



**HAL**  
open science

## Simulateur de mouvement pour l'accessibilité

Vincent Gibeaux

► **To cite this version:**

Vincent Gibeaux. Simulateur de mouvement pour l'accessibilité. Biomécanique [physics.med-ph].  
Université Claude Bernard - Lyon I, 2022. Français. NNT : 2022LYO10227 . tel-04390259

**HAL Id: tel-04390259**

**<https://theses.hal.science/tel-04390259v1>**

Submitted on 12 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## **THESE de DOCTORAT DE L'UNIVERSITE CLAUDE BERNARD LYON 1**

**Ecole Doctorale N°162  
Mécanique ; énergétique ; génie civil ; acoustique**

**Discipline** : Biomécanique

Soutenue publiquement le 14/12/2022, par :  
**Vincent Gibeaux**

---

### **Simulateur de mouvement pour l'accessibilité**

---

Devant le jury composé de :

PONTONNIER Charles	Maître de conférences, Ecole Normal Supérieur Rennes	Rapporteur
REZZOUG Nasser	Maître de conférences, Université de Poitier	Rapporteur
BOUAKAZ Saida	Professeure des universités, Université Lyon 1	Présidente
CHEVALLEREAU Christine	Directrice de recherche, CNRS	Examinatrice
DUMAS Raphaël	Directeur de recherche, Université Gustave Eiffel	Directeur de thèse
PRONOST Nicolas	Maître de conférences, Université Lyon 1	Co-directeur de thèse

# Titre : Simulateur de mouvement pour l'accessibilité

## Résumé :

Du fait du vieillissement de la population, l'accessibilité des personnes âgées lors de leurs déplacements est un enjeu grandissant. Avec l'âge, les capacités motrices et cognitives diminuent, ce qui augmente le risque de chute et donc de blessure et de décès. De plus, l'appréhension d'une blessure peut mener à la sédentarité alors que l'activité physique est recommandée pour conserver une bonne santé, aussi bien physique que mentale. Une entrave à l'activité physique des personnes à mobilité réduite est justement l'accessibilité des infrastructures en milieu urbain. En effet, si celles-ci ne sont pas adaptées, elles peuvent devenir un obstacle à la mobilité et donc favoriser la sédentarité. La meilleure solution pour attester de l'accessibilité d'un système est l'expérimentation in-vivo. Cette solution a cependant certaines limitations que ce soit au niveau de la conception du montage qui ne peut pas facilement simuler toutes les configurations possibles, du protocole expérimental qui doit prendre en compte la fatigue des participants, et du recrutement des participants qui peut s'avérer difficile. Une solution axillaire pour augmenter la quantité de résultat ou pour faire office de pré-tests pour aider à concevoir le montage de l'expérimentation est l'utilisation de la simulation numérique. La simulation de la marche est utilisée dans différents domaines tel que la biomécanique, l'infographie ou la robotique. Chaque domaine a ses propres objectifs et contraintes à respecter et fait donc appel à des méthodes de contrôles différentes. L'objectif de cette thèse est de concevoir un simulateur de marche qui doit être capable par la suite de simuler des marches pathologiques lors de passages d'obstacles. Pour ce faire, une méthode de contrôle généralement utilisée en infographie et robotique a été utilisée pour réaliser une cinématique et une dynamique suffisamment biofidèle pour une utilisation dans le domaine de la biomécanique. L'apport de ce travail et de cette approche peut alors être décrit en trois points. Tout d'abord, il permet d'explorer la faisabilité de cette méthode, identifier les problématiques et orienter les choix pour les travaux futurs. Deuxièmement, la conception d'un contrôle adoptant un contrôle différent de ce qui se fait habituellement dans le domaine de la biomécanique permet d'étudier la marche via un nouveau point de vue, en s'intéressant directement aux rôles de chaque articulation et à la manière de les contrôler en fonction de la situation. Enfin, s'il s'avère que cette méthode est faisable et qu'un simulateur complet est conçu, alors il permettra de réaliser des simulations combinant les avantages des méthodes de l'infographie et robotique, à savoir le temps de simulation et intuitivité du contrôle, et les avantages de la biomécanique, à savoir la biofidélité des résultats.

**Mots clefs :** Simulation de la marche, passage d'obstacle, marche pathologique, dynamique directe, contrôle de la cinématique articulaire, briques de contrôle, OpenSim

## **Title: Movement simulator for accessibility**

### **Abstract:**

Due to the aging of the population, accessibility for the elderly when traveling is a growing issue. When aging, motor and cognitive abilities decline, which increases the risk of falling and therefore of injury and death. Moreover, injury apprehension can lead to a sedentary lifestyle, whereas physical activity is recommended to maintain good health, both physical and mental. One inconvenience to the physical activity of people with reduced mobility is the accessibility of infrastructures in urban areas. Indeed, if they are not adapted, they can become an obstacle to mobility and therefore promote a sedentary lifestyle. The best solution to attest to the accessibility of a system is in-vivo experimentation. However, this solution has certain limitations, whether in terms of the design of the experimental set-up, which cannot easily simulate all the possible configurations, the experimental protocol, which must take into account the tiredness of the participants, and the recruitment of the participants, which may prove to be challenging. An auxiliary solution to increase the quantity of results or to act as pre-tests to help design the experimental set-up is the use of simulation. Gait simulation is used in different fields such as biomechanics, computer graphics or robotics. Each field has its own objectives and constraints to respect and therefore uses different control methods. The objective of this thesis is to design a movement simulator which must then be able to simulate pathological walking when passing obstacles. To this purpose, a control method generally used in computer graphics and robotics was used to achieve kinematics and dynamics sufficiently biofaithful for use in the field of biomechanics. The contribution of this work and this approach can then be described in three points. First, it explores the feasibility of this method, highlight potential problems and guide the choices for future work. Secondly, the design of a control adopting a different control from what is usually done in the biomechanics field allows for a new walking study point of view by taking a direct interest in the roles of each joint and the how to control them depending on the situation. Finally, if it turns out that this method is feasible and that a complete simulator is designed, then it will make it possible to carry out simulations combining the advantages of computer graphics and robotics methods, namely the simulation time and intuitiveness of the control, and the advantages of biomechanics, namely the biofidelity of the results.

**Key words:** Gait simulation, obstacle passing, pathological gait, direct dynamics, joint kinematics control, control loop, OpenSim

## Table des matières

I.	Introduction.....	1
II.	Etat de l'art.....	5
A.	Eléments du simulateur.....	5
1.	Modèle.....	5
2.	Le contrôleur.....	14
3.	Le solveur.....	14
B.	Les différents types de simulation de la marche.....	16
1.	Marcheurs passifs.....	16
2.	Pendule inversé.....	17
3.	Contrôle du mouvement par la cinématique articulaire.....	22
4.	Contrôle optimal.....	30
5.	Contrôle par réseau de neurones.....	33
C.	Résumé.....	34
III.	Modèle et contrôleur.....	36
A.	Définitions et notations.....	37
1.	Définitions et notations des paramètres de la marche.....	37
2.	Programmation orientée objet.....	38
B.	Modèle et solveur.....	40
1.	Modèle squelettique.....	40
2.	Modèle de contact.....	43
C.	Détermination des trajectoires et temporalités de référence.....	49
1.	Traitement de la base de données.....	50
2.	Détermination des poses clef.....	55
3.	Modèle statistique.....	78
D.	Contrôle.....	80
1.	Fichier principal.....	80
2.	Détermination des matrices de passage et jacobiniennes.....	87
3.	Contrôle global.....	88
E.	Réglage des paramètres.....	108
1.	Réglage manuel.....	108
2.	Réglage par optimisation.....	108

IV. Optimisations et résultats.....	111
A. Solution Initiale.....	111
B. Tests d'optimisation.....	114
C. Optimisation du contrôle du CoM.....	116
1. Fonction de coût.....	117
2. Résultats.....	118
D. Optimisation du contrôle du talon.....	121
1. Fonction de coût.....	121
2. Résultats.....	123
3. Test du résultat sur deux pas.....	126
E. Optimisation globale.....	130
1. Fonction de coût.....	130
2. Résultats.....	131
V. Acquisition et exploitation de nouvelles données.....	159
A. Motif d'expérimentation.....	159
B. Expérimentation.....	160
C. Article.....	160
1. Introduction.....	160
2. Méthode.....	161
3. Résultats.....	165
4. Discussion.....	167
D. Résultats complémentaires à l'article.....	170
E. Réflexions sur le contrôle lors de passage d'obstacles.....	172
1. Approche.....	172
2. Passage de l'obstacle.....	172
VI. Discussion.....	176
A. Evolution du simulateur.....	176
1. Choix d'un contrôle de base.....	176
2. Tentatives d'implémentations infructueuses.....	177
3. Modifications pérennes du contrôle de base.....	178
B. Difficultés rencontrées.....	181
1. Modèle de contact.....	182

2.	Méthode d'amélioration du contrôleur.....	184
3.	Réglage manuel des paramètres.....	185
C.	Perspectives.....	186
1.	Amélioration du contrôle de base.....	186
2.	Implémentation des pathologies.....	192
VII.	Conclusion.....	195
A.	Complexité du contrôle.....	195
B.	Modèle de contact.....	196
C.	Spécialisation du modèle.....	196
D.	Avantages de la méthode de contrôle.....	197
E.	Contributions.....	198
F.	Mesures expérimentales.....	199
G.	Synthèse.....	199
VIII.	Annexe.....	200
A.	Classe pour la détermination des trajectoires de référence.....	200
B.	Classe du contrôleur proportionnel dérivé.....	202
C.	Classe du contrôle du talon en phase oscillante.....	203
D.	Classe du contrôle du CoM.....	205
E.	Classe de compensation de la gravité.....	207
F.	Classe de calcul des forces de réaction horizontales.....	208
G.	Classe du contrôleur global.....	210
IX.	Références.....	216

# I. Introduction

L'accessibilité lors des déplacements est un enjeu majeur pour les personnes âgées ou à mobilité réduite. Des infrastructures inadaptées, en milieu urbain comme à domicile, peuvent devenir des obstacles à la mobilité et devenir un facteur de risque de blessure ou de sédentarité.

