déterminés par les conditions initiales. Il est cependant possible de contrôler les mouvements du pas suivant en choisissant astucieusement la durée du changement de support, celle-ci influençant les conditions initiales. Regardons l'illustration de la figure 4.7. Cette situation est celle que nous utilisons lorsque nous ramenons rapidement la jambe sur le sol lors d'un arrêt en urgence.

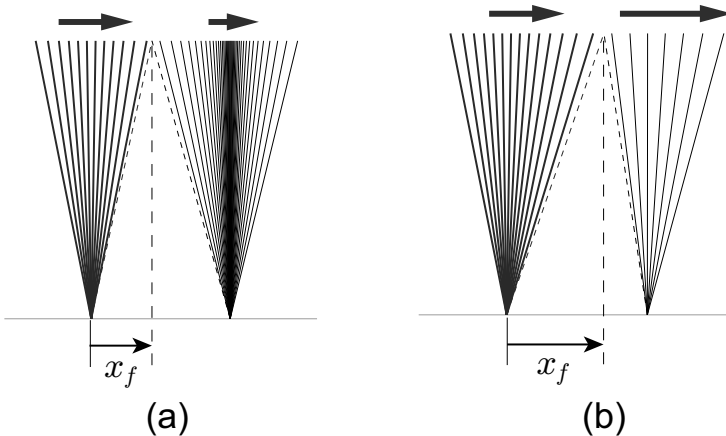


Figure 4.7 – Contrôle de la vitesse de marche à longueur de pas fixe : (a) une reprise de contact rapide lors du pas suivant provoque un ralentissement de l'allure de marche ; (b) une reprise de contact tardive lors du pas suivant provoque une accélération de l'allure de marche.

Nous allons établir la relation entre la rapidité du changement de support et les mouvements du pendule. La figure 4.8 modélise le pendule double au moment du changement de support. Nous notons s la longueur du pas, x_f la

position du CoM par rapport au point de contact précédent, et v_f la vitesse du CoM au moment du changement. Pour simplifier l'étude, nous utilisons l'hypothèse que le changement de la jambe de support est effectué instantanément. La vitesse du CoM à la fin d'une phase de support est donc aussi celle du début de la phase de support suivante.

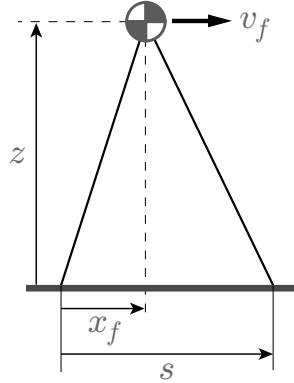


Figure 4.8 – État du système au moment du changement de support.

Notons E_1 l'énergie orbitale avant le changement et E_2 l'énergie orbitale après le changement. Les expressions de E_1 et E_2 sont les suivantes :

$$E_1 = -\frac{g}{2z}x_f^2 + \frac{1}{2}v_f^2 \quad (4.16)$$

$$E_2 = -\frac{g}{2z}(x_f - s)^2 + \frac{1}{2}v_f^2 \quad (4.17)$$

Quand les énergies orbitales E_1 et E_2 sont données, nous pouvons déterminer les conditions nécessaires pour que le changement soit réalisable en supprimant v_f des équations 4.16 et 4.17.

$$x_f = \frac{z}{gs}(E_2 - E_1) + \frac{s}{2} \quad (4.18)$$

La vitesse, au moment du changement, est calculée grâce à l'équation 4.16.

$$v_f = \sqrt{2E_1 + \frac{g}{z}x_f^2} \quad (4.19)$$

4.2.5 Planification d'une allure de marche élémentaire

Nous allons par la suite définir un modèle élémentaire de marche en utilisant les résultats des calculs précédents. Considérons un robot bipède idéal, avançant d'un seul pas sur un sol plat horizontal, puis s'arrêtant immédiatement (figure 4.9). Pour ce faire, le robot change deux fois de jambe de support et trois énergies orbitales doivent être définies.

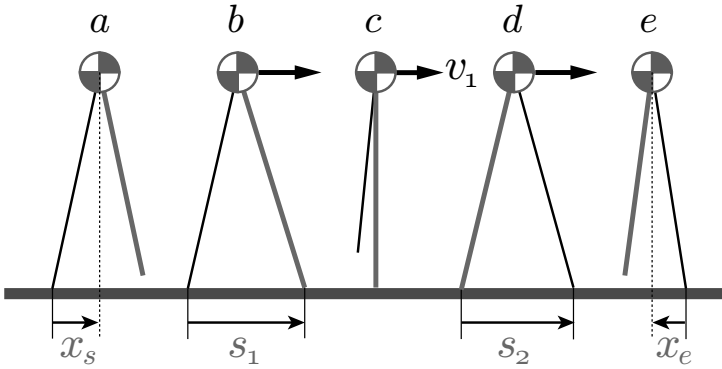


Figure 4.9 – Le pas nécessite deux changements de jambe de support.

Pour le départ de la marche (a→b), l'énergie orbitale est spécifiée par la position initiale du CoM :

$$E_0 = -\frac{g}{2z}x_s^2$$

Pour le pas (b→c→d), l'énergie orbitale est spécifiée par la vitesse v_1 au moment où le CoM passe à la verticale du point de support.

$$E_1 = \frac{1}{2}v_1^2$$

Pour la fin de la marche (d→e), l'énergie orbitale est spécifiée par la position finale du CoM.

$$E_2 = -\frac{g}{2z}x_e^2$$

L'équation 4.18 fournit la première condition pour le changement de la jambe de support. En effet, x_{f0} est obtenu par la connaissance de E_0 et E_1 . La deuxième condition, x_{f1} , est obtenue par la même équation en utilisant les valeurs de E_1 et E_2 . Le mouvement de marche désiré est réalisé en contrôlant le mouvement de balancier de la jambe qui n'est pas en contact avec le sol, de telle sorte que les deux conditions de changement de support soient satisfaites.

La vitesse de marche au moment du changement de support est calculée grâce à l'équation 4.19. L'état complet du système est donc connu sur l'ensemble du mouvement de marche. La figure 4.10 donne le profil de la trajectoire et de la vitesse du CoM au cours de la succession des pas. Sur la courbe de position, l'origine est donnée par la position au moment des changements de support. Cela explique les discontinuités observées, par exemple, au moment des échanges en b et d . La valeur du saut de discontinuité en ces points est fixée par la longueur des pas. La courbe des vitesses met en évidence les variations dynamiques au cours du mouvement par des pics de discontinuité au moment des changements de support.

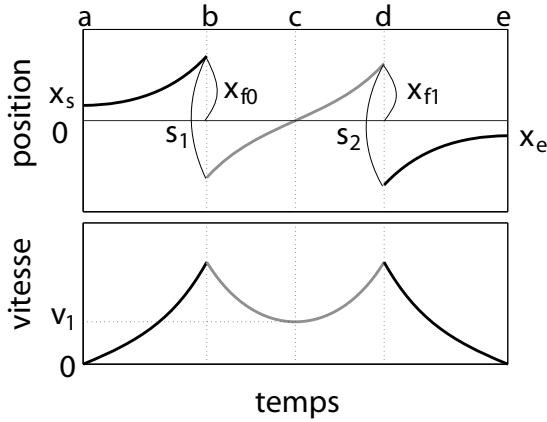


Figure 4.10 – Prédiction des mouvements du centre de masse lors de la marche (position et vitesse).

4.2.6 Extension à la marche sur sol non horizontal

La méthode expliquée précédemment est applicable essentiellement sur un sol plat horizontal. On utilise cependant une méthode similaire dans le cas de sols en pente : nous allons montrer dans ce qui suit quelles modifications apporter au modèle élémentaire précédent pour gérer cette différence.

Reprenons le modèle du pendule inversé de la figure 4.2. Nous allons considérer cette fois que le centre de masse se déplace sur une ligne droite inclinée représentée sur la figure 4.11. L'équation de la droite s'écrit

$$z = kx + z_c \quad (4.20)$$

où k définit la pente de la droite, z_c son intersection avec l'axe des z . Cette droite de déplacement du CoM est appelée *droite de contrainte*.

Nous allons calculer la force de propulsion f nécessaire pour réaliser le mouvement désiré du centre de masse. Dans un premier temps, exprimons les composantes horizontale f_x et verticale f_z de \mathbf{f} .

$$f_x = f \sin \theta = (x/r)f \quad (4.21)$$

$$f_z = f \cos \theta = (z/r)f \quad (4.22)$$

Pour que le CoM se déplace le long de la droite de contrainte, celle-ci doit être parallèle à la droite portant le vecteur résultant de la somme de la force de propulsion et des forces de gravité. Cela est expliqué par la relation suivante :

$$\frac{f_x}{f_z - Mg} = \frac{1}{k} \quad (4.23)$$

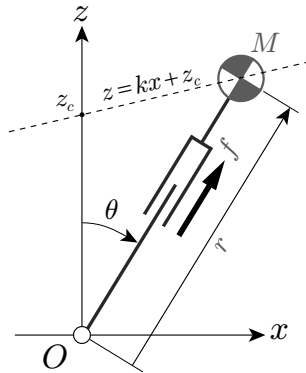


Figure 4.11 – Le centre de masse est contrôlé pour se déplacer selon la droite de contrainte (en pointillés) par la force de propulsion f . Les chevilles tournent librement ($\tau = 0$).

Nous obtenons l'expression de f en substituant les relations 4.21 et 4.22 dans l'équation 4.23 :

$$f = \frac{Mgr}{z - kx} \quad (4.24)$$

ou en utilisant l'équation de contrainte 4.20 :

$$f = \frac{Mgr}{z_c} \quad (4.25)$$

Cette relation montre que le CoM se déplace sur une droite de contrainte lorsqu'on le soumet à une force de propulsion f proportionnelle à la longueur de la jambe r . Pour planifier ces mouvements, il est nécessaire de connaître la condition initiale de la contrainte (z_c). Quelques résultats possibles sont illustrés par la figure 4.12, sur laquelle les flèches indiquent direction et amplitude de la force de propulsion nécessaire.

Étudions les variations dynamiques résultant de ces mouvements : dans le plan horizontal, la dynamique du centre de masse s'obtient en reportant l'équation 4.25 dans 4.21, avec $f_x = M\ddot{x}$.

$$\ddot{x} = \frac{g}{z_c} x \quad (4.26)$$

La relation obtenue est semblable à celle de l'équation 4.4 issue de la dérivation des mouvements du CoM sur le plan horizontal. Pour expliquer cette observation, regardons sur la figure 4.13 le résultat de deux simulations s'appuyant sur des conditions initiales de contrainte identiques et des pentes de signes opposés.

Les deux droites de contrainte ont la même intersection z_c , et le pendule commence à la même position horizontale x_0 . Nous pouvons raisonnablement nous attendre à ce que le pendule utilisant la droite ascendante ($k > 0$) ralentisse au cours du mouvement, et que celui utilisant la pente descendante

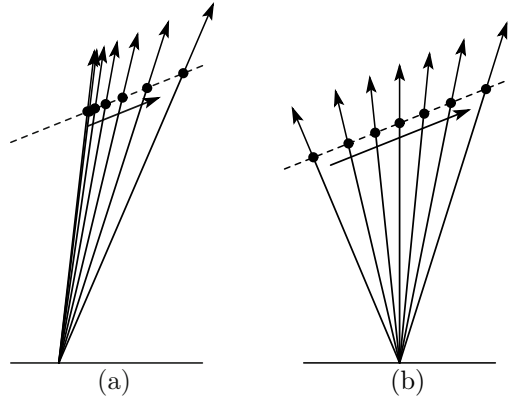


Figure 4.12 – Vecteur force de propulsion et déplacement pendulaire selon une droite de contrainte.

($k < 0$) accélère, du fait de l'influence des forces de gravité. Or, il n'en est rien : les déplacements du pendule dans le plan horizontal sont exactement les mêmes dans les deux cas. Ainsi, comme le montre la figure 4.13, les deux pendules atteignent la position x_1 simultanément. Cela s'explique par le fait que les forces de gravité sont compensées par la force de propulsion et que les mouvements du pendule ne sont définis que par la position horizontale du CoM.

La figure 4.14 illustre la planification d'une trajectoire du centre de masse, en utilisant la méthode décrite ci-dessus, appliquée au cas d'une montée d'escalier. Dans un premier temps, nous avons fixé des points de changement de support (triangles) compatibles avec les contraintes mécaniques du robot. Par la suite, nous avons défini les droites de contraintes en reliant les points situés à une hauteur z_c sur la verticale des points d'appuis. La commande de la trajectoire du CoM le contraint à évoluer le long de ces droites de contraintes. Nous pouvons exprimer la dynamique du CoM pour chaque pas :

$$\ddot{x} = \frac{g}{z_c}$$

puis appliquer la méthode de planification. Le second schéma de la figure 4.14 montre que la vitesse horizontale de déplacement du CoM n'est pas affectée par la montée des escaliers.

Cette méthode permet d'établir des modèles de marche pour différents types de sols. Par exemple, les auteurs de [35] ont développé un robot bipède simple (figure 4.15) qui peut monter des escaliers et enjamber des obstacles en utilisant la perception en temps réel.

4.3 Génération d'une marche en trois dimensions

La génération du modèle de marche en trois dimensions fait appel au LIP-3D. Nous allons donc étendre la méthode du LIP aux déplacements en trois

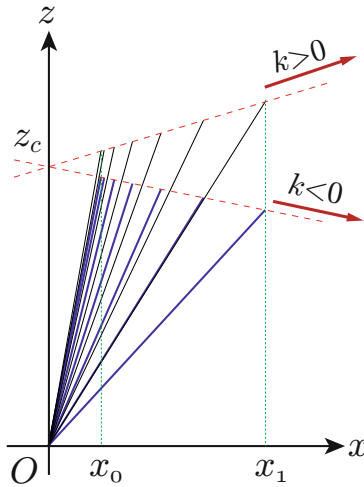


Figure 4.13 – Les mouvements horizontaux d'un LIP ne dépendent pas de la pente de la droite de contrainte.

dimensions, puis l'appliquer à un cas de génération d'un modèle de marche tridimensionnel.