La catégorie de personnes la plus affectée par la perte de mobilité correspond aux personnes âgées. D'après l'Agence Nationale de Santé Publique, 1 personne sur 3 âgée de 65 ans et plus chute chaque année et environ 9300 décèdent des suites de celle-ci. Il existe plusieurs causes possibles à ces chutes tels que des vertiges ou des absences, mais la cause principale reste le trébuchement (Blake et al., 1988). Les facteurs de risque sont eux même multiples. Une partie d'entre eux est d'origine extérieure et correspond à la prise de produits pharmaceutiques tels que des sédatifs, mais le reste correspond à des effets du vieillissement (Tinetti et al., 1988). Tout d'abord, le vieillissement a un effet connu sur les capacités physiques. La perte de force varie de 2,5 à 4% par an pour les personnes âgées de 75 ans (Mitchell et al., 2012), résultant en une augmentation de la difficulté à passer des obstacles de la vie courante. De plus, la capacité à produire de la force explosive diminue elle aussi avec l'âge, réduisant la capacité à répondre à une instabilité (Izquierdo et al., 1999). Deuxièmement, la réduction des capacités cognitives a aussi un impact sur la perte de mobilité. L'activité corticale pendant la marche augmente avec l'âge (Hausdorff et al., 2005; Hoang et al., 2022), indiquant que cette tâche demande de plus en plus de ressources cognitives et devient donc de moins en moins automatique. De plus, avec le vieillissement, peuvent survenir l'apparition de maladies neurodégénératives pouvant perturber la marche telles que la maladie de Parkinson (Tinetti et al., 1988) ou d'Alzheimer (Webster et al., 2006). En effet, ces maladies sont corrélées à une augmentation de la variabilité de la marche (Blin et al., 1990; Hausdorff et al., 1998; Sheridan et al., 2003), qui est associée à une augmentation du risque de chutes (Hausdorff, 2005). Avec le vieillissement de la population, le nombre de personne voyant leur mobilité se réduire va augmenter, rendant les problématiques d'accessibilité de plus en plus importantes. En effet d'après l'INSEE, la part de la population âgée de 65 ans et plus sera de 25% d'ici 2040, contre 20.5% en 2020, et continuera de monter pour atteindre 28% en 2070.

Une autre catégorie de personnes affectées par la perte de mobilité est celle des personnes à mobilité réduite. Une grande partie de ces personnes se déplace en fauteuil roulant manuel ou électrique et n'est pas concernée de la même façon par les problèmes de passage de seuil et de lacune ou par le risque de chute. Ces problèmes en lien avec la marche concernent plus particulièrement les personnes amputées du membre inférieur. Selon que l'amputation soit transtibiale ou transfémorale, la personne perd le contrôle de l'articulation de la cheville et potentiellement de l'articulation du genou. La cheville joue un rôle important dans le maintien de l'équilibre durant la phase de simple appui en permettant notamment de contrôler la position du centre de pression (Tropp & Odenrick, 1988). Pour différentes raisons comme le poids, le prix ou la maintenance, la majorité des prothèses de cheville sont des prothèses passives conçues pour stocker de l'énergie à la pose du pied et la restituer lors de la phase

de poussée et ainsi diminuer la force musculaire requise pour pouvoir avancer. Ce système ne permet cependant pas la même régulation du centre de pression qu'une cheville naturelle. Une personne amputée de la cheville est alors privée de tout un pan des stratégies de rattrapage de l'équilibre dont dispose normalement une personne saine. Dans le cas d'une amputation transfémorale, en plus des conséquences de la perte de la cheville, le contrôle du genou est évidemment réduit à un comportement passif, mais l'articulation de la hanche est elle aussi affectée. En effet, les points d'insertion des muscles moteurs de la hanche sont modifiés, résultant en des muscles plus courts et moins puissants. C'est notamment le cas des abducteurs de hanche, responsables entre autres de la stabilisation du bassin pendant la phase de simple appui (Ryser et al., 1988). Ainsi, une amputation transfémorale, en plus de priver du contrôle de l'équilibre par la cheville, réduit aussi la capacité de la hanche à contrôler l'équilibre, résultant en une altération de la marche plus importante (Highsmith et al., 2010). Les conséquences d'une amputation du membre inférieur sont alors similaires à celles du vieillissement. La variabilité des paramètres de la marche augmente (Beausoleil et al., 2019), les risques de chute augmentent également (Hunter et al., 2017), et la mobilité est réduite (Beausoleil et al., 2019; Davies & Datta, 2003).

Mais les chutes ne sont pas le seul risque d'une perte de mobilité. Des difficultés à se déplacer et une peur de la chute peuvent mener à la sédentarité. Il est admis que la pratique d'une activité physique est bénéfique pour la santé (Bauman, 2004; Press et al., 2003), y compris chez les personnes âgées (Aoyagi & Shephard, 2010). La sédentarité est connue pour être un facteur de risque lié à la mortalité (Rezende et al., 2014). L'activité physique a également une influence sur les capacités mentales des personnes âgées. Elle semble en effet avoir un effet bénéfique sur les capacités cognitives (Busse et al., 2009). Enfin, la perte de mobilité peut aussi avoir des conséquences psychologiques. Chez les personnes âgées, une perte de mobilité est synonyme de perte d'indépendance et de passage en maison de retraite (Finlayson & van Denend, 2003) alors qu'une activité régulière permet notamment d'améliorer le comportement (Landi et al., 2004) et de retarder le passage en maison de retraite. Chez les personnes amputées, les efforts à la fois physique et psycho-sociaux nécessaires à la réhabilitation en société peuvent provoquer un isolement social (Pell et al., 1993; Thompson & Haran, 1983).

S'assurer de l'accessibilité des infrastructures est donc à la fois un moyen de réduire le nombre de chutes mais aussi de ne pas dissuader les personnes à mobilité réduite ou avec une faiblesse musculaire de se déplacer, leur permettant de garder une meilleure forme physique, de meilleures capacités cognitives et un meilleur moral. Un certain nombre de recommandations existe pour concevoir un environnement urbain (bâtiments, voirie, transports en commun) qui soit accessible au plus grand nombre. La loi pour l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées du 11 février 2005 a créé l'obligation de mise en accessibilité des bâtiments et des transports dans un délai maximum de 10 ans. L'arrêté du 13 juillet 2009 a donné des dispositions techniques destinées à faciliter la mobilité des personnes en situation de handicap (taille des lacunes et marchepieds, pente des rampes). Néanmoins, il existe peu de données concernant l'adéquation entre ces dispositions techniques (par exemple, des lacunes horizontales maximales de 50 mm ou des pentes maximales de 8%) et les difficultés encore rencontrées par les personnes à mobilité réduite lors de leurs déplacements urbains. L'évaluation, a priori ou a posteriori, d'un environnement possédant des seuils, des lacunes, des rampes ou même des escaliers ou planchers roulants, n'est pas une chose simple.

La méthode la plus courante pour acquérir des données est l'expérimentation in-vivo. Il est possible de créer en laboratoire un montage permettant de mettre les sujets dans la ou les situations à évaluer, et de recruter les sujets parmi la population que l'on souhaite étudier. Il est alors possible d'obtenir des données qualitatives (tâches réussies ou non), voire quantitatives grâce à la capture de mouvement (avec ou sans marqueur), à des plateformes de force, des centrales inertielles ou à des EMG si l'on veut en sus étudier les mécanismes ayant conduit au succès ou à l'échec des tâches demandées. Le plus grand avantage de cette méthode est la fidélité des données obtenues qui n'est limitée que par la précision des outils de mesure. Cependant, elle présente également des inconvénients non négligeables à prendre en compte lors de la conception de l'expérimentation. Le premier concerne le montage. Sa conception se fait normalement après avoir défini les tâches à tester. Cependant, selon le type et le nombre de tâches que l'on souhaite réaliser, concevoir un montage pour répondre aux besoins de l'étude peut devenir contraignant. Les choix de conception de la maquette peuvent alors influencer l'étude elle-même. Le second inconvénient concerne les sujets. Selon le type de population que l'on souhaite étudier, trouver des sujets volontaires peut s'avérer difficile, et le nombre final de sujets disponibles peut être insatisfaisant par rapport aux attentes de l'investigateur, notamment en termes de représentativité ou même de puissance statistique. De plus, dans le cas où l'on étudie une population ayant précisément des difficultés à se mouvoir, une expérimentation mal conçue peut devenir physiquement éprouvante, ce qui peut introduire des biais dans les résultats et demander d'adapter l'expérimentation en fonction de ces contraintes.

Une autre méthode pour acquérir des données, rendue possible grâce aux avancées des capacités de calcul des ordinateurs, est l'expérimentation in-silico. Au moyen de modèles numériques, il est possible de simuler les tâches et les populations souhaitées sur ordinateur. L'avantage de cette méthode est que l'on réduit le besoin de maquette physique et de sujets volontaires, et donc de toutes les contraintes associées. De plus, le nombre de situations qu'il est possible de tester, que ce soit au niveau des tâches à réaliser ou au niveau du type et de la gravité d'une pathologie, devient virtuellement illimité. Evidemment, cette méthode comporte des inconvénients non négligeables, notamment au niveau de la modélisation. Par définition, un modèle est une approximation de la réalité, généralement basée sur des hypothèses simplificatrices. Ainsi, les résultats obtenus par simulation seront forcément moins fidèles à la réalité que ceux obtenus par expérimentation in-vivo, et seront fortement dépendant de la qualité du modèle et de la simulation.