4.3.1 LIP-3D

Nous allons modéliser un robot bipède marcheur dans l'espace 3D par un pendule inversé comme illustré sur la figure 4.16. Le robot est représenté par une masse ponctuelle positionnée en son centre de masse, une jambe sans masse établissant la connexion entre la masse ponctuelle et le sol au point de contact. Nous admettons que le pendule peut tourner librement autour de son point de support et que la longueur de la jambe peut être modifiée par l'intermédiaire de la force de propulsion f . Exprimons f selon les trois composantes x , y et z :

$$f_x = (x/r)f \quad (4.27)$$

$$f_y = (y/r)f \quad (4.28)$$

$$f_z = (z/r)f \quad (4.29)$$

où r représente la distance entre le point de support et le centre de masse (figure 4.11). Les seules forces agissant sur le centre de masse sont la force de propulsion et la gravité. Nous pouvons établir les équations du mouvement régissant les déplacements du CoM :

$$M\ddot{x} = (x/r)f \quad (4.30)$$

$$M\ddot{y} = (y/r)f \quad (4.31)$$

$$M\ddot{z} = (z/r)f - Mg \quad (4.32)$$

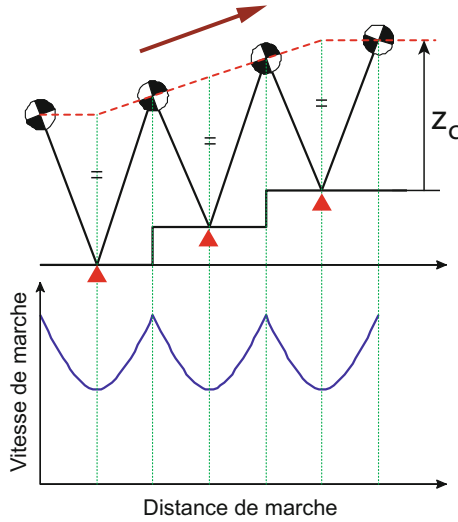


Figure 4.14 – Montée d’escaliers basée sur un LIP, avec définition des droites de contraintes (en haut) et vitesse horizontale du centre de masse (en bas).

Comme dans le cas en deux dimensions, nous établissons une contrainte de déplacement du CoM. Dans le cas 3D, nous ne parlerons pas d’une droite de contrainte mais d’un *plan de contrainte* défini de la manière suivante :

$$z = k_x x + k_y y + z_c \tag{4.33}$$

où k_x et k_y caractérisent la pente, et z_c est l’altitude à laquelle le plan coupe l’axe des z (figure 4.17).

Afin d’assurer un mouvement du CoM selon le plan de contrainte, l’accélération du centre de masse doit rester orthogonale à la normale au plan. Nous avons donc

$$\left[f \left(\frac{x}{r} \right) \quad f \left(\frac{y}{r} \right) \quad f \left(\frac{z}{r} \right) - Mg \right] \begin{bmatrix} -k_x \\ -k_y \\ 1 \end{bmatrix} = 0 \tag{4.34}$$

Réolvons cette équation vectorielle pour obtenir f . Pour cela, nous nous aidons des résultats de l’équation 4.33, et nous obtenons :

$$f = \frac{Mgr}{z_c} \tag{4.35}$$

Le centre de masse se déplace selon le plan de contrainte en appliquant une force de propulsion f proportionnelle à la longueur de la jambe r . La figure 4.17 illustre cette situation.

La dynamique horizontale du centre de masse est obtenue à partir des équations 4.30 et 4.31, en remplaçant la valeur de la force de propulsion par la valeur

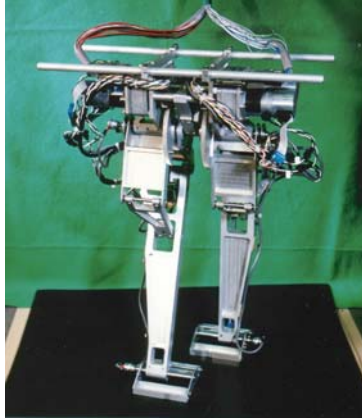


Figure 4.15 – Meltran II, robot bipède dont la géométrie des jambes est inspirée de l’oiseau. Le faible poids des jambes permet de considérer le robot comme un pendule inversé.

donnée à l’équation 4.35 :

$$\ddot{x} = \frac{g}{z_c} x \quad (4.36)$$

$$\ddot{y} = \frac{g}{z_c} y \quad (4.37)$$

C’est un système d’équations linéaires où z_c est le seul paramètre. Les paramètres d’inclinaison k_x et k_y du plan de contrainte n’affectent pas le déplacement horizontal du CoM puisqu’ils n’apparaissent pas dans les équations 4.36 et 4.37. Ce type de pendule est appelé **pendule inversé linéaire 3D**⁴ (LIP-3D).

4.3.2 Caractéristiques du LIP-3D

Le LIP-3D est un mélange entre deux modèles de LIP-2D et ses caractéristiques sont très intéressantes. La figure 4.18 montre différentes trajectoires de CoM, suivant des modèles de pendule inversé linéaire 3D définis par le même paramètre z_c , mais des pentes différentes.

Sur cette figure, nous pouvons observer des trajectoires centrées sur l’intersection $[0 \ 0 \ z_c]$ du plan avec l’axe des z présentant une ressemblance impressionnante avec celle d’une comète⁵. La dynamique de ces trajectoires en trois

⁴Le modèle du LIP-3D a été initialement proposé par Hara, Yokogawa et Sadao [19].

⁵Les trajectoires LIP sont différentes de celles, par exemple, des comètes qui tournent autour du soleil. Une analogie plus proche serait celle de la trajectoire de particules dans l’expérience de dispersion de Rutherford qui a mis en évidence l’existence d’un noyau atomique dans la particule. Nous verrons par la suite que cette analogie est effectivement très proche des phénomènes observés.

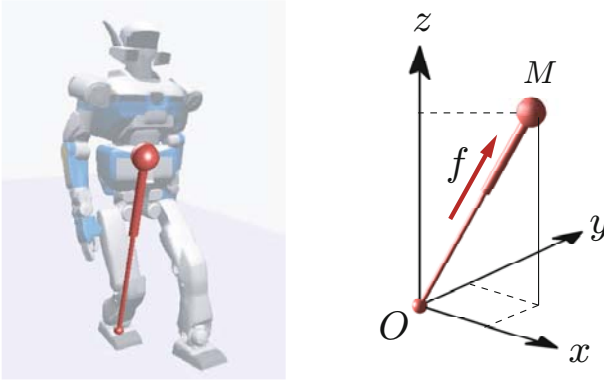


Figure 4.16 – Modélisation d'un robot en marche par un LIP-3D. Le point de support est représenté par une liaison sphérique permettant une rotation libre. La longueur de la jambe est modifiée moyennant une force de propulsion f .

dimensions, en projection sur le plan horizontal xy (figure 4.18(b)), peut être abordée en négligeant l'inclinaison du plan de contrainte.

La seconde loi de Kepler

Partant d'une analogie avec la mécanique céleste, nous allons vérifier la seconde loi de Kepler : une planète a une vitesse aréolaire constante. La vitesse aréolaire v_{area} est, par définition, la dérivée de l'aire balayée par le segment joignant un point mobile à l'origine des coordonnées. Par unité de temps, elle s'exprime de la manière suivante :

$$v_{area} = \frac{1}{2}(x\dot{y} - \dot{x}y) \quad (4.38)$$

Dans le cas du LIP-3D, le point mobile est la masse ponctuelle du pendule. La dérivée de la vitesse aréolaire est

$$\begin{aligned} \frac{d}{dt}(v_{area}) &= \frac{1}{2}(\dot{x}\dot{y} + x\ddot{y} - \ddot{x}y - \dot{x}\dot{y}) \\ &= \frac{1}{2}(x\ddot{y} - \ddot{x}y) \quad \leftarrow \text{(utilisation des équations 4.36 et 4.37)} \\ &= \frac{1}{2}\left(x\frac{g}{z_h}y - \frac{g}{z_h}xy\right) \\ &= 0 \end{aligned}$$

La vitesse aréolaire du LIP-3D est donc bien constante puisque sa dérivée dans le temps est nulle⁶.

⁶D'une manière générale, la vitesse aréolaire est conservée au cours d'un déplacement dans un champ de force. En d'autres termes, cela représente la conservation du moment cinétique.

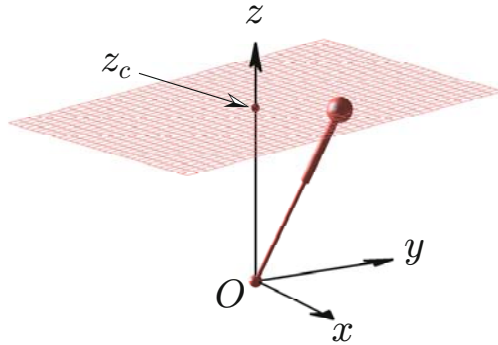


Figure 4.17 – LIP-3D : le centre de masse se déplace le long du plan de contrainte en choisissant une force de propulsion adaptée. L'inclinaison du plan de contrainte n'influe pas sur les déplacements horizontaux du CoM.

Rotation du repère de référence

Considérons le cas où le repère de référence xy subit une rotation d'un angle θ autour de son centre. Nous notons $x'y'$ le nouveau repère obtenu après transformation (figure 4.19). La transformation est définie de la manière suivante :

$$x = cx' - sy' \quad (4.39)$$

$$y = sx' + cy' \quad (4.40)$$

$$c \equiv \cos \theta, s \equiv \sin \theta$$

Nous allons montrer que les équations 4.36 et 4.37 du LIP-3D sont toujours valables dans le nouveau repère $x'y'$. En substituant les transformations de coordonnées ci-dessus dans les équations du mouvement, nous obtenons :

$$c\ddot{x}' - s\ddot{y}' = \frac{g}{z_h}(cx' - sy') \quad (4.41)$$

$$s\ddot{x}' + c\ddot{y}' = \frac{g}{z_h}(sx' + cy'). \quad (4.42)$$

La dynamique du système, exprimée dans le repère $x'y'$, s'obtient en isolant \ddot{x}' et \ddot{y}' dans les équations 4.41 et 4.42 :

$$\ddot{x}' = \frac{g}{z_c}x'$$

$$\ddot{y}' = \frac{g}{z_c}y'$$

Cette transformation est toujours valable quel que soit l'angle de rotation θ . Ce fait confirme que la dynamique du LIP-3D peut toujours être exprimée par deux composantes orthogonales *indépendamment du système de référence*⁷.

⁷Attention : les déplacements dus aux forces de gravité ou aux forces électrostatiques sont inversement proportionnels au carré de la distance, et ne peuvent donc pas être traités de la

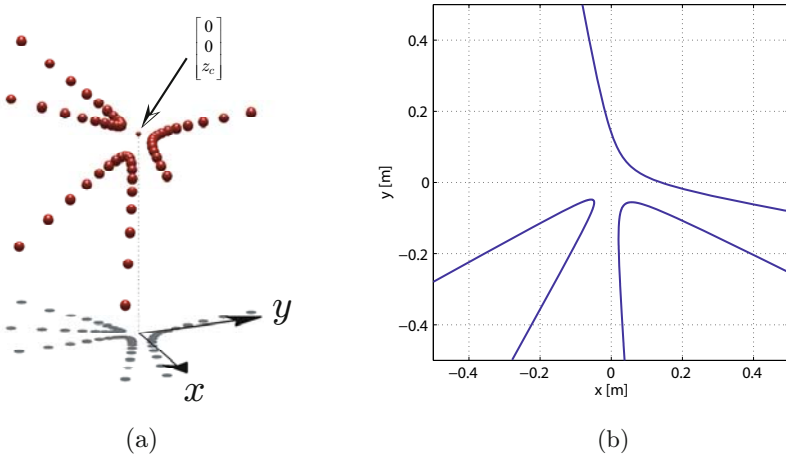


Figure 4.18 – Déplacements du LIP-3D : (a) trajectoires du CoM dans l’espace 3D ; (b) projections des trajectoires sur le plan horizontal.

Étude géométrique de la trajectoire du LIP-3D

Nous pouvons calculer la trajectoire géométrique grâce à la transformation illustrée par la figure 4.19 et aux énergies orbitales. L’énergie orbitale selon l’axe x' du nouveau repère est

$$E'_x = -\frac{g}{2z_c}(cx + sy)^2 + \frac{1}{2}(c\dot{x} + s\dot{y})^2 \tag{4.43}$$

E'_x est dépendante de l’angle d’orientation θ du repère, et prend des valeurs extrêmes lorsque x' ou y' correspond à l’axe de symétrie de la trajectoire.

$$\begin{aligned} \frac{\partial E'_x}{\partial \theta} &= -A \cos 2\theta + \frac{B}{2} \sin 2\theta = 0 \tag{4.44} \\ A &\equiv (g/z_c)xy - \dot{x}\dot{y} \\ B &\equiv (g/z_c)(x^2 - y^2) - (\dot{x}^2 - \dot{y}^2) \end{aligned}$$

La valeur de θ indiquant un axe de symétrie de la trajectoire s’obtient à partir de l’équation 4.44 :

$$\theta = \begin{cases} (1/2) \tan^{-1}(2A/B) & (\text{si } B \neq 0) \\ \pi/4 & (\text{si } A \neq 0, B = 0) \end{cases} \tag{4.45}$$

Dans le cas où $A = 0$ et $B = 0$, la trajectoire devient une ligne droite passant par l’origine et orientée de manière quelconque : cette ligne est un axe de symétrie.

même manière !

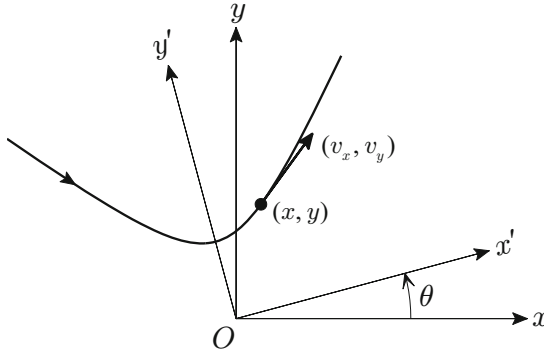


Figure 4.19 – Projection sur le plan horizontal d'une trajectoire LIPM 3D et rotation d'un angle θ du repère de référence.

Si nous choisissons le repère xy de telle sorte que nous soyons sur un axe de symétrie, alors l'énergie orbitale E'_x prend des valeurs extrêmes avec $\theta = 0$. En substituant $\theta = 0$ dans l'équation 4.44, nous obtenons

$$(g/z_c)xy - \dot{x}\dot{y} = 0$$

Soit, en élevant cette égalité au carré :

$$(g/z_c)^2 x^2 y^2 = \dot{x}^2 \dot{y}^2$$

Afin d'obtenir une équation ne dépendant que des paramètres de la trajectoire x et y , nous remplaçons les valeurs de \dot{x} et \dot{y} par leur expression issue de la définition de l'énergie orbitale :

$$\dot{x}^2 = 2E_x + \frac{g}{2z_c}x^2 \quad \dot{y}^2 = 2E_y + \frac{g}{2z_c}y^2$$

Nous obtenons en fin de compte une équation qui représente la forme géométrique de la trajectoire du LIP-3D.

$$\frac{g}{2z_c E_x} x^2 + \frac{g}{2z_c E_y} y^2 + 1 = 0 \quad (4.46)$$

C'est l'équation d'une hyperbole car l'un des termes E_x ou E_y est négatif et l'autre est positif⁸.