Ainsi, dans le cas d'une étude sur l'accessibilité, un simulateur de mouvement n'aura pas pour objectif de remplacer complètement les expérimentations in-vivo. L'objectif premier sera au contraire de compléter les résultats obtenus par celle-ci. Une autre utilisation possible sera la prédiction de résultats d'une expérimentation in-vivo. En effet, simuler une tâche avec différentes configurations pourrait permettre de faire ressortir les configurations les plus intéressantes et donc de concentrer l'expérimentation in-vivo sur celles-ci. Enfin, un autre type de résultat pourra venir de la conception du modèle numérique lui-même. Réussir à concevoir un modèle dynamique capable de marcher en faisant appel à un ensemble d'hypothèses peut être considéré comme un moyen différent d'étudier les mécanismes de la marche, de l'équilibre et des pathologies par rapport aux moyens utilisés classiquement en biomécanique.

L'objectif de cette thèse est donc de concevoir un simulateur de mouvement capable de simuler dans un premier temps des marches non pathologiques. A terme, il devra être capable de simuler la marche pathologique dans des situations potentiellement problématiques pour l'accessibilité et doit donc être pensé pour pouvoir s'adapter à ces tâches.

Suite à cette introduction, le deuxième chapitre de ce manuscrit présente l'état de l'art concernant la simulation de la marche, les différentes approches existantes, ainsi que les éléments composant un simulateur de mouvement et présente l'approche qui a été choisie pour cette thèse.

Le troisième chapitre présente en détails les caractéristiques du modèle et le fonctionnement du simulateur.

Le quatrième chapitre présente le processus d'optimisation pour déterminer les paramètres du contrôleur ainsi que les résultats du simulateur.

Le cinquième chapitre présente une expérimentation in-vivo effectuée durant la thèse. Il précise la nécessité de celle-ci et inclue un article scientifique présentant des résultats obtenus.

Le sixième chapitre présente une discussion sur l'évolution du simulateur au cours de sa conception, les difficultés notables rencontrées au cours de celle-ci, ainsi que les perspectives possibles pour continuer la conception du simulateur.

Enfin, le dernier chapitre propose une conclusion globale à ce projet.

# II. Etat de l'art

Bien que ce projet de thèse vise à terme à simuler des situations d'accessibilité à des bâtiments ou transports impliquant des seuils, lacunes, rampes voire des escaliers ou planchers roulants, l'élément central de la simulation reste la marche. Différents types de simulation de la marche humaine existent. Les différences concernent généralement le modèle utilisé pour représenter le corps humain et la méthode pour le mettre en mouvement. Le choix du modèle et de la méthode de pilotage dépendent notamment des objectifs de la simulation. Ainsi, une étude dont l'objectif est uniquement la mise en marche et la stabilité d'un bipède privilégiera un modèle et une méthode de pilotage relativement simple en comparaison des autres méthodes existantes. Une étude dont un objectif supplémentaire est le contrôle en temps réel et physiquement réalisable devra s'assurer que la méthode de pilotage en est capable. Enfin, une étude portée sur la compréhension du mouvement jusqu'au niveau musculaire devra utiliser un modèle détaillé ainsi qu'une méthode de pilotage capable de gérer l'ensemble des éléments du modèle choisi, notamment au niveau du nombre et du type de variables à contrôler. Ce chapitre aura donc pour objectif d'explicitier les éléments d'une simulation de la marche et leurs variantes, ainsi que de résumer les différentes méthodes de contrôles utilisées dans la littérature.

## A. Eléments du simulateur

### 1. Modèle

Le modèle représente l'ensemble des caractéristiques de l'objet physique que l'on souhaite simuler. Le nombre et la nature de ces caractéristiques vont dépendre de plusieurs facteurs tels que les hypothèses simplificatrices ou les objectifs de la simulation. En effet, différents domaines de recherche tels que la biomécanique, l'infographie et la robotique s'intéressent à la simulation de la marche, avec des contraintes et des objectifs différents.

Tous les modèles présentés ci-après sont des modèles multi-corps rigides polyarticulés. Cela signifie qu'ils sont constitués de solides indéformables et liés entre eux par différents types de liaisons (pivot, rotule, prismatique). Les solides représentent les différents segments du corps humain tandis que les liaisons représentent les articulations.

## 1.1. Propriétés du modèle

La première caractéristique d'un modèle de bipède pour la simulation de la marche est le nombre de degrés de liberté (DoF) du système et par conséquent le nombre de segments. Il dépend aussi du nombre d'articulations et de leur type. De plus la simulation peut être faite en 2D ou en 3D ; les simulations 2D étant généralement contraintes dans le plan sagittal pour ignorer les composantes médio-latérales.

Le modèle le plus simple en simulation numérique est celui du pendule inversé linéarisé en 2D (Kajita & Tani, 1991) consistant en une masse ponctuelle de hauteur fixe et liée au sol par un membre inférieur considéré sans masse et de longueur variable (Figure 1). Cette configuration permet de ne considérer que les DoF s'apparentant à la flexion de la hanche et simplifie donc grandement la simulation. En contrepartie, le modèle ne permet pas une représentation fidèle de la marche humaine et ne permet de simuler la marche que par successions de phases de simple support. Le pendant de ce modèle dans le domaine de la robotique est celui du « compass gait » en 2D (Kuo, 2001) qui reprend le concept de masse ponctuelle mais non contrainte en hauteur, et de membre inférieur sans masse mais de longueur fixe (Figure 1). Bien que deux configurations similaires, ces deux modèles sont utilisés dans des types de contrôles différents, explicités dans la partie II.B.1.

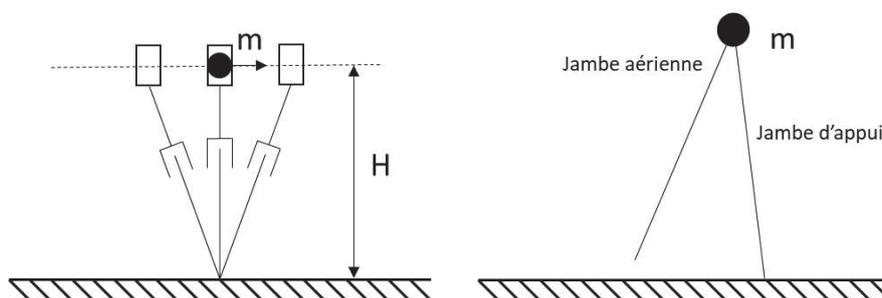


Figure 1 : Modèle de pendule inversé à hauteur fixe et du « compass gait »

Bien que la formulation de base du modèle de pendule inversé considère des membres inférieurs sans masse, certains modèles les prennent tout de même en compte en considérant des masses ponctuelles à chaque membre inférieur en plus de celle supposée représenter le centre de masse de toute la partie supérieure du corps humain dans cette version (Goswami et al., 1998) ou en utilisant un pendule double dans lequel la deuxième masse correspond au membre inférieur en phase oscillante (Park & Kim, 1998) (Figure 2). Associer une masse aux segments permet alors de prendre en compte la dynamique du système telle que l'effet inertiel de la projection en avant du membre inférieur et d'obtenir une marche plus stable (Albert & Gerth, 2003).

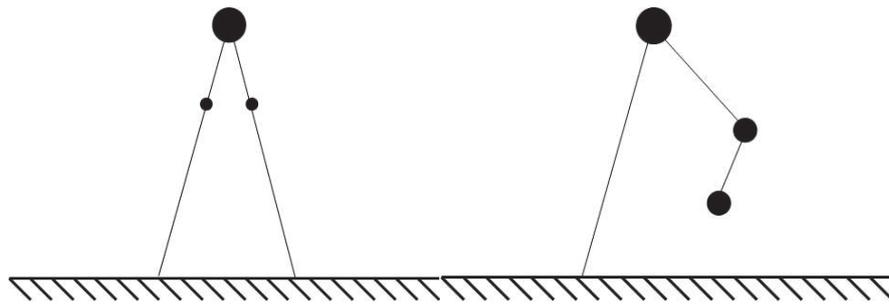


Figure 2 : Modèle de pendule inversé avec des masses et segments supplémentaires

Pour avoir la possibilité d'obtenir une marche davantage biofidèle, il est possible de complexifier le modèle en ajoutant des articulations. Après celle de la hanche, l'articulation la plus utilisée dans les modèles est celle du genou. Elle peut être modélisée par une articulation prismatique permettant au membre inférieur de varier de longueur au cours du cycle de marche (Gubina et al., 1974; Katoh & Mori, 1984) ou par une liaison pivot (McGeer, 1990) permettant notamment d'éviter le contact entre le membre inférieur en phase oscillante et le sol.

Les modèles de pendule inversé considèrent généralement un contact ponctuel et fixe avec le sol. Les simulations consistent alors généralement en une phase de simple appui ou en la succession de phases de simple appui avec une transition immédiate entre les points d'appui. Pour simuler une marche avec des phases de double appui, il faut donc modifier le contact. Cela se fait en modifiant la forme du segment en contact avec le sol ou en ajoutant un segment ayant le rôle du pied, et donc en ajoutant, en 2D, un degré de liberté identifiable à l'articulation de la cheville. Dans ce cas, l'appellation de modèle de pendule inversé n'est plus propice et l'on peut utiliser l'appellation de modèle squelettique. Dans un modèle squelettique, les différents solides sont assimilés à des segments du corps (cuisse, jambe, pied...). La modélisation se veut plus réaliste grâce à une représentation plus fidèle de la cinématique mais aussi de la dynamique en prenant mieux en compte la répartition de la masse dans chaque segment. Les modèles présentés par la suite seront donc des modèles squelettiques.

Ainsi, le point d'amélioration suivant concerne l'interaction entre le modèle et le sol. Dans les exemples cités précédemment, celle-ci était considérée comme un contact ponctuel pour laquelle le glissement n'est pas autorisé et l'articulation de la cheville n'est pas contrôlée. Dans le cas où la cheville n'est pas destinée à être contrôlée, une solution simple pour améliorer le contact est de modéliser le pied par un solide semi-circulaire (Borzova & Hurmuzlu, 2004; McGeer, 1990) (Figure 3). Le contact entre le modèle et le sol reste ponctuel mais le point de contact est alors autorisé à se déplacer sans que le modèle ne glisse sur le sol. Cependant, pour une meilleure biofidélité, il est possible d'ajouter une articulation entre le segment de la jambe et le segment du pied afin de modéliser la cheville et de pouvoir la contrôler (Figure 3).

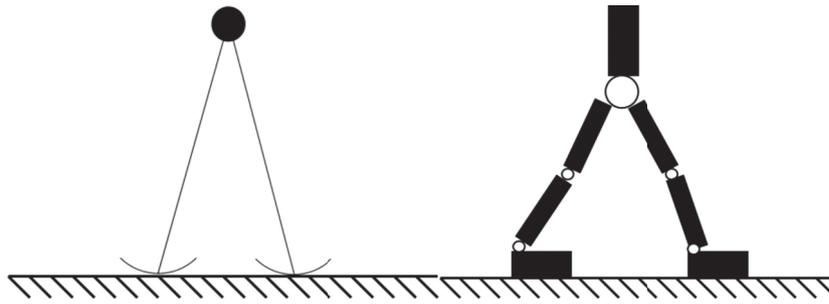


Figure 3 : Modèle de pendule inversé avec pied en arc de cercle et modèle squelettique

Ensuite, il est également possible de prendre en compte les actions des membres supérieurs (Collins et al., 2001). En effet, la marche provoque un moment cinétique qui est en partie compensé par le balancement des bras (Herbert, 1939). Prendre en compte le haut du corps permet donc d'améliorer la stabilité du modèle pendant la simulation.

Enfin, le dernier point qui gouverne le nombre de DoF est la contrainte imposée ou non au modèle d'être fixe dans un plan, généralement le plan sagittal. Dans le cas d'une simulation 2D, seules les articulations dont l'axe de rotation est normal au plan sagittal sont prises en compte. En 3D, certaines articulations, comme la hanche et la cheville, disposent de plusieurs DoF qu'il faut alors prendre en compte. L'ensemble des DoFs se composent également de la position et de l'orientation globale du modèle dans un repère fixe généralement lié au sol. En 2D, ces DoF sont la position dans la direction antéropostérieure, la position dans la direction verticale et l'inclinaison sagittale, d'un segment de la chaîne qui est généralement le segment bassin. En 3D, les trois autres DoF, à savoir la position médio latérale, l'inclinaison frontale et la rotation axiale du bassin, s'ajoutent aux DoF déjà listés.

Comme explicité précédemment, ce sont surtout les besoins et les objectifs de la simulation qui déterminent le nombre de segments et de DoF. On peut comparer par exemple un robot bipède contraint dans le plan sagittal et conçu pour tester une méthode de contrôle de la marche (Pratt et al., 2001) et un modèle numérique 3D pour simuler un saut (Anderson & Pandy, 1999). Le robot bipède est constitué de sept segments représentant le haut du corps, les cuisses, les jambes et les pieds ainsi que de six articulations représentant les hanches, les genoux et les chevilles, avec des axes de rotation tous normaux au plan sagittal. L'objectif n'étant pas la biofidélité du modèle ou du résultat, les auteurs ont pu se permettre d'utiliser un modèle de bipède simple. Au contraire, le modèle numérique pour étudier le saut a été conçu pour étudier précisément un mouvement humain, et a donc besoin d'être davantage biofidèle. Ainsi, il est composé de dix segments représentant le tronc, le pelvis, les cuisses, les jambes, les pieds et les orteils. Le modèle étant en 3D, les articulations lombaires, des hanches et des chevilles se composent de plusieurs DoF (respectivement trois, trois et deux). La modélisation en 3D permet de plus à certains de ces DoF d'avoir des axes de rotation ne coïncidant pas avec les axes des segments auxquels ils sont liés, notamment ceux des chevilles et des métatarso-phalangiennes, pour plus de fidélité à la réalité anatomique. Dans certains cas, les modèles peuvent encore s'approcher de la réalité comme Hamner et al. (Hamner et al., 2010) qui définissent le genou comme étant une articulation à un degré de liberté, mais dont la position de l'axe de rotation est couplée à la valeur de la rotation, ou Guess et al (Guess et al., 2014) qui pour étudier précisément les charges dans le genou durant la marche, définissent

un genou déformable à douze degrés de libertés pour inclure, dans un modèle musculosquelettique, les articulations tibio-fémorale et patello-fémorale.

## 1.2. Les actionneurs

Dans un modèle, les actionneurs ont pour rôle de mettre en mouvement les segments les uns par rapport aux autres. Ils sont l'équivalent des muscles chez l'être humain. Certaines méthodes de simulation de la marche ne font pas appel à des actionneurs mais ne fonctionnent que dans des situations précises, i.e. marche sur une pente (cf. partie II.B.1.). La grande majorité des modèles fait donc appel à des actionneurs qui peuvent être de plusieurs types. Les simulations dont un objectif est la biofidélité du contrôle utilisent des actionneurs de types lignes d'action qui représentent les forces musculaires ou directement leur action sur les articulations via des couples articulaires. Les contrôles dont la biofidélité n'est pas un objectif peuvent appliquer des forces ou des couples externes sur le modèle, notamment pour le maintenir en équilibre. Cependant, ces actions n'ayant pas d'origine réel (muscle, réaction du sol, gravité), elles ne sont pas traitées.

La première solution consiste à modéliser les systèmes musculo-tendineux par des lignes d'action et de piloter ces actionneurs en force, activation, voire excitation neurale. De la même manière que les muscles dans le corps humain, les lignes d'action sont définies par une origine et une insertion et, si besoin, peuvent suivre une trajectoire spécifique le long du modèle, c'est-à-dire en passant par des « via points » ou s'enroulant sur des géométries de contournement. Quand le pilotage n'est pas directement réalisé en force, la modélisation la plus courante est celle du muscle de type Hill qui permet de relier l'activation musculaire (bornée entre 0 et 1) à la force générée (Figure 4). Le muscle modélisé est composé d'un élément contractile et d'un ressort en parallèle pour représenter la force et l'élasticité du muscle, et d'un ressort en série pour représenter l'élasticité des tendons (Hill, 1938; van den Bogert et al., 1998).

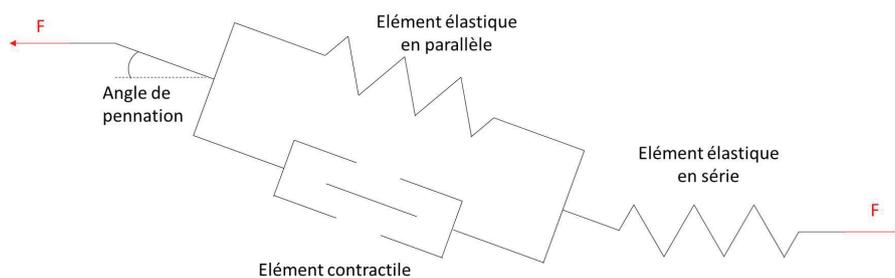


Figure 4 : Muscle de Hill

Par la suite, des variations de ce modèle ont été proposées afin de l'améliorer (Arslan et al., 2019) en faisant varier le nombre d'éléments et le fait qu'ils soient en séries ou en parallèle. Les différents paramètres de la modélisation du muscle (comme par exemple la raideur non-linéaire du tendon) sont déterminés expérimentalement pour chacun des muscles. L'utilisation des forces musculaire (directement ou en passant par l'activation) comme actionneur présente divers avantages. Tout d'abord,

elle permet une modélisation biofidèle de la manière dont sont actionnées les articulations chez l'humain. En effet, en faisant concorder les chemins par lesquels passent les forces avec ceux connus (voire mesurés individuellement) des muscles réels, il est possible de reproduire la même configuration de force dans le modèle que dans la réalité. Cela est notamment utile pour la prise en compte du rôle des bras de levier dus à la position des origines et insertions, ainsi que et surtout de l'effet des muscles polyarticulaires sur les articulations qu'ils traversent. De cette modélisation biofidèle du système musculosquelettique découle alors la possibilité d'étudier précisément le rôle d'un muscle ou l'impact d'une modification d'un muscle sur une tâche. Cependant, cette modélisation est aussi la source de l'inconvénient majeur de cette solution. En effet, le pilotage d'une articulation est géré par plusieurs muscles en même temps. Une modélisation utilisant des forces musculaires comme actionneurs et cherchant la biofidélité se verra alors sous-déterminé, et le rôle isolé d'un muscle dans la mise en mouvement d'un DoF d'une articulation sera impossible à définir sans connaissances a priori et sans opérations numériques coûteuses en temps de calcul. Les forces musculaires comme actionneurs sont donc majoritairement utilisées dans le domaine de la biomécanique où l'objectif d'une simulation est généralement l'étude des causes d'un mouvement, le rôle d'un muscle sur un mouvement ou l'impact d'une modification d'un muscle sur un mouvement après une chirurgie (Anderson & Pandy, 1999; Glitsch & Baumann, 1997; Guess et al., 2014). Bien que moins répandu en infographie, cette méthode a tout de même été utilisée dans différentes études du domaine (Geijtenbeek et al., 2013; Lee et al., 2014; J. M. Wang et al., 2012).