⁸Des trajectoires hyperboliques sont aussi obtenues lors du calcul de Rutherford (*scattering*) ou de la gravitéviation (manœuvre utilisant l'attraction d'un corps céleste pour modifier le vecteur vitesse d'un engin spatial) pour l'approche de Jupiter par Voyager I et II. Il est très intéressant de retrouver des similarités dans l'expression des trajectoires de particules élémentaires, de voyages planétaires et de locomotion bipède.

4.3.3 Génération des trajectoires de marche

La figure 4.20 montre un exemple de modèle de marche basé sur le LIP-3D. En modifiant judicieusement le choix du plan de contrainte, le même modèle peut être utilisé pour des déplacements dans un escalier ou sur un sol en pente.

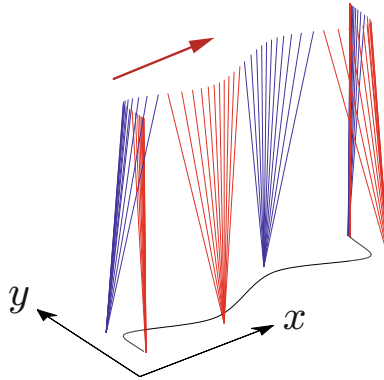


Figure 4.20 – Modèle de marche sur sol plat utilisant la méthode du LIP-3D : avancement de trois pas entre deux positions immobiles.

Dans le cas de la marche en trois dimensions, il est nécessaire de générer un changement de support simultané dans les directions x et y . Nous ne pouvons donc pas utiliser la méthode de génération de modèles de marche exposée dans la partie 4.2.5 car celle-ci nécessite un changement de support à des instants arbitraires. Dans la discussion qui suit, nous considérons une marche dont la longueur du pas est constante, et nous notons T_{sup} le temps de support de chaque pas.

Primitive de marche

Une *primitive* est, d'une manière générale, un élément de base pouvant servir à des constructions complexes. Nous allons définir une portion de trajectoire, appelée *primitive de marche*, par la méthode du LIP-3D (figure 4.21).

Cette trajectoire primitive, montrée sur la figure 4.21(a), est une portion d'hyperbole symétrique par rapport à l'axe y et définie sur l'intervalle de temps $[0 \ T_{sup}]$.

Lorsque le temps de support T_{sup} et l'intersection du plan de contrainte z_c sont donnés, une primitive de marche est déterminée de façon unique (car symétrique) par ses coordonnées extrêmes (\bar{x}, \bar{y}) . La vitesse finale (\bar{v}_x, \bar{v}_y) peut alors être calculée, et nous allons détailler son calcul.

Si la condition initiale selon l'axe x est notée $(-\bar{x}, \bar{v}_x)$, la position finale est \bar{x} en utilisant la nature symétrique de la primitive de marche. À partir de la solution analytique du LIP-3D donnée à l'équation 4.5, on a

$$\bar{x} = -\bar{x}C + T_c \bar{v}_x S \quad (4.47)$$

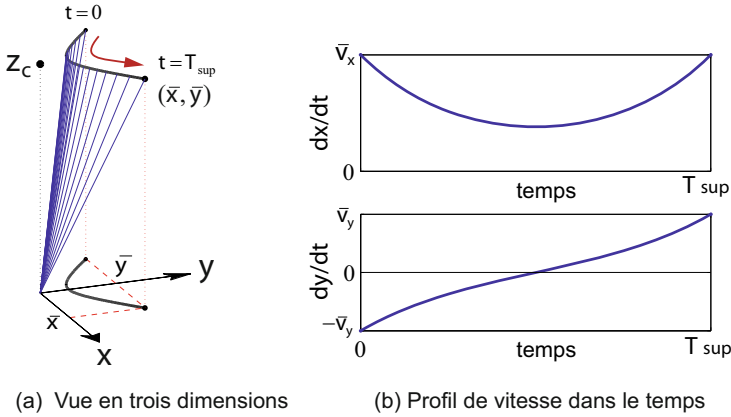


Figure 4.21 – Primitive de marche : un modèle 3D élémentaire.

où

$$T_c \equiv \sqrt{\frac{z_c}{g}}, \quad C \equiv \cosh \frac{T_{sup}}{T_c}, \quad S \equiv \sinh \frac{T_{sup}}{T_c}.$$

La résolution de l'équation 4.47 permet d'obtenir la vitesse finale \bar{v}_x selon l'axe x :

$$\bar{v}_x = \bar{x}(C + 1)/(T_c S) \quad (4.48)$$

Nous utilisons une méthode semblable pour déterminer la composante en y de la vitesse finale, notée \bar{v}_y : à partir de la condition initiale $(\bar{y}, -\bar{v}_y)$ selon l'axe y , la position finale est \bar{y} par symétrie. Ainsi,

$$\begin{aligned} \bar{y} &= \bar{y}C + T_c(-\bar{v}_y)S \\ \bar{v}_y &= \bar{y}(C - 1)/(T_c S) \end{aligned} \quad (4.49)$$

Les primitives de marche permettent de réaliser très facilement une trajectoire de marche. Par exemple, une marche en ligne droite, de longueur de pas constante et égale à $2\bar{x}$, peut être réalisée en connectant des primitives de marches identiques en alternant successivement le signe de y .

Paramètres de la marche

Dans beaucoup de cas pratiques, il est nécessaire de fixer les positions successives des pieds pour pouvoir déterminer la trajectoire de déplacement correspondante. Un déplacement dans des escaliers ou l'enjambement d'un obstacle constituent deux bons exemples de ces situations. La figure 4.22 montre un cas de marche classique illustrant bien le choix des paramètres. Nous utilisons comme entrées les longueurs et largeurs des pas suivantes :

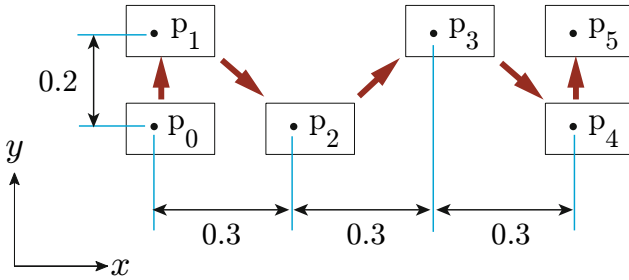


Figure 4.22 – Points de référence de placement des pieds $p_0 \dots p_N$. Les rectangles dessinés autour des points représentent les empreintes des pieds du robot.

n	1	2	3	4	5
$s_x^{(n)}$	0.0	0.3	0.3	0.3	0
$s_y^{(n)}$	0.2	0.2	0.2	0.2	0.2

où s_x représente la longueur du pas dans le sens de la marche et s_y la largeur du pas selon la normale au déplacement. Les données regroupées dans ce tableau sont appelées *paramètres de marche*. L'indice (n) réfère aux données du n -ième pas, et les caractéristiques du n -ième positionnement de pied $(p_x^{(n)}, p_y^{(n)})$ sont définies par

$$\begin{bmatrix} p_x^{(n)} \\ p_y^{(n)} \end{bmatrix} = \begin{bmatrix} p_x^{(n-1)} + s_x^{(n)} \\ p_y^{(n-1)} - (-1)^n s_y^{(n)} \end{bmatrix} \tag{4.50}$$

Nous avons considéré arbitrairement le pied droit comme support de départ, et $(p_x^{(0)}, p_y^{(0)})$ représente l'emplacement du premier point de support. Si nous choisissons le pied gauche comme point de départ de la marche, il faut alors remplacer $-(-1)^n$ par $+(-1)^n$ dans l'équation précédente.

La primitive de marche correspondant au n -ième pas s'écrit

$$\begin{bmatrix} \bar{x}^{(n)} \\ \bar{y}^{(n)} \end{bmatrix} = \begin{bmatrix} s_x^{(n+1)}/2 \\ (-1)^n s_y^{(n+1)}/2 \end{bmatrix} \tag{4.51}$$

Les caractéristiques de la n -ième primitive sont donc fixées par les $(n+1)$ -ièmes paramètres de marche. C'est une condition nécessaire pour assurer une bonne coordination entre le placement des pieds et le mouvement de marche.

Les composantes en x et y de la vitesse finale d'une primitive de marche sont obtenues par les équations 4.48 et 4.49 :

$$\begin{bmatrix} \bar{v}_x^{(n)} \\ \bar{v}_y^{(n)} \end{bmatrix} = \begin{bmatrix} (C + 1)/(T_c S) \bar{x}^{(n)} \\ (C - 1)/(T_c S) \bar{y}^{(n)} \end{bmatrix} \tag{4.52}$$

Les primitives utilisées pour générer la marche ci-dessus sont toutes identiques, excepté la première et la dernière qui correspondent aux états de transition de

démarrage et d'arrêt. Nous allons maintenant décrire une méthode qui permet de générer des modèles de marche continus et réalisables.

Modification du placement des pieds

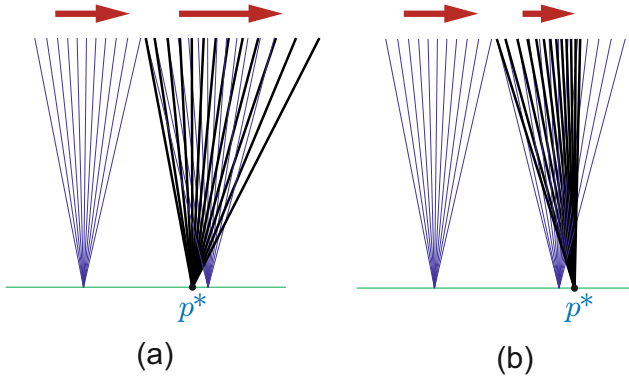


Figure 4.23 – Modification de la vitesse de marche [51] : (a) accélération par le raccourcissement du pas; (b) ralentissement par l'allongement du pas.

Dans le cas d'un cycle de marche donné, la modification de la longueur des pas peut permettre de contrôler la vitesse de la marche⁹, comme illustré sur la figure 4.23.

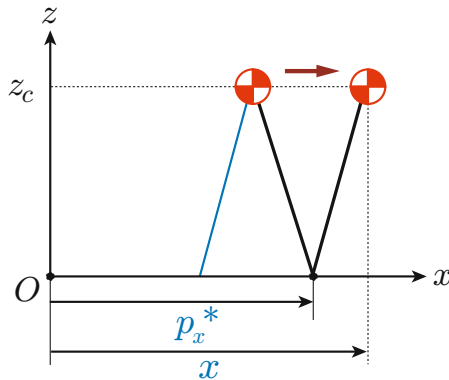


Figure 4.24 – Représentation d'un LIP dans le repère de référence.

Soit p_x^* le positionnement modifié du pied selon la direction de marche. Étudions les variations dynamiques du mouvement dues à ce changement : la

⁹Dans la méthode de contrôle de la figure 4.7 (page 104), la longueur du pas était fixée. La vitesse de marche était alors contrôlée en modifiant l'instant de reprise de contact avec le sol.

figure 4.24 représente un pendule inversé linéaire dans le repère fixe lié à la terre¹⁰. L'équation de la dynamique selon l'axe x s'écrit

$$\ddot{x} = \frac{g}{z_c}(x - p_x^*) \quad (4.53)$$

et la solution analytique à l'équation de la dynamique est

$$x(t) = (x_i^{(n)} - p_x^*) \cosh(t/T_c) + T_c \dot{x}_i^{(n)} \sinh(t/T_c) + p_x^* \quad (4.54)$$

$$\dot{x}(t) = \frac{x_i^{(n)} - p_x^*}{T_c} \sinh(t/T_c) + \dot{x}_i^{(n)} \cosh(t/T_c), \quad (4.55)$$

où $(x_i^{(n)}, \dot{x}_i^{(n)})$ est la condition initiale du n -ième pas. Nous pouvons donc extraire une relation entre le placement modifié du pied p_x^* et l'état final $(x_f^{(n)}, \dot{x}_f^{(n)})$ du n -ième pas :

$$\begin{bmatrix} x_f^{(n)} \\ \dot{x}_f^{(n)} \end{bmatrix} = \begin{bmatrix} C & T_c S \\ S/T_c & C \end{bmatrix} \begin{bmatrix} x_i^{(n)} \\ \dot{x}_i^{(n)} \end{bmatrix} + \begin{bmatrix} 1 - C \\ -S/T_c \end{bmatrix} p_x^* \quad (4.56)$$

étape 1 Déterminer la période de support T_{sup} et les paramètres de marche s_x, s_y .

Donner les positions initiales du CoM (x, y) et du pied $(p_x^*, p_y^*) = (p_x^{(0)}, p_y^{(0)})$.

étape 2 $T := 0, n := 0$.

étape 3 Intégrer l'équation 4.53 du pendule inversé et l'équation selon l'axe y de T à $T + T_{sup}$.

étape 4 $T := T + T_{sup}, n := n + 1$

étape 5 Calculer la position suivante $(p_x^{(n)}, p_y^{(n)})$ du pied en utilisant l'équation 4.50.

étape 6 Déterminer les caractéristiques de la primitive de marche suivante, $(\bar{x}^{(n)}, \bar{y}^{(n)})$, en utilisant les équations 4.51 et 4.52.

étape 7 Calculer l'état cible (x^d, \dot{x}^d) par l'équation 4.57, et l'état cible (y^d, \dot{y}^d) par l'équation correspondante.

étape 8 Calculer la position modifiée du pied (p_x^*, p_y^*) par l'équation 4.59 (ainsi que la composante y).

étape 9 Retourner à l'**étape 3**.

Figure 4.25 – Algorithme de génération d'un modèle de marche basé sur la méthode du LIP-3D.

Nous choisissons comme état cible l'état final de la primitive de marche dans le repère lié au sol.

$$\begin{bmatrix} x^d \\ \dot{x}^d \end{bmatrix} = \begin{bmatrix} p_x^{(n)} + \bar{x}^{(n)} \\ \bar{v}_x^{(n)} \end{bmatrix} \quad (4.57)$$

¹⁰Nous abordons seulement les modifications selon l'axe d'avancement de la marche x . Les modifications selon l'axe y se traitent de manière identique.