Le second type d'actionneur possible est le couple articulaire. Dans cette approche, un couple est associé à chaque DoF lié à une articulation et agit comme un servomoteur sur les segments auxquels il est attaché (Jie Tan et al., 2011; Pratt et al., 2001). Dans une simulation de la marche humaine, ces couples représentent les moments résultant des différents muscles traversant l'articulation. Dans le domaine de la robotique, ils représentent directement les servomoteurs réellement utilisés dans les robots. L'avantage de cette solution pour la simulation est qu'il y a moins de variables à contrôler qu'en utilisant des forces musculaires. Puisque chaque DoF est contrôlé par un seul couple, le système n'est pas sous-déterminé, ce qui permet au contrôle d'utiliser moins de ressources de calcul et d'atteindre plus facilement un contrôle en temps réel. Cependant, les actions des différents muscles étant fusionnées en un seul couple par DoF, une simulation ayant recours à ce type d'actionneur permettra d'obtenir des résultats différents d'une simulation utilisant des forces musculaires. Il est aisé d'obtenir les couples articulaires à partir des forces musculaires mais l'inverse se heurte à la sous-détermination mentionnée plus haut et requiert à nouveau des calculs numériquement coûteux. Les couples articulaires comme actionneurs sont donc majoritairement utilisés dans les simulations pour lesquelles aucune donnée sur les muscles ne sont attendues. Cela est notamment le cas dans le domaine de l'infographie pour lequel l'utilisation des couples articulaires permet le contrôle en temps réel d'un personnage virtuel (Coros et al., 2010; Hodgins & Pollard, 1997; Jie Tan et al., 2011; Tsai et al., 2010). C'est également le cas dans le domaine de la robotique puisque les robots utilisent effectivement des couples articulaires pour se mouvoir (Azevedo et al., 2004; Hirai, 1999; Pratt et al., 2001).

### 1.3. Le modèle de contact

Un point important dans une simulation basée sur la physique est la modélisation des interactions entre le modèle et son environnement. Dans le cas de la marche, les deux interactions à prendre en compte sont la force de gravité et les forces de réaction du sol. Pour que la simulation soit réussie, il faut que le modèle reste stable. Les forces musculaires ou les couples articulaires doivent donc répondre aux éventuelles perturbations pour assurer la stabilité du modèle. Or, une mauvaise modélisation du contact avec le sol peut entraîner des perturbations non naturelles qui doivent alors être compensées par les actionneurs d'une manière non naturelle. De plus, les 3 ou 6 DoF correspondant à la position et l'orientation globale du modèle n'étant pas directement contrôlés, c'est l'action combinée des moments articulaires et des forces de réaction du sol qui permettent de les contrôler indirectement. Il est donc indispensable pour obtenir un résultat correct de modéliser le plus fidèlement possible les interactions entre les pieds et le sol.

Une première méthode pour calculer les forces de réaction du sol est celle du contact rigide. Elle consiste en l'application de contraintes cinématiques qui se répercutent sur les équations du mouvement (Hamner et al., 2013; Lin et al., 2011). En contraignant la position du point de contact entre le pied et le sol et en connaissant les équations du mouvement, il est possible de déterminer exactement les forces de contact permettant au point de contact de maintenir cette position. Cette méthode montre des résultats proches de la réalité mais demande de connaître à l'avance le point d'application des forces de réaction.

Afin de pouvoir calculer la force de réaction du sol sans a priori, il est possible d'utiliser divers éléments produisant une force en fonction de la position et de la vitesse du pied par rapport au sol, similairement à un système ressort-amortisseur. L'ensemble des utilisations de ces éléments est regroupé sous l'appellation de « méthode de pénalité » (Flores et al., 2011) et sera traité ci-après.

La base de la plupart des modèles de contact s'appuie sur la loi d'Hertz décrivant la force de contact par une fonction non linéaire de la déformation :

$$F_n = K \delta^n$$

$K$  représente la rigidité généralisée des corps en contact. Il dépend de la propriété des matériaux des corps, mais aussi de leurs propriétés géométriques.  $\delta$  représente la déformation relative des corps en contact. Quant à l'exposant  $n$ , il est généralement égal à 3/2 (Flores et al., 2011). Ce modèle représente donc le contact comme un ressort non linéaire mais sans prendre en compte les effets d'amortissements. Afin de remédier à ce manque, le modèle de Kelvin-Voigt propose de modéliser le contact en associant un ressort linéaire avec un amortisseur linéaire (Khulief & Shabana, 1987) :

$$F_n = K\delta + D\dot{\delta}$$

Ici,  $D$  est un coefficient d'hystérésis et  $\dot{\delta}$  est la vitesse relative normale des corps. Cependant, l'ajout du terme d'amortissement introduit deux problèmes majeurs. Premièrement, la force au moment du

contact est censée être nulle. Or, puisque la vitesse de contact prise en compte dans le terme d'amortissement ne l'est pas, la force de réaction calculée ne l'est pas non plus. Et deuxièmement, à la fin du contact, la vitesse relative des corps étant de signe opposé par rapport au début du contact, la force calculée l'est aussi et tend à faire s'attirer les corps. Hunt et Crossley (Hunt & Crossley, 1975) et Lankarani et Nikravesh (Lankarani & Nikravesh, 2021) ont également proposé un modèle améliorant celui d'Hertz en y ajoutant un amortissement non linéaire :

$$F_n = K \delta^n + \chi \delta^n \dot{\delta} \quad (1)$$

$\chi$  est appelé le facteur d'amortissement hystérétique. Hunt et Crossley, et Lankarani et Nikravesh le définissent respectivement comme :

$$\chi = \frac{3K(1-c_r)}{2\delta^{\frac{1}{2}} \dot{\delta}}$$

et

$$\chi = \frac{3K(1-c_r^2)}{4\delta^{\frac{1}{2}} \dot{\delta}}$$

$c_r$  est le coefficient de restitution et  $\dot{\delta}$  est la vitesse de contact initiale. L'avantage de ce modèle est que l'amortissement étant fonction de la déformation, il n'y a pas de discontinuité de force au moment de l'impact et de la fin du contact.

Un des modèles de contact le plus couramment utilisée dans la littérature consiste à calculer les forces de contact par des éléments viscoélastiques ponctuels. Anderson et Pandy (Anderson & Pandy, 1999) utilisent un modèle à 5 éléments dans lequel la force de réaction de la partie « ressort » est exponentielle à la distance entre les éléments ponctuels et le sol. La force de réaction n'étant jamais nulle, cela permet d'éviter les discontinuités au moment du contact ; et la force étant exponentielle à la distance avec le sol, elle est considérée comme négligeable lorsque le pied s'éloigne du sol. McLean (McLean et al., 2003) utilise un modèle personnalisé à 35 éléments constitués d'un ressort linéaire mais avec un élément dissipatif dépendant de la distance d'interpénétration. D'autres études utilisent elles aussi des modèles personnalisés mais ressemblant davantage aux modèles de Hunt-Crossley et Lankarani-Nikravesh puisque les modèles se composent d'un ressort non linéaire et d'un élément dissipatif en fonction de la distance d'interpénétration (Ackermann & van den Bogert, 2010; Gerritsen et al., 1995; Peasgood et al., 2007) avec respectivement 10, 1 et 2 éléments. Enfin, Miller (Miller, 2014) utilise directement le modèle de Hunt-Crossley avec 5 éléments. Cette solution est simple à implémenter mais présente un inconvénient majeur dû au caractère ponctuel des éléments qui ne permettent pas de modéliser correctement le contact sur des surfaces importantes. Pour répondre à ce problème, une solution habituelle consiste à augmenter le nombre d'éléments sous le pied comme (Jackson et al., 2016; Mahboobin et al., 2010; Neptune et al., 2000) qui utilisent respectivement des modèles à 38, 89 ou 72 et 66 éléments.

Une méthode pour améliorer le modèle de contact sans multiplier le nombre d'éléments consiste à remplacer les éléments ponctuels par des géométries telles que des sphères, cylindres ou

ellipsoïdes. Chaque élément produit une force ponctuelle mais l'utilisation des géométries de contact permet de déterminer la position du point d'application en fonction de la situation. La force est calculée selon une généralisation des modèles de Kelvin-Voigt (Cenk Güler et al., 1998), Hunt-Crossley (Dorn et al., 2015; Moreira et al., 2009) ou de Lankarani et Nikravesh (Pàmies-Vilà et al., 2012). L'intersection créant une surface de contact, cette modélisation permet une plus grande adaptation à la géométrie du sol et du pied.

La méthode du contact volumétrique fait elle aussi appel à des géométries de contact telles que des sphères ou des ellipsoïdes mais se base sur le modèle de fondation élastique qui considère la surface de contact comme un ensemble continu de ressorts (Boos & McPhee, 2013; Brown & McPhee, 2018; Shourijeh & McPhee, 2015). Ainsi, c'est le volume d'interpénétration (Figure 5) qui est pris en compte pour calculer les forces de réaction plutôt que la distance d'interpénétration :

$$F_n = K_v V (1 + a_v v_{cn})$$

Dans ce calcul,  $V$  est le volume d'interpénétration,  $v_{cn}$  est la vitesse normale relative des deux corps,  $K_v$  est la rigidité volumétrique et  $a_v$  une constante d'amortissement. Puisque cette méthode est basée sur le modèle de fondation élastique, elle calcule une pression sur l'ensemble de la surface de contact plutôt qu'une force ponctuelle et permet donc l'utilisation de géométries de contact plus complexes.

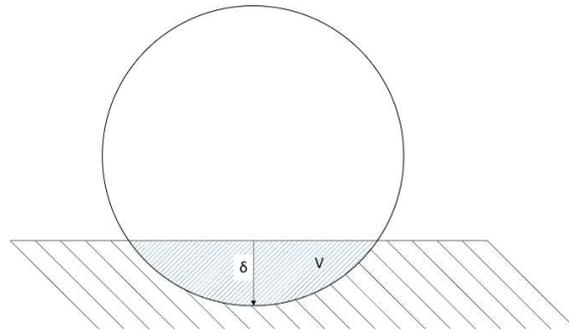


Figure 5 : Schéma des volumes de contact avec la distance et le volume d'interpénétration

Les modèles de contact cités précédemment servent à calculer la force de réaction normale du contact. Dans le cas de la simulation de la marche il faut aussi calculer les forces de frictions. Anderson et Pandy (Anderson & Pandy, 1999) les déterminent par un simple système de ressort amortisseur et les limitent à 0.7 fois la valeur de la force normale. Les autres études utilisent une approximation du modèle de friction de Coulomb. Par exemple, celle utilisée par McLean (McLean et al., 2003) est définie ainsi :

$$F_f = \begin{cases} \mu F_n \text{ si } v_x < \frac{-\mu F_n}{b} \\ -b v_x \text{ si } -\frac{\mu F_n}{b} < v_x < \frac{\mu F_n}{b} \\ -\mu F_n \text{ si } v_x > \frac{\mu F_n}{b} \end{cases}$$

$F_n$  est la force normale,  $\mu$  le coefficient de friction,  $v_x$  la vitesse de l'élément attaché au pied par rapport au sol et  $b$  une constante positive. Dans ce cas, la force de friction est uniquement dépendante de la vitesse de glissement du pied sur le sol et non de sa position. Tous les modèles ont une similarité qui est de limiter la force de friction maximale à la force normale multipliée par un coefficient de frottement. Dans ce cas, le modèle d'Anderson et Pandy (Anderson & Pandy, 1999) qui contrôle aussi la position des éléments autorise le glissement du pied sur le sol en modifiant la longueur au repos des ressorts.

## 2. Le contrôleur

Dans une simulation ou dans un robot, le contrôleur a pour rôle de piloter les actionneurs du modèle. Différents types de contrôle existent ayant chacun leurs particularités et étant utilisés en fonction de l'objectif d'étude (Cruz Ruiz et al., 2017; Geijtenbeek & Pronost, 2012; Xiang et al., 2010). Ceux-ci peuvent être tout d'abord classés en deux grandes familles : les contrôleurs à boucle ouverte et les contrôleurs à boucle fermée.

Un contrôleur est dit en boucle ouverte ou « open loop » lorsque l'activation des actionneurs suit un schéma d'activation défini à l'avance par l'utilisateur sans en dévier. La mise en pratique d'un contrôleur en boucle ouverte est simple puisqu'elle ne requiert pas la définition d'un système de boucle de rétroaction « feedback ». En revanche, l'incapacité d'un tel contrôleur à corriger une erreur non planifiée le rend extrêmement sensible aux changements d'états initiaux, aux changements du modèle ou aux perturbations et instabilités numériques durant la simulation (Peasgood et al., 2007). Il est alors défini comme peu robuste.

Un contrôleur est dit en boucle fermée ou « closed loop » lorsque le contrôle du système dépend en partie de son état actuel. Dans ce cas, le contrôleur a pour objectif de suivre ou d'atteindre un objectif en activant les actionneurs en fonction, entre autres, de l'état du système. Ainsi, le contrôleur pourra s'adapter à une modification de l'état initial du système, du modèle ainsi qu'à d'éventuelles perturbations. Dans la recherche, que ce soit dans les domaines de la biomécanique, de l'infographie ou de la robotique, la simulation étant utilisée pour étudier ou prédire un mouvement, le développement de contrôleurs s'est naturellement concentré autour de contrôleurs en boucle fermée. De ce fait, l'intégralité des contrôleurs discutés ci-après est en boucle fermée ou mixtes si le contrôleur utilise à la fois un schéma d'activation prédéfini et des boucles de contrôles fermés. Cependant, les différents types de contrôleurs étant fortement liés aux méthodes de simulation, ceux-ci seront explicités dans la partie II.B.

## 3. Le solveur

Dans le cas d'une simulation en dynamique directe, le mouvement n'est pas connu à l'avance. Il est donc nécessaire de calculer l'état du modèle à chaque nouveau pas de temps de la simulation.

Connaissant les couples articulaires à chaque DoF, ainsi que les paramètres de masses et d'inerties des segments du système, il est possible d'utiliser le principe fondamental de la dynamique pour calculer l'accélération induite par les couples sur les segments. L'équation du mouvement du système est décrite par :

$$\tau(t) = M(q)\ddot{q}(t) + C(q, \dot{q}) + G(q)$$

Dans cette équation,  $\tau$  est l'ensemble des couples articulaires,  $q$ ,  $\dot{q}$  et  $\ddot{q}$  sont respectivement les positions, vitesses et accélérations angulaires,  $t$  est le temps,  $M$  est la matrice d'inertie du système,  $C$  représente les forces de Coriolis et centrifuge et  $G$  représente les forces de gravité et les forces extérieures. En reformulant cette équation, on obtient l'expression de l'accélération angulaire des segments :

$$\ddot{q}(t) = M(q)^{-1} \{ \tau(t) - C(q, \dot{q}) - G(q) \}$$

Il est alors possible de déterminer l'état du système par intégration :

$$\dot{q}(t) = \int \ddot{q}(t) dt$$

$$q(t) = \int \dot{q}(t) dt$$

L'équation du mouvement étant une équation différentielle, il est nécessaire d'utiliser des méthodes d'intégration adaptées. Différentes méthodes d'intégration existent avec leurs avantages et leurs inconvénients. Une propriété des méthodes d'intégration pouvant orienter le choix de la méthode à utiliser est sa capacité à produire une solution numériquement stable malgré les variations de raideur et d'amortissement qui peuvent intervenir dans les équations. Pour des calculs de multicorps rigides articulés, ces variations sont notamment causées par les contacts entre les pieds et le sol pendant lesquels l'état du système évolue beaucoup plus rapidement que le reste du temps. Ces variations rapides de l'état du système peuvent alors provoquer des instabilités dans la solution si le pas de temps n'est pas assez petit.

Un premier type de méthode est dit explicite. Il a pour principe de déterminer l'état du système au prochain pas de temps en fonction de l'état actuel du système :

$$Y(t + \Delta t) = F(Y(t))$$

avec  $Y(t)$  l'état du système au temps  $t$  et  $\Delta t$  le pas de temps. Cette méthode est plus simple à implémenter et requiert moins de calculs que les méthodes ci-après mais est aussi plus sensible aux variations de raideur et d'amortissement dans les équations.

Un deuxième type de méthode est dit implicite. Dans ce cas, l'état du système au prochain pas de temps est trouvé en résolvant une équation mettant en relation l'état actuel et l'état au prochain pas de temps :

$$G(Y(t), Y(t + \Delta t)) = 0$$

Cette méthode fait appel à des calculs supplémentaires par rapport à la méthode explicite (convergence vers 0 d'un résidu, c'est-à-dire  $\|G\| < \varepsilon$ ) mais est plus résistante aux variations de raideur et d'amortissement dans les équations et peut donc tolérer un pas de temps plus important. Ainsi, selon les cas, il peut être préférable, pour diminuer le temps de calcul, d'utiliser une méthode implicite avec un pas de temps plus élevé qu'une méthode explicite avec un pas de temps faible.

Enfin un troisième type de méthode est dit explicite-implicite ou semi-implicite. Il décompose le problème en deux équations différentielles différentes dont l'une est résolue explicitement et l'autre implicitement.

## B. Les différents types de simulation de la marche

Les deux aspects à contrôler pour simuler une marche stable sont la mise en mouvement du modèle et sa stabilité. Les différents types de simulation ont une approche différente du problème et seront explicitées dans cette partie.

### 1. Marcheurs passifs

Les marcheurs passifs, majoritairement développés dans le domaine de la robotique, ont pour principe de pouvoir marcher en descendant une pente, uniquement grâce à l'action de la gravité sans faire appel à des actionneurs (Figure 6). La projection en avant du membre inférieur en phase oscillante se fait par balancement similaire à celui d'un pendule, et l'énergie perdue lors du contact entre le pied et le sol est compensée par la conversion de l'énergie potentielle en énergie cinétique lors de la descente de la pente. Plus l'angle de la pente est élevé, plus la vitesse de marche augmente. Cependant, un angle trop élevé provoque une instabilité de la marche.

Les avantages de cette méthode de simulation sont qu'elle permet d'utiliser un modèle simple tel que celui du « compass gait » (Kuo, 2001) et de s'affranchir du besoin d'un contrôleur. Malgré cela, elle permet tout de même de réaliser des études sur les principes de la marche tels que la relation entre la vitesse et la longueur de pas (Kuo, 2001), la dépense d'énergie (Donelan et al., 2002), le travail mécanique lors de la transition entre deux pas (Kuo et al., 2005) ou la transition entre marche et course (Srinivasan & Ruina, 2006). Un marcheur passif peut aussi utiliser des modèles plus complexes intégrant des genoux, des chevilles et des segments représentant le tronc et les bras pour étudier leur effet sur la stabilité du mouvement (Borzova & Hurmuzlu, 2004; Collins et al., 2001, 2009).

La simplicité et le principe de cette méthode l'empêchent toutefois d'être étendu à des situations plus diverses et ne permettent pas une analyse plus poussée des mécanismes de la marche et du contrôle de l'équilibre. On peut néanmoins s'inspirer de ces principes pour se permettre de ne pas contrôler certains DoF d'un modèle en remplaçant non pas tous les actionneurs mais certains d'entre eux par des couples passifs.