Calculons le placement de pied correspondant à l'état final le plus proche de l'état désiré (x^d, \dot{x}^d) . La fonction d'évaluation peut être définie par

$$N \equiv a(x^d - x_f^{(n)})^2 + b(\dot{x}^d - \dot{x}_f^{(n)})^2 \tag{4.58}$$

où a et b sont des poids positifs. En reportant les valeurs de l'équation 4.56 dans la fonction d'évaluation, nous obtiendrons le placement de pied qui minimise N en appliquant la condition $\partial N / \partial p_x^* = 0$:

$$\begin{aligned} p_x^* &= -\frac{a(C-1)}{D}(x^d - Cx_i^{(n)} - T_c S \dot{x}_i^{(n)}) \\ &\quad - \frac{bS}{T_c D}(\dot{x}^d - \frac{S}{T_c}x_i^{(n)} - C\dot{x}_i^{(n)}) \\ D &\equiv a(C-1)^2 + b(S/T_c)^2 \end{aligned} \tag{4.59}$$

La méthode qui génère le modèle de marche est décomposée en neuf étapes successives reportées sur l'algorithme de la figure 4.25. Le résultat est illustré par la figure 4.26 en utilisant les paramètres de marche précédents (figure 4.22). Observez les premières et dernières positions de pied proposées : la première est légèrement en retrait sur l'axe x par rapport au placement de départ pour le premier pas (précédant l'accélération), et la dernière est légèrement en avant juste avant l'arrêt (précédant la décélération). Ces modifications sont inévitables : la méthode ne permet pas le respect formel des paramètres de marche désirés. L'équation 4.59 garantit cependant une convergence de l'erreur vers zéro, et le robot peut suivre les spécifications des paramètres au cours d'une marche établie, c'est-à-dire en dehors périodes d'accélération et de décélération (ou plus généralement des périodes de transitions).

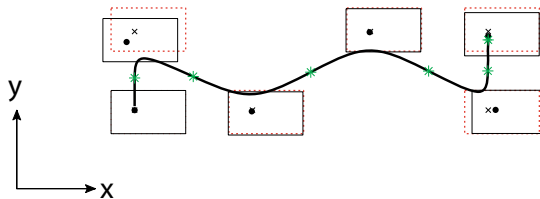


Figure 4.26 – Modèle de marche généré par l'algorithme avec $z_c = 0.8$, $T_{sup} = 0.8$, $a = 10$, $b = 1$. La ligne représente la trajectoire du CoM, les croix sont les positions spécifiées, les points sont les positions modifiées. Les positions sont modifiées au départ et à l'arrêt de la marche pour obtenir les accélérations et décélérations adaptées.

Connaissant la technique de base pour générer un modèle de marche simple, nous pouvons maintenant jouer avec les paramètres pour obtenir différents types de marche. Par exemple, pour une marche en diagonal comme celle de la figure 4.27, la valeur de s_y est modifiée à chaque pas :

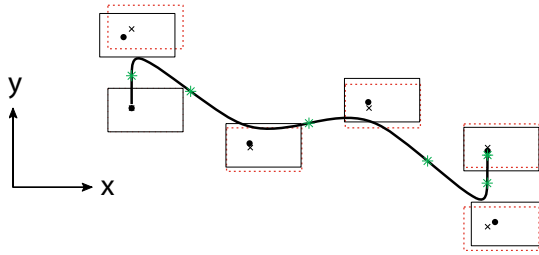


Figure 4.27 – Exemple de marche diagonale avec $z_c = 0.8$, $T_{sup} = 0.8$, $a = 10$ et $b = 1$. La valeur de s_y est modifiée à chaque pas.

n	1	2	3	4	5
$s_x^{(n)}$	0.0	0.2	0.2	0.2	0
$s_y^{(n)}$	0.2	0.3	0.1	0.3	0.2

Si nous fixons tous les paramètres s_x à zéro, nous obtiendrions alors un modèle de marche en crabe. Voyons maintenant comment faire tourner le robot pendant la marche.

Changement de direction

Pour changer la direction de la marche, nous avons besoin d’ajouter une information sur la direction dans les paramètres de marche. Soit s_θ la direction d’un pas, comme illustré sur la figure 4.28¹¹.

Le placement du pied $(p_x^{(n)}, p_y^{(n)})$ pour le n -ième pas est déterminé par

$$\begin{bmatrix} p_x^{(n)} \\ p_y^{(n)} \end{bmatrix} = \begin{bmatrix} p_x^{(n-1)} \\ p_y^{(n-1)} \end{bmatrix} + \begin{bmatrix} \cos s_\theta^{(n)} & -\sin s_\theta^{(n)} \\ \sin s_\theta^{(n)} & \cos s_\theta^{(n)} \end{bmatrix} \begin{bmatrix} s_x^{(n)} \\ -(-1)^n s_y^{(n)} \end{bmatrix} \quad (4.60)$$

La primitive de marche correspondante pour le n -ième pas est la suivante :

$$\begin{bmatrix} \bar{x}^{(n)} \\ \bar{y}^{(n)} \end{bmatrix} = \begin{bmatrix} \cos s_\theta^{(n+1)} & -\sin s_\theta^{(n+1)} \\ \sin s_\theta^{(n+1)} & \cos s_\theta^{(n+1)} \end{bmatrix} \begin{bmatrix} s_x^{(n+1)}/2 \\ (-1)^n s_y^{(n+1)}/2 \end{bmatrix} \quad (4.61)$$

La vitesse d’avancement de la primitive de marche est définie par

$$\begin{bmatrix} \bar{v}_x^{(n)} \\ \bar{v}_y^{(n)} \end{bmatrix} = \begin{bmatrix} \cos s_\theta^{(n+1)} & -\sin s_\theta^{(n+1)} \\ \sin s_\theta^{(n+1)} & \cos s_\theta^{(n+1)} \end{bmatrix} \begin{bmatrix} (1 + C)/(T_c S) \bar{x}^{(n)} \\ (C - 1)/(T_c S) \bar{y}^{(n)} \end{bmatrix} \quad (4.62)$$

L’utilisateur peut générer un modèle de marche avec une direction arbitraire en remplaçant l’équation 4.50 de l’étape 5 de l’algorithme de génération de

¹¹La définition que nous proposons dans ce livre est un peu plus simple que celle que nous utilisons réellement. Par exemple, la méthode donnée ne peut pas générer un tournant sur place. Déterminer une définition plus élaborée est un exercice pour le lecteur.

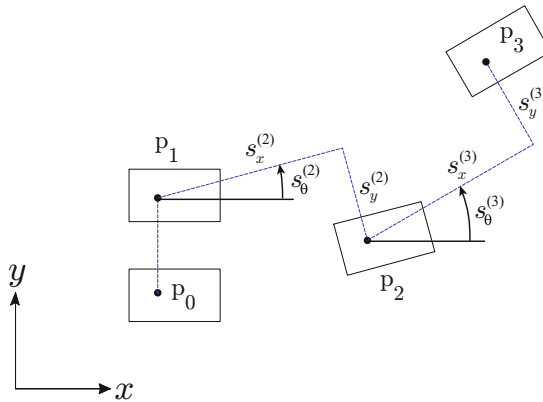


Figure 4.28 – Placement de pied avec changement de direction de marche. s_θ est mesuré à partir de l'axe x positivement dans le sens trigonométrique direct.

modèles par l'équation 4.60, et en remplaçant les équations 4.51 et 4.52 de l'étape 6 par les équations 4.61 et 4.62.

Par exemple, pour générer une marche selon un arc de cercle, nous pouvons utiliser les paramètres du tableau suivant, qui spécifient une rotation de 20° par pas.

n	1	2	3	4	5
s_x	0.0	0.25	0.25	0.25	0
s_y	0.2	0.2	0.2	0.2	0.2
s_θ	0	20	40	60	60

La figure 4.29 montre la trajectoire de marche correspondant à ces paramètres.

4.3.4 Introduction de la phase de double support

Nous avons considéré jusqu'ici que le changement de support s'effectuait instantanément. Dans ce cas, la position du ZMP « saute » du pied de support précédent au nouveau, et donc l'accélération passe d'un maximum en un minimum (figure 4.30(a)). Si nous utilisons cette approche, la discontinuité de l'accélération crée des forces d'impact très importantes qui pourraient endommager le robot.

Pour obtenir une marche plus fluide, et surtout adaptée à un robot réel, nous allons insérer une phase de double support avec une période T_{dbl} prédéterminée au moment du changement de la jambe de support.

Par l'introduction de la phase de double support, nous désirons établir un profil de vitesse sans brusques changements de pente, cela afin de supprimer les discontinuités de l'accélération du CoM. Dans cet objectif, nous générerons un

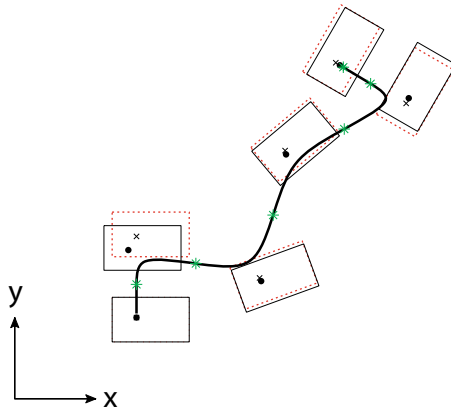


Figure 4.29 – Marche selon un arc de cercle. Chaque pas subit une rotation de 20° . $z_c = 0.8$, $T_{sup} = 0.8$, $a = 10$ et $b = 1$.

profil de vitesses du CoM en utilisant des polynômes du troisième degré. La trajectoire du CoM est donc décrite par un polynôme de degré 4 (figure 4.30(b)) :

$$x(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 \quad (4.63)$$

Les coefficients $a_0 \dots a_4$ sont déterminés par les conditions aux limites sur la position, la vitesse et l'accélération du CoM au moment du changement de support.

Par l'insertion d'une phase de double support, le robot fait des pas plus grands que prévu. Cela peut être rétabli en réduisant proportionnellement par avance les primitives de marche. La figure 4.31 illustre un exemple d'une trajectoire de marche avec phase de double support.

En faisant des essais de paramétrages, on peut remarquer qu'une période de double support plus longue permet une trajectoire plus lisse du CoM lors du changement de support. Cependant, cela crée aussi un phénomène indésirable de balancement plus rapide de la jambe. Il y a donc un compromis à trouver lors de la détermination de T_{dbl} .

4.3.5 Du pendule linéaire inversé au modèle multicorps

La méthode la plus facile pour générer un modèle de marche par le LIP est de faire suivre à la hanche les mouvements LIP du CoM. Premièrement, la position réelle du CoM est calculée en utilisant le modèle multicorps. Pendant cette opération, la position du CoM par rapport à la hanche est déterminée. Par la suite, la position de la hanche est déterminée directement par le pendule inversé linéaire, en assumant que la position du CoM par rapport à la hanche reste constante. En plus, il faut calculer la trajectoire de balancement de la jambe de manière à ce que le pied se positionne pour le contact au sol, de la façon désirée et au moment désiré.

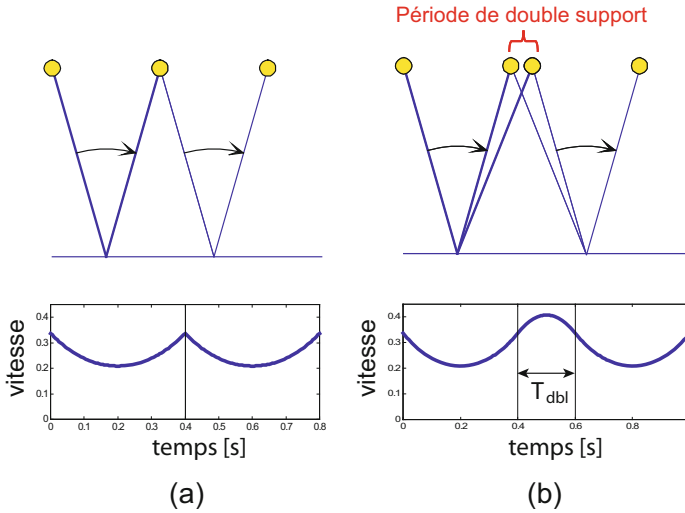


Figure 4.30 – Amélioration du modèle de changement de support : (a) le changement instantané créé des points de fléchissement sur la courbe de vitesse du CoM ; (b) une phase de double support permet d’obtenir des profils continus de vitesse et d’accélération du CoM.

Une fois que les trajectoires de la hanche et des deux pieds sont déterminées, les angles articulaires peuvent être obtenus par la cinématique inverse présentée au chapitre 2.

Cette méthode est basée sur l’hypothèse que la dynamique multicorps du robot peut être approximée par le modèle simple du pendule inversé, et que la validité de cette hypothèse est assurée par l’utilisation du ZMP, comme expliqué dans le chapitre précédent. Le calcul du ZMP sur la base d’un modèle multicorps permet d’évaluer les effets du balancement de la jambe et les erreurs sur la position du CoM, qui étaient négligés dans la modélisation par pendule inversé linéaire. La figure 4.32 montre deux trajectoires de ZMP. L’une utilise le pendule inversé linéaire, l’autre se base sur la dynamique multicorps puis sur la génération du modèle de marche. Les deux trajectoires sont suffisamment proches. Cet exemple permet d’affirmer que la dynamique multicorps peut être approchée dans ce cas par le modèle simple du pendule inversé.

4.3.6 Exemple d’implémentation

Étudions un exemple d’implémentation du générateur de modèle de marche proposé. La figure 4.33 montre le robot bipède HRP-2L développé au cours du *Humanoid Robotics Project* (HRP). Ce robot a été construit pour évaluer les capacités des jambes du HRP-2, robot humanoïde qui constituait le but final de ce projet de recherche. Chaque jambe possède six degrés de liberté et le robot est équipé d’un Pentium II 933MHz embarqué dans son corps. Le poids

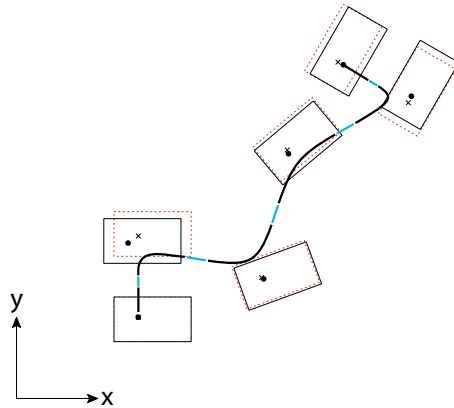


Figure 4.31 – Trajectoire de marche incluant une phase de double support. Les trajectoires du CoM pendant le double support sont signalées par des lignes gris clair. $z_c = 0.8$, $T_{sup} = 0.7$, $T_{dbl} = 0.1$, $a = 10$, $b = 1$.

total est 58.2 kg, comprenant 11.4 kg de batteries et un poids factice de 22.6 kg placé dans le corps.

L’algorithme présenté à la figure 4.25 peut générer un modèle de marche sur au moins deux pas successifs. Nous pouvons donc construire une boucle de contrôle sur la marche qui permette une modification en temps réel des caractéristiques des pas en fournissant les paramètres (s_x, s_y, s_θ) des deux pas suivants avec un joystick. La figure 4.34 montre des instantanés de l’expérimentation correspondante sur HRP-2L.

4.4 Génération d’une marche basée sur le ZMP

4.4.1 Modèle du chariot sur la table

Étudions le nouveau modèle de génération de la marche illustré par la figure 4.35. Nous observons un chariot de masse M qui se déplace sur une table de masse négligeable. Bien que le pied de la table soit trop petit pour conserver l’équilibre statique lorsque le chariot est sur le bord du plateau, nous pouvons maintenir l’équilibre dynamique en conférant une accélération suffisante au chariot. Cela représente le modèle du chariot sur la table.

Ce modèle considère une masse unique à hauteur constante. Nous retrouvons le cas présenté dans le chapitre 3.5.2. La position du ZMP est donc donnée par

$$p = x - \frac{z_c}{g} \ddot{x} \quad (4.64)$$

Cette équation est appelée l’équation du ZMP. Rappelons que l’équation du

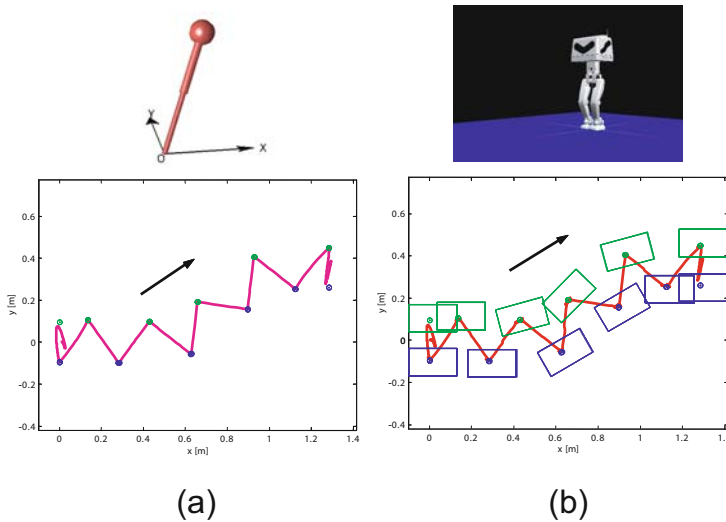


Figure 4.32 – Comparaison de trajectoires du ZMP : (a) le ZMP est calculé à partir du modèle du LIP-3D; (b) le ZMP est calculé par la dynamique multicorps dont la hanche se déplace comme un LIP-3D.

pendule inversé linéaire est donnée de la manière suivante (figure 4.24) :

$$\ddot{x} = \frac{g}{z_c}(x - p) \quad (4.65)$$

Nous pouvons immédiatement voir que les équations 4.64 et 4.65 sont les mêmes avec une sortie différente. De plus, en considérant p comme coordonnée du ZMP et non comme le placement du pied comme précédemment, nous pouvons traiter en même temps le couple à appliquer à la cheville du robot et la phase de double support [73].

Les modèles du LIP et du chariot sur la table sont comparés sur la figure 4.36. Pour le premier, la trajectoire du CoM est générée à partir de la position désirée du ZMP (figure 4.36(a)). Pour le second, la position du ZMP est générée par la trajectoire désirée du CoM (figure 4.36(b)). La relation entre ces deux modèles est donc l'inversion des entrées et sorties.

Comme nous l'avons décrit dans la partie précédente, la méthode basée sur le LIP s'appuie sur l'hypothèse de la relation entrée-sortie décrite par la figure 4.36(a). Le modèle de marche correspondant est calculé par le processus suivant :

(Spécifications trajectoire CoM) \Rightarrow (Position ZMP)

Dans ce cas, il est difficile de réaliser la trajectoire du ZMP prévue. En effet, il y a modification de la position du ZMP (placement du pied de support)

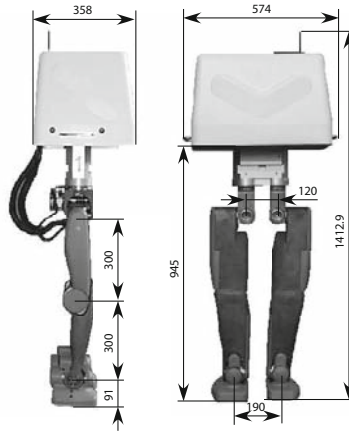


Figure 4.33 – Le robot bipède HRP-2L.

pendant la transition de marche par rapport à la méthode présentée au chapitre précédent.

Considérons à présent un modèle de marche généré à partir de la méthode du chariot sur la table. Dans ce cas, calculons le modèle de marche en tenant compte de la relation CoM-ZMP de la figure 4.36(b)¹² :

(Spécifications position du ZMP) \Rightarrow (Trajectoire du CoM)

En conséquence, nous obtenons un modèle de marche qui réalise la trajectoire du ZMP désirée. Cette méthode est appelée *génération du modèle de marche basé sur le ZMP*.

4.4.2 Génération d'un modèle de marche hors-ligne

La génération d'un modèle de marche basé sur le ZMP a été proposée initialement par Vukobratović et Stepanenko dans leur article de 1972 [83]. L'algorithme proposé dans cette publication était particulièrement gourmand en temps de calcul. Par la suite, Takanishi et al. ont proposé une méthode pratique qui transforme le modèle ZMP désiré, résout l'équation du ZMP 4.64 en fréquence et extrait une trajectoire du CoM en utilisant l'inverse FFT [76]. Le générateur de modèles, basé sur cette méthode, a joué un rôle très important dans la mise en place du Projet robotique humanoïde HRP.

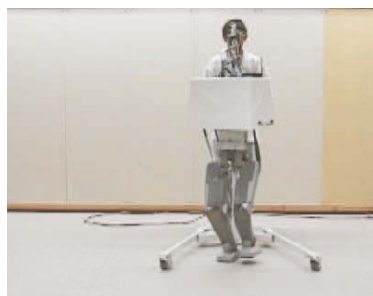
Dans cette partie du chapitre 4, nous allons présenter un algorithme de calcul rapide et efficace récemment proposé par Nishiwaki et al. [62]¹³. Nous

¹²Il existe une infinité de possibilités pour la trajectoire du CoM compatibles avec la trajectoire désirée du ZMP. Presque toutes présentent des divergences du modèle. La figure 4.36(b) peut être considérée comme une méthode assurant une solution réalisable.

¹³Une autre méthode performante et efficace a été introduite par Nagasaka [53].



Premier pas



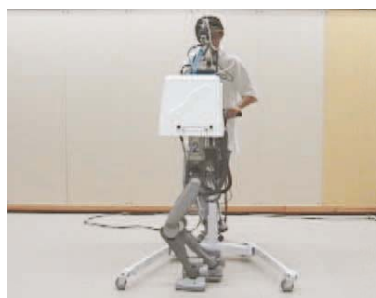
Quatrième pas



Huitième pas



Douzième pas



Quinzième pas



Dix-neuvième pas

Figure 4.34 – Instantanés de la boucle de contrôle sur la marche avec génération du modèle en temps réel.

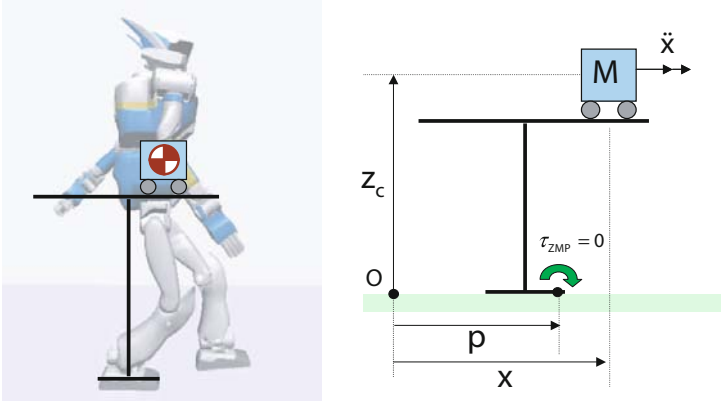


Figure 4.35 – Modèle du chariot sur une table : la dynamique du robot en train de marcher est modélisée par un chariot se déplaçant sur une table sans masse. L'état du chariot en mouvement détermine la position du centre de pression sur le sol, en d'autres termes le ZMP.

allons discrétiser l'équation du ZMP en considérant un pas de temps Δt . Dans ce but, la valeur de l'accélération \ddot{x} est approchée par

$$\ddot{x}_i = \frac{x_{i-1} - 2x_i + x_{i+1}}{\Delta t^2} \tag{4.66}$$

avec $x_i \equiv x(i\Delta t)$. En tenant compte de cette approximation, l'équation du ZMP discrétisée est

$$\begin{aligned} p_i &= ax_{i-1} + bx_i + cx_{i+1} \\ a_i &\equiv -z_c/(g\Delta t^2) \\ b_i &\equiv 2z_c/(g\Delta t^2) + 1 \\ c_i &\equiv -z_c/(g\Delta t^2) \end{aligned} \tag{4.67}$$

Reportons les équations 4.67 dans une colonne sur l'intervalle de temps spécifié ($1 \dots N$), nous obtenons l'équation matricielle suivante :

$$\begin{bmatrix} p'_1 \\ p_2 \\ \vdots \\ p_{N-1} \\ p'_N \end{bmatrix} = \begin{bmatrix} a_1 + b_1 & c_1 & 0 & & & & \\ & a_2 & b_2 & c_2 & \ddots & & \\ & & & & \ddots & & \\ & & & & & \ddots & \\ & & & & & & a_{N-1} & b_{N-1} & c_{N-1} \\ & & & & & & 0 & a_N & b_N + c_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} \tag{4.68}$$

où p'_1, p'_N sont fixés par les vitesses initiales et finales v_1, v_N :

$$p'_1 = p_1 + a_1 v_1 \Delta t, \quad p'_N = p_N - c_N v_N \Delta t$$

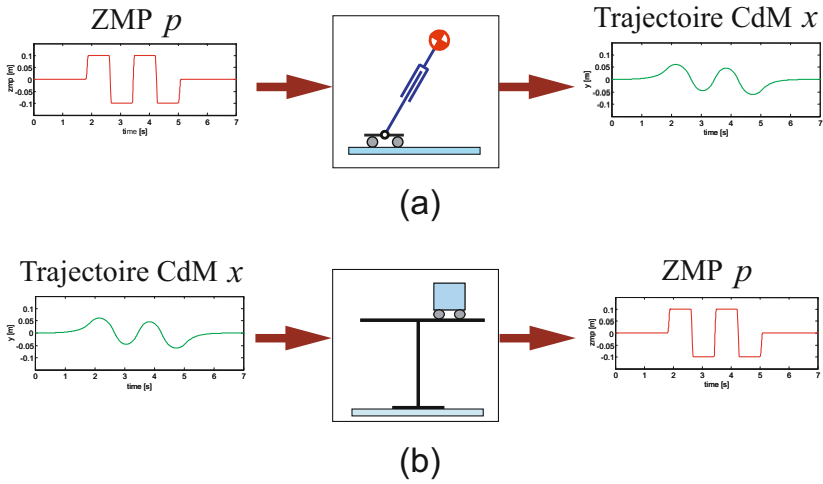


Figure 4.36 – Comparaison de deux modèles basés sur les relations entre la position du ZMP et la trajectoire du CoM : (a) LIP-3D. Entrée : position du ZMP ; Sortie : trajectoire du CoM ; (b) chariot sur la table. Entrée : trajectoire du CoM ; Sortie : position du ZMP.

L'équation matricielle 4.68 peut s'écrire sous la forme

$$\mathbf{p} = \mathbf{A}\mathbf{x}$$

et elle peut être résolue de la manière suivante :

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{p} \quad (4.69)$$

Bien que la matrice \mathbf{A} soit une matrice carrée de très grande dimension (plusieurs milliers de lignes et colonnes), il existe un algorithme efficace pour l'inverser [65] qui prend en compte le fait que cette matrice est tridiagonale (tous les éléments de la matrice sont nuls à l'exception de la diagonale principale et des deux diagonales immédiatement adjacentes).

À partir de cette trajectoire discrétisée, nous pouvons générer un modèle de marche basé sur le modèle multicorps présenté dans le chapitre 4.3.5. Enfin, nous pouvons calculer la trajectoire du ZMP à partir du modèle multicorps.

$$\mathbf{p}^* = \text{RealZMP}(\mathbf{x}) \quad (4.70)$$

La fonction $\text{RealZMP}()$ calcule le ZMP à partir du modèle multicorps. \mathbf{p}^* est la position résultante du ZMP. L'erreur sur la position du ZMP, $\mathbf{p}^* - \mathbf{p}^d$, présente des informations sur les différences entre l'approche par la méthode multicorps ou celle du chariot sur la table. Regardons à nouveau l'équation 4.69. Nous pouvons calculer les variations du CoM qui vont compenser l'erreur sur la position du ZMP :

$$\Delta\mathbf{x} = \mathbf{A}^{-1}(\mathbf{p}^* - \mathbf{p}^d)$$

La nouvelle trajectoire du CoM est

$$\mathbf{x} := \mathbf{x} + \Delta\mathbf{x}.$$

Nous initions une boucle avec l'équation 4.70 et répétons l'opération jusqu'à ce que l'erreur devienne suffisamment petite.

C'est un algorithme très efficace. D'après Nishiwaki et al. [60], il ne faut que 140 ms pour générer un modèle de marche de trois pas (3,2 s) sur le robot humanoïde H7 [61] qui possède 32 DDL et est équipé d'un Pentium II 750 MHz. Cette équipe a réalisé un contrôleur de marche en-ligne (par joystick), en générant le modèle de marche sur les trois pas suivants et en l'actualisant à chaque temps de cycle.

4.4.3 Génération de modèle de marche en-ligne

Nous allons présenter dans ce chapitre la méthode actuellement utilisée sur notre robot humanoïde HRP-2.

Commande en suivi du ZMP

En considérant un modèle chariot-table comme un système dynamique, on peut imaginer un servosystème qui réalise le suivi d'une cible ZMP par le biais d'une commande avec retour comme celle illustrée sur la figure 4.37.

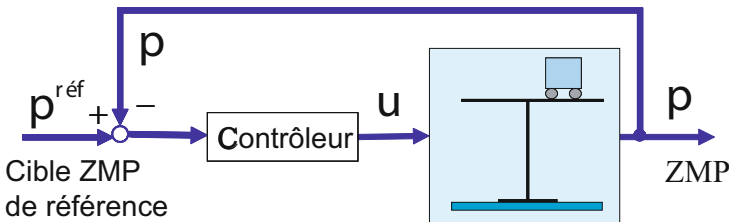


Figure 4.37 – Servocontrôleur pour suivre la trajectoire cible du ZMP.