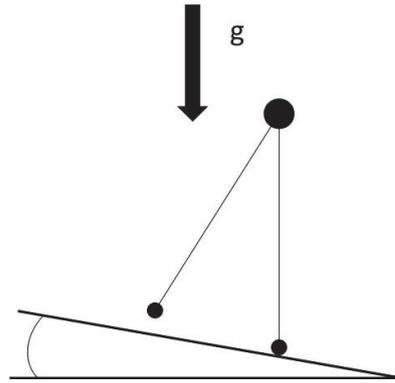


Figure 6 : Schématisation d'un marcheur passif

## 2. Pendule inversé

Les méthodes de simulation utilisant le principe de pendule inversé ne sont pas forcément utilisées conjointement avec une modélisation en pendule inversé. Certaines études utilisent des modèles squelettiques mais l'approximent en pendule inversé lors de la phase de simple appui. En effet, l'avantage de la modélisation en pendule inversé est la simplicité des équations du mouvement qui en découle. Cette modélisation sert généralement à déterminer où poser le pied en fin de phase oscillante pour assurer la stabilité du modèle.

### 2.1. Pendule inversé linéarisé

Une première méthode de contrôle utilisant le pendule inversé a été proposée par Kajita et Tani (Kajita & Tani, 1991) et Kajita et al. (Kajita et al., 1992). Dans celle-ci, le modèle utilisé représente un bipède idéal par un pendule inversé planaire constitué d'un solide disposant d'une masse et d'une inertie dont la hauteur du centre de masse (CoM) est considérée constante, avec des membres inférieurs sans masse et de longueurs variables et marchant selon une succession de phases de simple appui. Cette configuration permet de trouver l'équation du mouvement suivant :

$$\ddot{x}_g = g \frac{x_g}{y_g}$$

avec  $x_g$  et  $y_g$  la position et la hauteur du CoM,  $\ddot{x}_g$  l'accélération du CoM et  $g$  l'accélération de la pesanteur. Les auteurs multiplient alors cette équation par  $\dot{x}_g$  puis l'intègrent pour obtenir :

$$\frac{1}{2} \dot{x}_g^2 - \frac{g}{2 y_h} x_g^2 = \text{Constante} = E$$

Les auteurs interprètent cette équation comme montrant la conservation d'une énergie qu'ils nomment « énergie orbitale » en considérant que le premier terme représente l'énergie cinétique tandis que le second représente l'énergie potentielle. La valeur de cette constante caractérise alors le comportement du CoM. Une valeur nulle indique que le pendule est à l'état d'équilibre (CoM à la verticale du point de contact avec le sol et vitesse nulle), ou dans l'état qui y mène. Une valeur positive indique que le CoM avance et sera capable de dépasser le point de contact avec le sol alors qu'une valeur négative indique qu'il ne le sera pas. Le type de mouvement du bipède dépend donc de cette constante. La valeur de cette constante ne pouvant être modifiée qu'au changement d'appui et donc de manière instantanée, il est possible de contrôler la marche en adaptant la longueur de pas de sorte que  $E$  ait la valeur désirée pour le prochain pas. En notant  $x_{fn}$  et  $v_{ex}$  la position et la vitesse du CoM au moment du changement d'appui et  $s_{n+1}$  la longueur du pas, il est possible d'exprimer  $E_n$  et  $E_{n+1}$ , les valeurs de  $E$  avant et après le changement d'appui :

$$E_n = \frac{1}{2} v_{ex}^2 - \frac{g}{2 y_h} x_{fn}^2$$

$$E_{n+1} = \frac{1}{2} v_{ex}^2 - \frac{g}{2 y_h} (x_{fn} - s_{n+1})^2$$

Il est alors possible de déterminer la longueur de pas à utiliser via l'équation suivante :

$$x_{fn} = \frac{y_h}{g s_{n+1}} (E_{n+1} - E_n) + \frac{s_{n+1}}{2}$$

Ce contrôle a par la suite été amélioré de différentes façons. Tout d'abord, Kajita et al. (Kajita et al., 2001) ont eux-mêmes amélioré leur modèle en le faisant évoluer dans un environnement 3D, en ajoutant des couples au niveau des points de contact avec le sol et en simulant la marche par des successions de phases de simple appui et double appui. Les longueurs et largeurs de pas sont toujours déterminées par la valeur de l'énergie orbitale, mais des couples supplémentaires permettent au CoM simulé de suivre la trajectoire souhaitée du CoM, notamment lors des phases de double appui.

Park et Kim (Park & Kim, 1998) ont cherché à améliorer la stabilité du modèle. En effet, le centre de pression (CoP), défini comme le point d'application des forces de réaction du sol, d'un robot bipède dont le contrôle se base sur un simple pendule inversé a tendance à décrire des mouvements larges, ce qui est un signe d'une faible stabilité. Les auteurs ayant estimé que le problème venait de la mauvaise représentation de la répartition de la masse, ils ont utilisé un modèle de double pendule dont la masse ponctuelle supplémentaire représente celle du membre inférieur en phase oscillante. Cette masse supplémentaire se répercute sur l'équation du mouvement tel que :

$$\ddot{x} = g \frac{x}{y_h} + F(t)$$

où  $F(t)$  représente l'effet du membre inférieur en phase oscillante sur le CoM. En déterminant à l'avance la trajectoire du membre inférieur en phase oscillante, il est donc possible de prendre en compte son effet lors du calcul de la trajectoire du CoM, améliorant ainsi sa stabilité. Albert et Gerth

(Albert & Gerth, 2003) ont par la suite développé cette idée en concevant un modèle dont le membre inférieur en phase oscillante est constitué de 3 masses.

## 2.2. Pendule inversé à hauteur variable

Tsai et al. (Tsai et al., 2010) proposent une méthode où le pendule n'est pas considéré à hauteur constante (non linéarisé). En se basant sur la loi de conservation de l'énergie, le mouvement angulaire du membre inférieur en phase d'appui peut être décrit par :

$$\frac{1}{2} I \omega_s^2 + \int_{\theta_s}^{\theta_e} \tau(\theta) d\theta = \frac{1}{2} I \omega_e^2 \quad (2)$$

Avec

$$\tau(\theta) = mgr \sin(\theta)$$

où  $\theta_s$  et  $\theta_e$  sont les angles initiaux et finaux du pendule,  $\omega_s$  et  $\omega_e$  les vitesses angulaires initiales et finales,  $\tau$  est le couple dû à la gravité et  $r$  la longueur du pendule. En calculant l'intégrale, l'équation devient :

$$\frac{1}{2} I \omega_s^2 - mgr (\cos(\theta_e) - \cos(\theta_s)) = \frac{1}{2} I \left( \frac{d\theta_e}{dt} \right)^2 \quad (3)$$

Elle permet de calculer la trajectoire du CoM. La position de pose du pied est alors déterminée en fonction d'un état futur souhaité du pendule, i.e.  $\theta_{SW2}$  et  $V_{SW2}$  respectivement l'angle du pendule et la vitesse du CoM. En utilisant les équations (2) et (3) il est possible d'écrire :

$$\frac{1}{2} I \omega_{SW1}^2 - mgr \dot{\theta}$$

Il est possible d'exprimer  $\omega_{SW1}$  et  $\omega_{SW2}$  tel que :

$$\omega_{SW1} = \frac{1}{r} V_{SW1} = \frac{1}{r} V_{ST} \cos(\theta_{SW1} + \theta_{ST})$$

$$\omega_{SW2} = \frac{1}{r} V_{SW2}$$

où  $V_{ST}$  et  $V_{SW1}$  sont les vitesses du CoM immédiatement avant et après le contact du membre inférieur en phase oscillante. La seule inconnue restante est alors  $\theta_{SW1}$ , permettant de déterminer où poser le pied de manière optimale.

Une autre méthode se basant sur un pendule à hauteur variable est proposée par Coros et al. (Coros et al., 2010) en mettant en équation l'énergie du système à l'état actuel et l'énergie à un état futur :

$$\frac{1}{2} m v'^2 + mgh = \frac{1}{2} m v'^2 + mgh'$$

où  $v'$  et  $h'$  représentent la vitesse et la hauteur du pendule dans un état futur. Ils résolvent alors l'équation de sorte à ce que l'état futur soit à l'équilibre. Soit  $v' = 0$  et  $h' = L = \sqrt{h^2 + d^2}$  où  $L$  est la longueur du pendule et  $d$  la distance entre le pendule et le point d'appui. Il en résulte :

$$d = v \sqrt{\frac{h}{g} + \frac{v^2}{4g^2}} \quad (4)$$

Cette valeur de  $d$  est celle nécessaire pour que le pendule arrive à l'état d'équilibre après le prochain appui. La position du pas est alors déterminée par :

$$d' = d - \alpha V_d \quad (5)$$

où  $V_d$  est la vitesse moyenne désirée et  $\alpha$  une constante. En diminuant la longueur de pas par rapport à ce qu'il faudrait pour s'arrêter, le pendule peut alors dépasser son point d'appui et donc continuer son mouvement.