Comme préparation, définissons une dérivée de l'accélération du chariot (jerk) comme une entrée du système. Cela permet de traiter le modèle chariot-table dans le cadre classique de la théorie actuelle du contrôle.

$$u = \ddot{x}$$

En utilisant cette entrée, nous pouvons récrire l'équation 4.64 selon la *repré-*

sentation suivante¹⁴ :

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \quad (4.71)$$

$$p = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}$$

Si nous utilisons un contrôleur standard, comme celui de la figure 4.37, nous n'obtiendrons pas un modèle de marche approprié. Prenons l'exemple de la figure 4.38(a) : le robot doit effectuer un pas de 30 cm vers l'avant. Le ZMP cible varie selon un échelon à 1,5 s, mais reste constant avant et après ce changement. Nous pouvons remarquer que le mouvement du centre de masse commence avant la variation du ZMP. Cela signifie que le chariot doit se déplacer avant le changement d'entrée dans le système représenté sur la figure 4.37. D'un autre côté, dans un servosystème ordinal, nous obtenons naturellement en sortie le mouvement désiré avec un délai par rapport au changement de la référence, comme montré sur la figure 4.37(b). Dans le cadre de la génération de modèle de marche pour des robots bipèdes, une information future doit revenir modifier le passé !

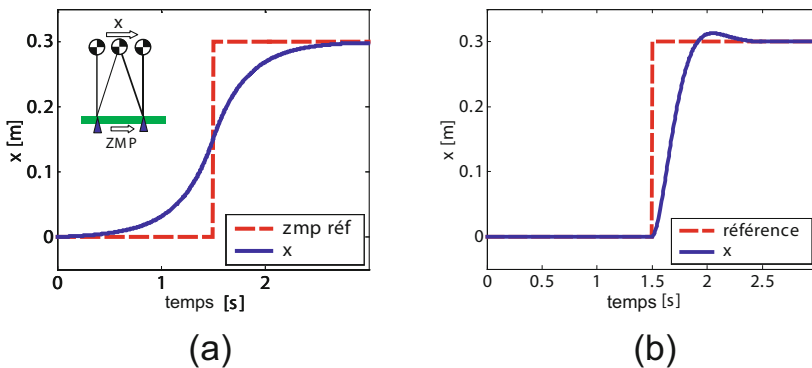


Figure 4.38 – Effets entrées-sorties : (a) avancée de un pas : ZMP (ligne pointillée) et CoM (ligne épaisse). Le centre de masse commence son mouvement avant la variation du ZMP ; (b) dans un servosystème ordinal, la sortie (ligne épaisse) varie après le changement de l'entrée de référence (ligne pointillée).

Prévisualisation de la commande

Nous avons donc besoin d'informations du futur pour générer le comportement présent du robot. Il n'est cependant pas nécessaire de construire une

¹⁴Dans la théorie du contrôle moderne, tout type de système dynamique est décrit selon ce format. Il est appelé *représentation du système*.

machine à remonter le temps¹⁵. Par exemple, lorsque nous conduisons, nous tenons compte de l'état de la route vers laquelle nous nous dirigeons, et non pas de celle qui est sous nos pneus. Connaître à l'avance l'état de la route nous permet d'adopter une conduite souple. Nous pouvons facilement imaginer les conséquences sur notre conduite d'une visibilité limitée à quelques centimètres de l'avant du véhicule.

Une méthode de contrôle basée sur des informations futures est appelée **précontrôle** [22, 69, 46]. Nous allons détailler la conception d'un contrôleur basé sur cette théorie.

Dans un premier temps, nous discrétisons le système d'équations 4.71 continu dans le temps. Pour cela, nous utilisons un échantillon de temps Δt pour la conception d'un contrôleur digital.

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{b}u_k \\ p_k = \mathbf{c}\mathbf{x}_k \end{cases} \quad (4.72)$$

où

$$\begin{aligned} \mathbf{x}_k &\equiv [x(k\Delta t) \quad \dot{x}(k\Delta t) \quad \ddot{x}(k\Delta t)]^T \\ u_k &\equiv u(k\Delta t) \\ p_k &\equiv p(k\Delta t) \end{aligned}$$

et

$$\mathbf{A} \equiv \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{b} \equiv \begin{bmatrix} \Delta t^3/6 \\ \Delta t^2/2 \\ \Delta t \end{bmatrix} \quad \mathbf{c} \equiv [1 \quad 0 \quad -z_c/g]$$

Afin que la sortie p_k du système suive d'aussi près que possible le ZMP cible p_k^{ref} , considérons le problème de minimisation de l'indice de performance suivant :

$$J = \sum_{j=1}^{\infty} \{Q(p_j^{ref} - p_j)^2 + Ru_j^2\} \quad (4.73)$$

où Q et R sont des poids positifs. Nous venons de définir le **problème de commande en suivi**. D'après la théorie du précontrôle, l'indice de performance J peut être minimisé par l'entrée suivante, qui utilise les références de la cible future jusqu'à N pas de temps [39] :

$$u_k = -\mathbf{K}\mathbf{x}_k + [f_1, f_2, \dots, f_N] \begin{bmatrix} p_{k+1}^{ref} \\ \vdots \\ p_{k+N}^{ref} \end{bmatrix} \quad (4.74)$$

¹⁵Une technologie réaliste pour construire une machine temporelle pouvant envoyer des informations dans le passé est décrite dans le roman de James P. Hogan, *Thrice upon a Time*. Cette technologie utilise les principes de la physique des particules : c'est un chef-d'œuvre de la littérature de science fiction.

où

$$\begin{aligned} K &\equiv (R + \mathbf{b}^T \mathbf{P} \mathbf{b})^{-1} \mathbf{b}^T \mathbf{P} \mathbf{A} \\ f_i &\equiv (R + \mathbf{b}^T \mathbf{P} \mathbf{b})^{-1} \mathbf{b}^T (\mathbf{A} - \mathbf{b} \mathbf{K})^{T*(i-1)} \mathbf{c}^T \mathbf{Q} \end{aligned} \quad (4.75)$$

La matrice \mathbf{P} est une solution de l'équation de Riccati suivante¹⁶ :

$$\mathbf{P} = \mathbf{A}^T \mathbf{P} \mathbf{A} + \mathbf{c}^T \mathbf{Q} \mathbf{c} - \mathbf{A}^T \mathbf{P} \mathbf{b} (\mathbf{R} + \mathbf{b}^T \mathbf{P} \mathbf{b})^{-1} \mathbf{b}^T \mathbf{P} \mathbf{A} \quad (4.76)$$

En observant l'équation 4.74, nous pouvons constater qu'un précontrôle consiste en un retour d'état (le premier terme du membre de droite) et en une commande prédictive d'un produit interne entre la cible de référence future à N pas et les poids $[f_1, \dots, f_N]$ (le second terme).

Amélioration du précontrôleur

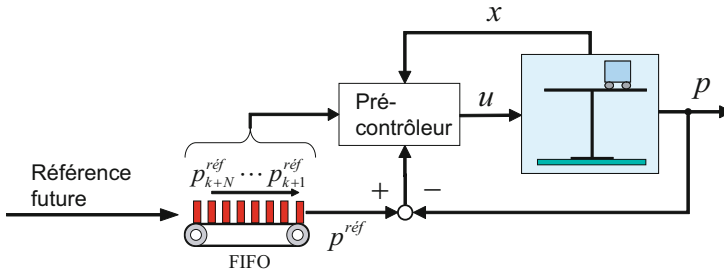


Figure 4.39 – Génération du modèle de marche par précontrôle.

Nous avons constaté un *offset* dans l'erreur de poursuite du ZMP lors de génération de modèles de marche sur des longues distances en utilisant l'équation 4.74. Pour résoudre ce problème, nous avons récrit l'équation 4.72 sous la forme développée suivante :

$$\begin{cases} \mathbf{x}_{k+1}^* = \tilde{\mathbf{A}} \mathbf{x}_k^* + \tilde{\mathbf{b}} \Delta u_k \\ p_k = \tilde{\mathbf{c}} \mathbf{x}_k^* \end{cases} \quad (4.77)$$

où la nouvelle entrée et le vecteur d'état sont définis par

$$\Delta u_k \equiv u_k - u_{k-1} \quad \Delta \mathbf{x}_k \equiv \mathbf{x}_k - \mathbf{x}_{k-1} \quad \mathbf{x}_k^* \equiv \begin{bmatrix} p_k \\ \Delta \mathbf{x}_k \end{bmatrix}$$

Les matrices s'écrivent

$$\tilde{\mathbf{A}} \equiv \begin{bmatrix} 1 & \mathbf{c} \mathbf{A} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \quad \tilde{\mathbf{b}} \equiv \begin{bmatrix} \mathbf{c} \mathbf{b} \\ \mathbf{b} \end{bmatrix} \quad \tilde{\mathbf{c}} \equiv [1 \ 0 \ 0 \ 0]$$

¹⁶Il n'est pas nécessaire de comprendre la résolution de cette équation matricielle compliquée. En utilisant la commande `dlqr` de la toolbox « Control System » de Matlab ou GNU Octave, la solution numérique est obtenue immédiatement pour \mathbf{P} et \mathbf{K} [26].

Concevons un contrôleur pour le système d'équations 4.77, qui minimisera l'indice de performance suivant :

$$J = \sum_{i=k}^{\infty} \{Q(p_j^{ref} - p_j)^2 + R\Delta u_k^2\} \tag{4.78}$$

Le précontrôleur est

$$\Delta u_k = -\tilde{\mathbf{K}}\mathbf{x}_k^* + \sum_{j=1}^N \tilde{f}_j \Delta p_k^{ref} \tag{4.79}$$

où $\tilde{\mathbf{K}}, \tilde{f}_j$ sont des gains, obtenus en substituant $\tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}, Q$ et R dans les équations 4.75 et 4.76. En effectuant la somme des équations 4.79 pour $k = 1, \dots, N$, nous obtenons le précontrôleur pour le système initial de l'équation 4.72.

$$u_k = -K_s \sum_{i=0}^k (p_j^{ref} - p_j) - K_x \mathbf{x}_k + \sum_{j=1}^N \tilde{f}_j p_k^{ref}$$

$$\begin{bmatrix} K_s \\ K_x \end{bmatrix} \equiv \tilde{\mathbf{K}}$$

Un diagramme illustrant la génération de modèle basée sur le précontrôle est donné à la figure 4.39. La cible future du ZMP de référence est enregistrée dans un tampon FIFO (*First-In-First-Out*). Sa valeur de sortie est vue comme la référence *actuelle*. Le précontrôleur calcule la valeur d'entrée de la commande en utilisant la référence du ZMP contenue dans le tampon et l'état du chariot. L'état du chariot x, \dot{x} est le résultat de la génération de modèle, c'est-à-dire le mouvement du CoM qui satisfait le ZMP cible.

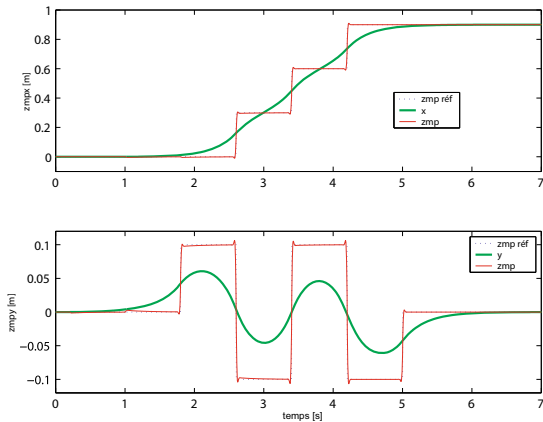


Figure 4.40 – Trajectoire du CoM obtenue par précontrôle.

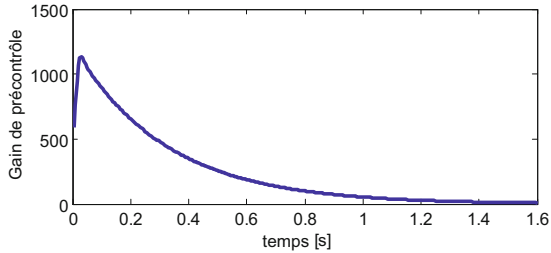
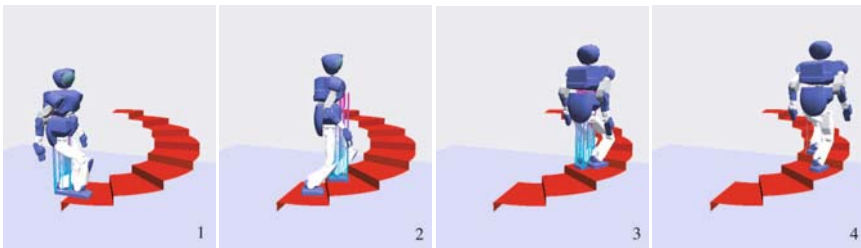
Figure 4.41 – Précontrôle de gain f .

Figure 4.42 – Simulation de marche sur des escaliers en spirale.

La trajectoire du CoM calculée par le biais de cette méthode et le ZMP résultant sont montrés sur la figure 4.40. Le graphe du haut représente les mouvements selon la direction de marche. Celui du bas représente la direction latérale¹⁷. Nous pouvons voir que les mouvements désirés du ZMP sont générés pour un profil en escalier du ZMP cible ou un profil de marche. Le gain utilisé pour ces calculs est montré sur la figure 4.41. En remarquant sur ce graphe que ce gain devient négligeable à 1,6 s, nous pouvons déduire que l'information qui provient d'au moins 1,6 s dans le futur n'affectera pas les performances de la commande.

La figure 4.42 illustre un modèle de marche sur des escaliers en spirale par la méthode expliquée précédemment.

4.4.4 Filtres dynamiques utilisant le précontrôle

Structure d'un filtre dynamique

La génération du modèle de comportement basée sur le précontrôle dépend du modèle chariot-table. De ce fait, elle ne garantit pas la stabilité d'un mouvement qui ne peut pas être représenté par le modèle simplifié. Par exemple, un changement important dans les configurations de la partie supérieure du

¹⁷Le plan contenant le vecteur de direction et l'axe vertical est appelé plan sagittal. Le plan contenant le vecteur de direction et le vecteur latéral est appelé plan latéral.

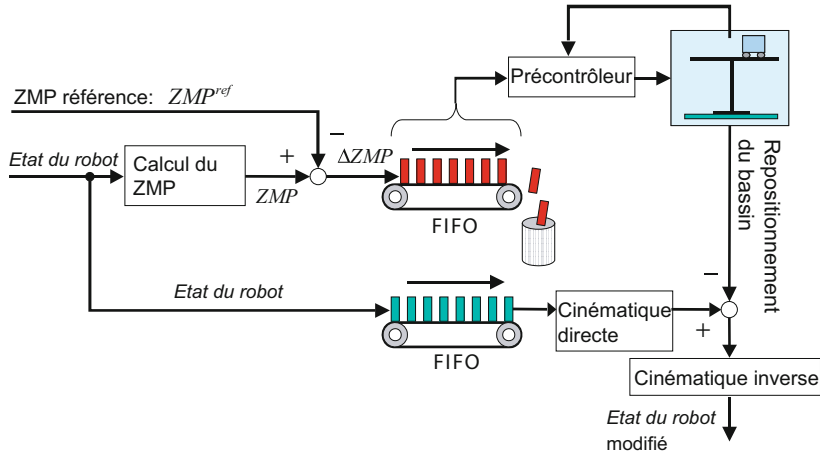


Figure 4.43 – Structure d'un filtre dynamique basé sur le précontrôle.

robot (tronc et bras) pendant la marche. Dans une telle situation, nous pouvons toujours utiliser le modèle chariot-table mais comme un système erroné autour du mouvement désiré. Par la suite, les modifications à apporter à la trajectoire du CoM pour compenser l'erreur sur le ZMP peuvent être calculées *via* le précontrôle.

La figure 4.43 montre la structure entière du système proposé. Les entrées sont le ZMP cible (ZMP^{ref}) et l'état complet du robot (*Robot state*), qui contient les positions et vitesses articulaires de l'ensemble du robot, la configuration de la hanche, ses vitesses linéaire et angulaire. Nous pouvons calculer l'erreur sur le ZMP (ΔZMP) à partir de ces entrées, et les entrer dans le tampon FIFO. Il faut aussi conserver une copie de l'état du système dans un autre tampon. Lorsque l'on récupère ΔZMP en sortie du FIFO après un délai, nous lisons l'erreur future sur le ZMP en fonction du délai. À partir de ces données, nous pouvons calculer la compensation adaptée à appliquer au CoM par le précontrôle. En appliquant cette compensation à l'état du robot retardé, nous obtenons les trajectoires angulaires qui permettent de mieux réaliser le ZMP cible de référence. D'une manière générale, le système représenté sur la figure 4.43 est appelé filtre dynamique. Un filtre dynamique convertit un modèle de mouvement donné en un modèle amélioré qui respecte un ensemble de propriétés [53].

Évaluation du filtre dynamique

Comme exemple, réalisons un modèle qui fait s'arrêter le robot humanoïde HRP-2 pendant la marche, puis reprend la marche normalement, comme montré sur la figure 4.44. Dans un premier temps, nous ajoutons simplement le modèle de squat au modèle de marche en considérant une altitude constante

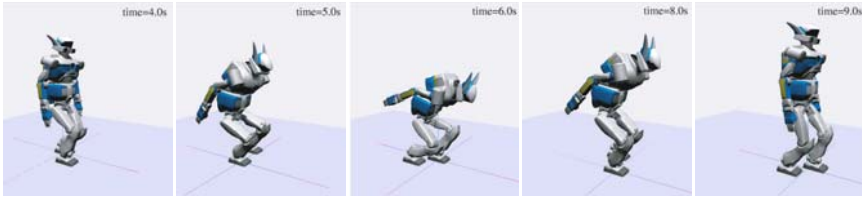


Figure 4.44 – Exemple de modèle de marche obtenu grâce à un filtre dynamique basé sur le précontrôle.

du CoM. La trajectoire résultante du ZMP est montrée sur la figure 4.45(a). Nous pouvons voir que le ZMP (ligne épaisse) approche de la limite du polygone de sustentation (lignes en pointillés) et le modèle de marche a une faible marge de stabilité. À cause de cela, le robot tombe immédiatement après le mouvement de squat au cours de la simulation dynamique.

Ce modèle de marche a été modifié par le filtre dynamique de la figure 4.43. Nous obtenons alors le modèle de la figure 4.45(b). Nous avons utilisé pour cela un délai de 0,8 s pour le tampon FIFO et un précontrôle sans l'intégrateur (équation 4.74). L'erreur absolue maximale, qui était à l'origine de 0,11 m, décroît à 0,05 m par l'utilisation du filtre dynamique. Puisque le nouveau modèle de marche a une marge de stabilité suffisante, le robot parvient à réaliser le mouvement de squat puis reprendre la marche lors de la simulation dynamique.

Le modèle de marche modifié est illustré sur la figure 4.44. En appliquant le filtre dynamique proposé, nous obtenons une marche réalisable malgré les modifications non négligeables éloignant le comportement de la dynamique du modèle chariot-table.

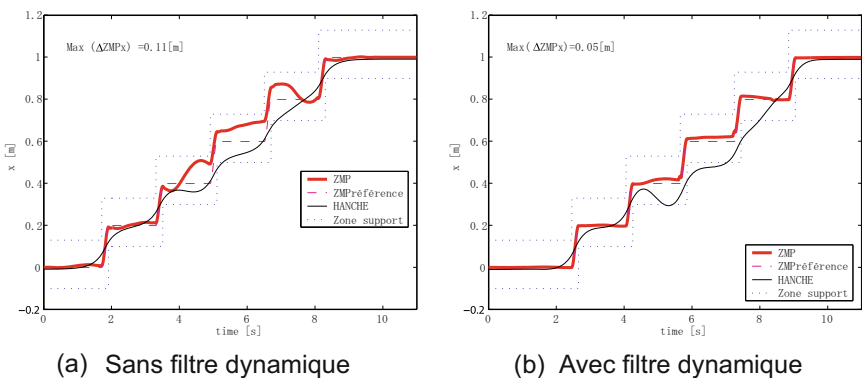


Figure 4.45 – Effets du filtre dynamique : le ZMP (ligne grasse) est modifié pour être forcé à l'intérieur du polygone de sustentation (lignes pointillées), afin d'augmenter la marge de stabilité du système.

4.5 Stabilisateur

La figure 4.46 montre une séquence filmée du HRP-2 marchant sur un sol inégal. Nous avons expliqué, au début de ce chapitre, que l'environnement réel contient des incertitudes non intégrables dans un modèle, telles les irrégularités du sol. Un robot bipède, suivant un modèle de marche réalisé hors-ligne, souffre rapidement de l'accumulation des erreurs entre l'état de référence et l'état actuel, et tombe au bout de quelques pas. Afin de supprimer l'accumulation des erreurs et garantir la marche tout au long de l'itinéraire demandé, nous devons introduire un stabilisateur.

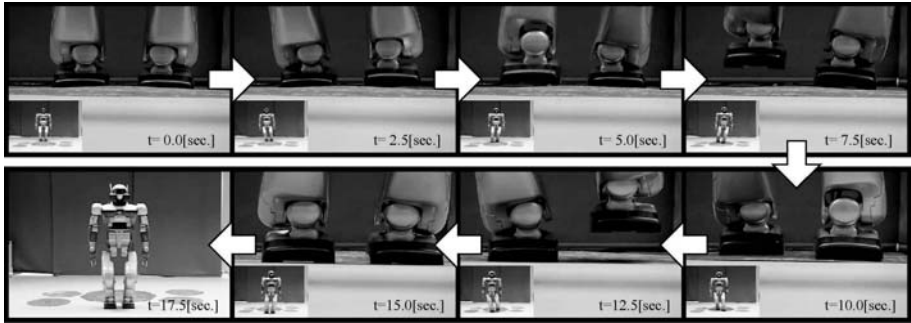


Figure 4.46 – Les pieds du robot HRP-2 marchant sur un sol irrégulier.

4.5.1 Principes de stabilisation de la commande

Nous allons présenter les principes de base utilisés pour mettre en place un stabilisateur. Afin de simplifier la description, nous expliquons la méthode pour stabiliser un état immobile. Comme le stabilisateur a pour objectif l'absorption de petites erreurs autour d'un modèle de référence, nous ne perdons pas, dans cas particulier, la généralité de l'approche.

1 Commande basée sur le couple de la cheville

Nous pouvons stabiliser l'équilibre de l'ensemble du corps ou de la posture par l'utilisation du couple de la cheville de la jambe de support en modélisant le robot complet comme un simple pendule inversé. Le couple de la cheville et le pendule inversé linéaire sont liés par la relation

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{M z_c} \tau. \quad (4.80)$$

La loi de commande la plus simple pour stabiliser ce pendule est la suivante¹⁸ :

$$\tau = -k_p x - k_d \dot{x}, \quad (4.81)$$

¹⁸Pour l'utilisation d'une loi de commande plus élaborée, il faut considérer le problème

où k_p, k_d sont les gains de la boucle de retour déterminés pour une fréquence de réponse ω donnée et un coefficient d'amortissement ζ ,

$$k_p \equiv M(z_c \omega_n^2 + g) \quad k_d \equiv 2M z_c \zeta \omega_n$$

Bien que cette loi de commande ait l'air simple, sa réalisation est difficile. Premièrement, une commande précise du couple de la cheville est un problème très difficile pour la plupart des robots marcheurs équipés de réducteurs à fort coefficients. De plus, pour réaliser une marche stable, nous avons besoin d'un réglage précis des gains de retour et de bornes de couples pour supprimer les effets de décollement talons-pointes causés par un couple trop grand au niveau de la cheville.

Cette méthode a été utilisée par de nombreux robots développés dans les années 1980 et 1990. Par exemple, WL-10RD réalisé par Takamishi et al. [76], Idaten II par Miyazaki and Arimoto [2], Kenkyaku 2 par Sano et Furusho [68], Meltran II par Kajita and Tani [34].

2 Commande modifiant le placement des pieds

La seconde approche de stabilisation basée sur le pendule inverse consiste à modifier le placement des pieds. Pour cette commande, nous pouvons appliquer le même principe de contrôle de la vitesse de marche que celui décrit dans le chapitre 4.3.3 (voir la figure 4.23 page 120). Il y a très peu de robots qui ont utilisé une modification du placement des pieds pour la stabilisation : nous pouvons citer BIPER-3, un échassier marcheur développé par Shimoyama et Miura [51, 71], et la série des robots sauteurs de Raibert et ses collègues [66].

3 Contrôle du ZMP par l'accélération du CoM

Considérons un stabilisateur basé sur le modèle chariot-table. Dans ce cas, nous devons mesurer le ZMP pour concevoir une commande à retour. En notant T la constante de temps du capteur ZMP, l'équation du ZMP est

$$p = \frac{1}{1 + sT} \left(x - \frac{z_c}{g} \ddot{x} \right) \quad (4.82)$$

La représentation du système, avec l'accélération du chariot \ddot{x} comme entrée, est

$$\frac{d}{dt} \begin{bmatrix} p \\ x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -1/T & 1/T & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} -z_c/(gT) \\ 0 \\ 1 \end{bmatrix} \ddot{x} \quad (4.83)$$

Nous pouvons établir une loi de commande à retour d'état pour stabiliser ce système :

$$\ddot{x} = -k_1 p - k_2 x - k_3 \dot{x} \quad (4.84)$$

de l'intégration mécanique. Il apparaît une accumulation d'erreurs d'intégration menant à la saturation du couple de la cheville.

Les gains k_1 , k_2 , et k_3 de la boucle de retour peuvent être déterminés par la théorie du contrôle classique, par exemple le placement de pôle ou la commande optimale. Cette loi de commande a été introduite en tant que *trunk compliance control* par Nagasaka, Inaba and Inoue [54]¹⁹. Cette loi de commande est efficace pour des robots marcheurs équipés de pieds ou possédant une grande rigidité. Elle a été utilisée pour les robots humanoïdes H5 et H7 [30].

Comme méthode alternative, Okada, Furuta et Tomiyama ont proposé une stabilisation basée sur le contrôle de l'accélération du ZMP *via* un changement d'échantillonnage du temps. Cette approche a été mise en place sur le robot bipède MK.3 et le robot humanoïde morph3 [31, 63].

4 Contrôle de la posture du corps par les angles de la hanche

La plupart des robots marcheurs ont pour contrainte de conserver le corps vertical pendant la marche. Le moyen le plus simple est de faire effectuer une rotation aux angles de la hanche de telle sorte que le corps conserve sa verticalité par rapport aux sorties fournies par les capteurs posturaux. Cela est possible même à partir des données récoltées au niveau du pied car le couple généré à la hanche découle directement des forces de frottement au niveau du sol. C'est ce type de commande que Raibert a implémenté sur ses robots sauteurs [66], ainsi que Kumagai et al. [45] sur leur robot humanoïde.

5 Commande par modèle de ZMP

Une nouvelle méthode de commande posturale a été proposée par Hirose, Takenaka et al. : c'est la *commande par modèle de ZMP*. D'après leurs explications, le principe de fonctionnement de cette approche est le suivant :

Lorsque le corps du robot réel est d'avantage incliné que le modèle, le modèle est fortement accéléré par rapport à la trajectoire planifiée. Cela modifie les forces inertielles désirées et le ZMP cible recule par rapport au ZMP original. De ce fait, le robot réel retrouve la posture convenable [25].

Considérons la signification physique de cette commande par modèle de ZMP, en utilisant la version du modèle chariot-table de la figure 4.47. Nous assumons que le pied de la table est doté d'une articulation pivot libre qui incline la table d'un angle θ . L'accélération du chariot \ddot{x} correspond à l'accélération du corps du robot.

Les équations du mouvement obtenues par la méthode de Lagrange s'écrivent

$$\begin{cases} (x^2 + z_c^2)\ddot{\theta} + \ddot{x}z_c - g(z_c \sin \theta + x \cos \theta) + 2x\dot{x}\dot{\theta} = \tau/M \\ \ddot{x} + \ddot{z}_c - \dot{\theta}^2 x + g \sin \theta = f/M \end{cases} \quad (4.85)$$

¹⁹Dans leur travail original, ils ont utilisé différentes procédures pour dériver la loi de commande équivalente. Napoleon et al. ont montré une autre interprétation de leur loi de commande en terme de théorie de la commande à zéro dynamique [57].

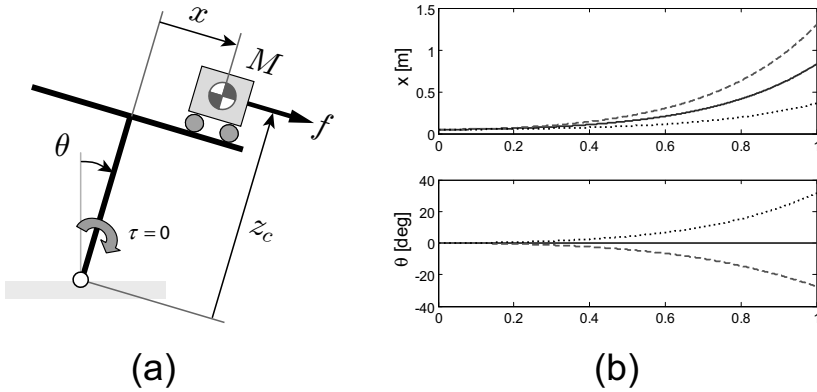


Figure 4.47 – (a) un modèle chariot-table doté d’une rotation libre. θ prend des valeurs positives dans le sens trigonométrique indirect ; (b) accélération du chariot et inclinaison de la table. Lignes continues : la table reste verticale si l’accélération est régulière ($\ddot{\theta} = 0$) ; lignes pointillées : avec une accélération trop grande, la table se redresse ($\ddot{\theta} < 0$) ; lignes en tirets : avec une accélération trop petite, la table tombe ($\ddot{\theta} > 0$). La commande par modèle de ZMP utilise ce phénomène.

où τ est le couple agissant au pied de la table, f est la force permettant d’accélérer le chariot sur la table. En linéarisant la première équation autour de $\theta, \dot{\theta} \approx 0$ et en substituant $\tau = 0$, nous obtenons l’équation suivante :

$$(x^2 + z_c^2)\ddot{\theta} = gx + gz_c\theta - z_c\ddot{x} \quad (4.86)$$

Ensuite, nous considérons que la position désirée du chariot x_d (la cible) est générée par la dynamique suivante :

$$\ddot{x}_d = \frac{g}{z_c}(x_d - p_d) \quad (4.87)$$

où p_d est le ZMP cible. Intuitivement, lorsque le ZMP cible est placé en amont ($p_d < 0$), le chariot accélère fortement. Lorsque le ZMP cible est placé en aval ($p_d > 0$), l’accélération devient faible. L’équation 4.87 représente donc un modèle pour générer l’accélération dans lequel le ZMP cible *peut être défini en dehors du polygone de sustentation*.

Supposons maintenant que nous puissions contrôler l’accélération comme nous le désirons. En substituant $x = x_d$ et l’équation 4.87 dans l’équation 4.86, nous obtenons :

$$\ddot{\theta} = \frac{gz_c}{x_d^2 + z_c^2}\theta + \frac{g}{x_d^2 + z_c^2}p_d \quad (4.88)$$

À partir de ce résultat, nous pouvons voir que l’inclinaison de la table θ peut être contrôlée par le ZMP cible p_d , et sa dynamique est déterminée par la position du chariot et l’accélération de la gravité.

6 Absorption des impacts par recul angulaire

On peut réaliser un système virtuel ressort-amortisseur en instrumentant la plate-forme d'un capteur de couple articulaire et en décrivant une position de référence comme une fonction de la mesure de couple. Takamishi et al. ont utilisé un tel système pour absorber les vibrations liées à l'impact de reprise de contact avec le sol [76]. Lorsque l'articulation possède un faible rapport de réduction, approximativement de 1/1 à 1/50, elle subit un recul dû aux forces extérieures du contact avec l'environnement. De ce fait, il est nécessaire que la commande en position elle-même se comporte comme un ressort-amortisseur. Kenkyaku 1, robot bipède développé par Furusho et al., a utilisé cette approche pour la commande de la marche [15]. Sorao, Murakami et Ohnishi ont mentionné une absorption d'impacts au moment du changement de support par l'utilisation d'un contrôleur d'impédance basé sur la moindre mesure du capteur de couple à partir d'un *observateur de perturbation* [72].

7 Stabilisation par commande LQ

Un robot marcheur peut être modélisé comme un système multi-entrées-multi-sorties (multi-input-multi-output : MIMO) utilisant en entrée le vecteur des couples articulaires \mathbf{u} et en sortie l'état de toutes les articulations. Définissons un vecteur d'état \mathbf{x} contenant la configuration et les vitesses de toutes les articulations, et linéarisons l'équation du mouvement de \mathbf{u} à \mathbf{x} . Par une application directe de la théorie de la commande LQ, nous obtenons une loi de commande par retour d'effort :

$$\mathbf{u} = -\mathbf{K}\mathbf{x}$$

où \mathbf{K} est un gain de retour consistant en une matrice de très grandes dimensions $N \times 2N$ où N est le nombre d'articulations. Cette approche a été appliquée à un bipède de 6 DDL, CW-2 développé par Mita et al. [49], ainsi qu'à un bipède de 12 DDL développé par Yoshino [91]. Les travaux effectués rapportent que le robot a pu marcher convenablement à une vitesse de 3 km/h sur un sol présentant des inégalités de 6 mm.

4.5.2 Stabilisation du robot humanoïde Honda

Un système efficace de stabilisation a été construit en combinant plusieurs des principes de commande mentionnés précédemment. Nous allons décrire une des implémentations réussies.

La figure 4.48 montre le robot humanoïde P2 développé par Honda Motor Co., Ltd et présenté officiellement en 1996. La qualité de son système de commande est encore compétitive de nos jours²⁰ et sa technologie est bien divulguée [77, 25, 24].

²⁰Nous supposons que le même type de commande est utilisé pour d'autres robots Honda, comme ses deux successeurs les humanoïdes P3 et ASIMO.



Figure 4.48 – Le robot humanoïde P2 (1996) (autorisation de Honda Motor Co., Ltd).

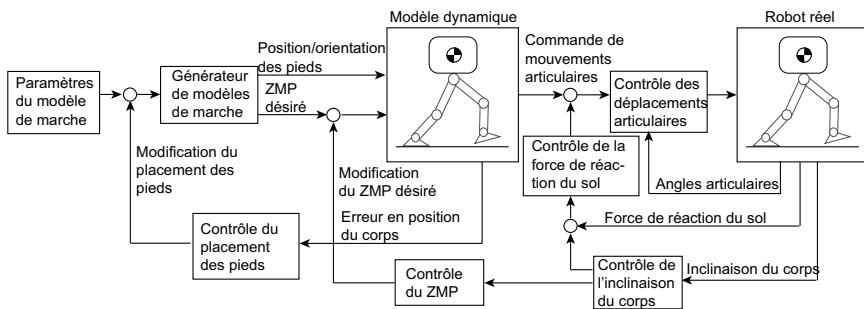


Figure 4.49 – Le système de commande de la marche du Honda P2.

Le système de commande de la marche du P2 est illustrée par la figure 4.49. La boucle de retour qui passe par l'inclinaison du corps et les forces de réaction du sol correspond à la commande en couple de la cheville décrite dans **1**. Remarquons que les mécanismes passifs insérés dans les pieds du P2 facilitent l'implémentation de sa commande. Lorsque le corps du robot subit une inclinaison excessive, la commande à modèle ZMP **5** prend le relais pour rétablir la posture. L'erreur de déplacement horizontal qui en découle est corrigée par la modification du placement du pied **2** qui se prépare à atterrir. Comme on peut le voir, une commande robuste de la marche du P2 est assurée par la combinaison ingénieuse de plusieurs schémas de commande.

4.6 Les pionniers de la technologie en marche dynamique

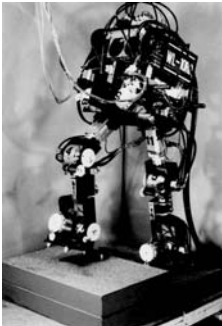
Nous allons effectuer un retour dans le temps pour exposer l'histoire de la commande dynamique de la marche bipède. Depuis les débuts de la recherche en robotique, la marche bipède a été reconnue comme un challenge de grande difficulté. Les travaux de recherche les plus anciens pour développer le hardware des plate-formes bipèdes ont commencé avec Ichiro Kato de l'université de Waseda en 1966, et avec D.C. Witt de l'université d'Oxford en 1968. Comme expliqué dans le chapitre 1, en 1973, Kato et ses collègues ont construit le premier robot humanoïde WABOT-1, avec deux bras et deux jambes, et commandé par un ordinateur. Ce travail était un exploit, bien que WABOT-1 ne puisse effectuer qu'une marche statique. Vers 1980, il y eut un effort énorme dans la recherche pour réaliser la marche bipède en dynamique, et beaucoup de chercheurs japonais ont développé des théories et des plate-formes humanoïdes. En 1986, il existait déjà de nombreux robots bipèdes pouvant réaliser la marche dynamique comme le montre la figure 4.50 [1, 14, 42, 50].

Pendant la même année, un groupe de chercheurs de Honda Motor Co., Ltd. a lancé un projet secret de robot marcheur. Après un silence absolu de 10 années²¹, leurs efforts ont abouti au robot humanoïde P2. Ce robot est soudainement apparu avec une conception mécanique d'excellente qualité, ainsi qu'une technologie de marche particulièrement aboutie (et toujours d'actualité!). Cela a découragé de nombreux chercheurs actifs du domaine, et en même temps inspiré de nombreux autres chercheurs en leur montrant les incroyables possibilités qu'offraient les robots humanoïdes. En l'occurrence, un projet national japonais de recherche et développement a été inauguré à la suite de cette révélation : le *Humanoid Robotics Project*²² (1998-2000).

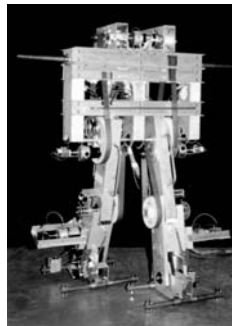
On peut penser, en s'habituant à des robots comme ASIMO ou QRIO, que les robots de la figure 4.50 sont d'aspect primitif. Il faut cependant garder en mémoire que les chercheurs qui les ont développés avec si peu de moyens, des ordinateurs et des moteurs non performants, sont réellement les pionniers de la technologie de la marche bipède actuelle. En effet, la plupart des techniques de commande de la marche bipède proviennent de cette époque. Il est de plus difficile de croire que le projet de recherche de Honda a été mené indépendamment des publications déjà parues en 1986. Les pas de géants produits pour atteindre ce niveau de qualité dans le progrès ne peuvent être effectués par un seul génie en une nuit, et il n'y a pas de percée spectaculaire indépendamment des recherches qui l'ont précédée. De ce fait, le vrai progrès ne peut se faire que par des échanges d'idées et des collaborations de beaucoup de chercheurs, quels que soient leurs entreprises d'appartenance respectives ou leurs pays d'accueil.

²¹Le budget investi, ainsi que l'investissement humain pendant cette période, est estimé à plus de cent fois supérieur à celui des universités. Comme, à cette époque, la communauté robotique avait une approche plutôt négative à l'encontre des robots bipèdes, la vision à long terme des membres de Honda doit être grandement respectée!

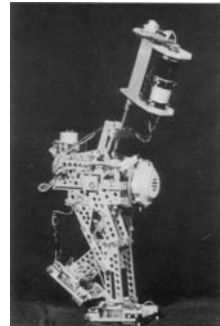
²²Projet robotique humanoïde



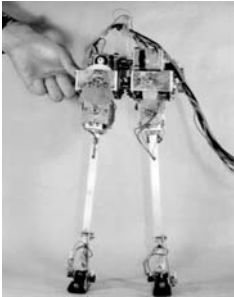
Univ. Waseda WL-10RD (1984)



Univ. Gifu Kenkyaku 2 (1983)



Inst Tech. Tokyo MEG-2 (1982)



Univ. Tokyo BIPER-3 (1982)



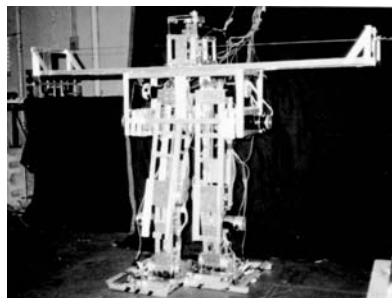
Univ. Tokyo BIPER-4 (1983)



Inst Tech Tokyo BIPMAN2 (1982)



Univ. Chiba CW-2 (1983)



Univ. Osaka Idaten II (1982)

Figure 4.50 – Les robots bipèdes qui pouvaient marcher dynamiquement avant 1986 [1, 14, 42, 50].

4.7 Autres méthodes pour la commande bipède

Nous avons présenté précédemment les concepts les plus utilisés pour la commande dynamique bipède. Nous allons nous attarder dans cette partie sur d'autres commandes existantes et qui diffèrent totalement des précédentes.

4.7.1 Marche dynamique passive

Un marcheur dynamique passif est un robot qui descend une pente douce en n'utilisant que son énergie potentielle. Des jouets utilisant cette technique sont connus de longue date. Cependant, Tad McGeer a intensément analysé la dynamique de ce mouvement et a montré qu'un marcheur passif équipé d'articulations libres aux genoux peut marcher sans moteurs ni commande... et présenter un mouvement extraordinairement ressemblant au mouvement de marche humaine. Il a montré les résultats de ses analyses aussi bien en simulation qu'en expérimentation [47, 48]. Il a suggéré que son marcheur passif deviendrait un robot très efficace si l'on y ajoute des petits moteurs lui permettant de marcher sur sol horizontal. Cette pensée raisonnable a rappelé les débuts de l'aviation où des planeurs furent motorisés pour aboutir à l'aviation lourde que nous connaissons. Des chercheurs du monde entier se sont par la suite joints à son idée et de nombreux projets de robots marcheurs semi-passifs ont vu le jour [3, 5, 64].

4.7.2 Oscillateurs non linéaires et CPG

Certains chercheurs considèrent que la marche bipède ne devrait pas être planifiée de manière analytique, mais devrait être représentée par des oscillations non linéaires issues des retours et des interactions dynamiques entre le système et son environnement.

Kato et Mori ont construit le robot échassier BIPMAN2 qui utilise un cycle limite stable généré par un oscillateur non linéaire basé sur les équations couplées de Van der Pol. Ce robot a pu effectuer un pas dynamique en avant [41].

Taga et al. ont simulé un système musculaire humain avec des oscillateurs non linéaires distribués (générateur de modèles central, *Central Pattern Generator* : CPG) et ont observé que le robot peut naturellement générer un mouvement de marche ou de course de manière robuste par rapport aux perturbations [75].

Inspirés par le travail de Taga, Hase et al. ont simulé un modèle humain 3D précis contrôlé par un système CPG hiérarchique. Ils ont effectué une marche et une course 3D stables [20, 21], et aussi utilisé l'algorithme d'escalade pour optimiser les paramètres CPG pour de nombreux indices de performance. Ils ont conclu que les modèles de marche basés sur les propriétés humaines donnaient des résultats très convenables.

4.7.3 Les méthodes d'apprentissage et évolutionnaires

L'approche la plus radicale serait de construire un robot qui puisse apprendre ou auto-générer la marche de lui-même. Doya a construit un bipède simple qui consiste en trois articulations et un algorithme d'apprentissage dans la boucle de contrôle. Le système a généré un modèle de marche aléatoire et a appris par l'algorithme d'escalade en utilisant la distance parcourue comme évaluation de sa performance [8]. Le robot a pu ainsi acquérir une variété de modèles de marche y compris par sautilllements et par acrobaties, ce qui n'était pas du tout attendu. De Garis a conçu une commande de marche basée sur un réseau de neurones dont les poids sont automatiquement définis par un algorithme génétique. Il a simulé un bipède pendant la marche [7].

Tedrake et al. ont réalisé un bipède 3D semi-passif équipé de quatre moteurs dont les mouvements peuvent être obtenus par renfort d'apprentissage en ligne. Il est rapporté que le robot a pu apprendre à marcher convenablement sur des sols de conditions différentes en vingt minutes environ [79].