### 2.3. Centre de masse extrapolé et marge de stabilité

Une troisième approche d'utilisation du modèle de pendule fait appel au principe de centre de masse extrapolé et de la marge de stabilité. Hof et al. (Hof et al., 2005) partent de l'équation d'Euler pour un pendule simple :

$$\sum M = I\alpha \quad (6)$$

En considérant  $l$  la longueur du pendule,  $I = ml^2$ ,  $x$  la projection verticale du CoM et  $u$  la position du CoM, l'équation (6) devient :

$$(u - x)mg = I\alpha \approx -ml^2 \frac{\ddot{x}}{l}$$

Puis :

$$u - x = \frac{-l}{g} \ddot{x} = \frac{-\ddot{x}}{\omega_0^2}$$

où  $\omega_0 = \sqrt{g/l}$  est la fréquence d'oscillation d'un pendule de longueur  $l$ . Dans le cas idéal où le CoP est fixe et en prenant en compte la vitesse initiale du CoM, sa position est décrite par :

$$x(t) = u + (x_0 - u) \cosh(\omega_0 t) + \frac{v_0}{\omega_0} \sinh(\omega_0 t)$$

La condition pour que le CoM ne dépasse pas le CoP est donc que  $x(t) \leq u$  pour tout  $t$ . Cela se traduit par :

$$x_0 + \frac{v_0}{\omega_0} \leq u$$

Lors de la marche d'un bipède, le CoP peut se déplacer dans la base de support (BoS) définie par la surface sous le pied lors d'une phase de simple appui, et la surface sous et entre les pieds lors d'une phase de double appui. Ainsi, si cette inégalité n'est pas respectée mais que le CoP peut se déplacer suffisamment rapidement de sorte à ce qu'elle le devienne, alors il est possible d'empêcher le CoM de dépasser le CoP. Une condition pour que le CoM ne dépasse pas la BoS à tout instant est proposée telle que :

$$x + \frac{v}{\omega_0} \leq BoS_{max}$$

Le terme de gauche est désigné comme le centre de masse extrapolé (XCoM). De cette notion, une mesure pour quantifier la stabilité de la marche nommée marge de stabilité (MoS) est définie telle que :

$$MoS = BoS_{max} - XCoM$$

ou encore

$$MoS = CoP - XCoM \text{ (Hof et al., 2007)}$$

Une MoS positive indique que le pendule n'arrivera pas à dépasser le point d'appui, et inversement pour une MoS négative. La valeur de la MoS peut alors être étudiée en analyse de la marche. Cependant, Hof et al. (Hof, 2008) ont utilisé le concept de MoS pour la détermination de la position de contact du pied en phase oscillante dans le cas d'un bipède idéal marchant par une succession de phases de simple appui et dont le CoP change de manière instantanée lors du changement d'appui.

Le contrôle dans la direction antéropostérieure est réalisé en plaçant le CoP du nouveau pas à :

$$CoP = XCoM - b_x \quad (7)$$

avec

$$b_x = \frac{s_c}{e^{\omega_0 T_c} - 1}$$

où  $s_c$  est la longueur de pas prescrite et  $T_c$  la durée de pas prescrite. Le CoP étant placé en arrière du XCoM, le CoM est amené à dépasser le CoP et continuer le mouvement en avant. De plus, une perturbation du XCoM se répercute sur la position du CoP, assurant la stabilité du modèle.

Le contrôle médio latéral est similaire :

$$CoP_z = XCoM_z + (-1)^n b_z + K_1 (XCoM_z - x_c)$$

avec

$$b_z = \frac{w_c}{e^{\omega_0 T_c} + 1}$$

où  $w_c$  est la largeur de pas prescrite,  $n$  le numéro du pas et  $x_c$  est le milieu du chemin à suivre. Une perturbation du  $XCoM_z$  se répercute sur le placement du CoP. Cependant la correction apportée est définitive, ce qui implique que la trajectoire du pendule se décale définitivement. Cette correction est amendée par le dernier terme.

Comme explicité précédemment, le contact du pied avec le sol entraîne une perte d'énergie. Tsai et al. (Tsai et al., 2010) explicitent la variation de vitesse du CoM avant et après impact par :

$$V_2 = V_1 \cos \alpha$$

avec  $\alpha$  l'angle entre les jambes du modèle. La perte d'énergie est alors exprimée par :

$$\Delta E = \frac{m}{2} V_1^2 - \frac{m}{2} V_2^2 = \frac{m}{2} (V_1 \sin \alpha)^2$$

Cette perte d'énergie doit être compensée pour assurer au modèle de pouvoir continuer à avancer. L'ajout d'énergie vient alors des efforts articulaires dont l'action engendre des forces de réaction du sol sur le modèle pour le faire se mouvoir. Par exemple, dans le cas du modèle de Kajita et al. (Kajita & Tani, 1991) dont le contrôle repose uniquement sur le principe de pendule inversé, ce rôle est joué par l'articulation prismatique à chaque jambe qui est capable d'exercer une force dont le contrôle est simplement basé sur la conservation de la hauteur du CoM. Cependant, dans un cas comme celui de Tsai et al. (Tsai et al., 2010) qui utilise un modèle squelettique, les efforts articulaires qui jouent ce rôle ne peuvent pas être uniquement contrôlés par le principe du pendule inversé. Dans ce cas, la modélisation en pendule inversé fait en réalité partie d'un contrôle plus global et est généralement utilisé pour prescrire la position de pose du pied en fin de phase aérienne.

### 3. Contrôle du mouvement par la cinématique articulaire

Le principe de cette méthode de contrôle est de déterminer et de suivre un état cinématique, généralement les angles et les vitesses angulaires pour chaque articulation. Ces états cinématiques sont déterminés par le contrôleur en fonction de l'état actuel du système ainsi que potentiellement par des poses de référence. Chaque articulation est alors contrôlée par un « feedback controller » local produisant un couple articulaire visant à minimiser la différence entre l'état actuel du système et l'état

cible. De manière générale, des trajectoires initiales permettant de donner un mouvement global cohérent avec le résultat final attendu sont déterminées hors ligne avant le début de la simulation. Une première brique de contrôle assure le suivi de ces trajectoires tandis qu'en parallèle, d'autres briques modifient ces trajectoires pour assurer le contrôle, entre autres, de l'équilibre et de la vitesse du modèle. L'avantage de cette méthode est qu'en se basant uniquement sur la cinématique, elle ne fait pas directement appel aux équations du mouvement. En contrepartie, les articulations étant contrôlées de manière individuelle, le mouvement résultant n'est pas issu d'une coordination des articulations décidée spécifiquement par le contrôleur (Figure7).

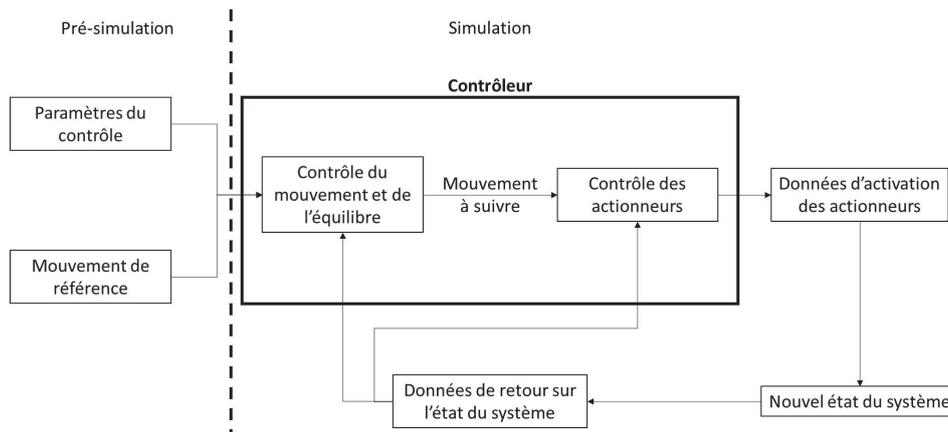


Figure 7 : Représentation schématique du contrôle de mouvement par la cinématique articulaire

Les briques de contrôle modifiant ou créant des trajectoires pendant la simulation font généralement appel au principe de cinématique inverse. La cinématique inverse consiste à prescrire à un système articulé des angles articulaires permettant au point en fin de chaîne d'être à une position souhaitée. Dans le cas du contrôle du membre inférieur en phase oscillante, le point en fin de chaîne est un point appartenant au pied. Sa position cible évoluant à chaque pas de temps, le calcul de cinématique inverse est utilisé pour déterminer les variations d'angle  $\Delta\theta$  à prescrire pour permettre au pied de parcourir la variation de position prescrite  $\Delta p$ . La méthode la plus couramment utilisée pour les calculs de cinématique inverse en robotique et en infographie est celle de la jacobienne inverse :

$$\Delta(\theta) = J(p)^{-1} \Delta p \quad (8)$$

où  $J(p)^{-1}$  est l'inverse de la jacobienne :

$$J(p)_{i,j} = \frac{\delta p_i}{\delta \theta_j}$$

Un potentiel problème à cette méthode est que selon le système, la jacobienne n'est pas carrée et ne possède donc pas d'inverse. Une première approximation pour résoudre ce problème est l'utilisation de la pseudo-inverse de la jacobienne. Cependant, une seconde approximation consiste à utiliser la transposée de la jacobienne. Cette approximation, bien que moins précise que celle de la pseudo inverse, demande également moins de ressource de calcul pour sa détermination et est donc généralement préférée lorsque l'un des objectifs est le contrôle en temps réel.

### 3.1. Détermination des états cibles

#### 3.1.1. Génération procédurale

Une première méthode pour générer des états cibles (ou poses de référence) consiste à générer les trajectoires de manière procédurale. Elles sont alors le résultat de fonctions ou d'algorithmes créés en se basant sur l'intuition ou des connaissances issues de la robotique ou de la biomécanique. Le contrôle est ensuite basé sur une machine à états finis qui traque les trajectoires en fonction de la phase de marche dans laquelle se trouve le modèle. Cette méthode a pu être utilisée pour simuler la marche (Raibert & Hodgins, 1991), la course (Hodgins et al., 1995), ou d'autres mouvements tels que le saut, s'asseoir, se relever ou même tomber (Faloutsos et al., 2001a, 2001b; Wooten & Hodgins, 2000). Deux limitations sont à noter concernant cette méthode. La première est le long processus d'essais et erreurs pour arriver à simuler un mouvement correct et, généralement, la sensibilité à une modification du modèle. La seconde est due à l'utilisation de fonctions pour générer les trajectoires articulaires, résultant en un mouvement paraissant peu naturel.

#### 3.1.2. Poses contrôlées par l'animateur

Cette méthode est similaire à celle présentée précédemment à la différence que la machine à états finis est composée de poses clefs déterminées par l'animateur au lieu de fonctions. L'avantage est que les poses clefs servant de cibles peuvent être déterminées de manière intuitive, facilitant la création d'un mouvement. De même, la modification d'un mouvement peut se faire par simple modification des poses clefs. Cette méthode a pu être utilisée pour simuler la marche en 2D (van de Panne et al., 1994), en 3D (Laszlo et al., 1996) ou d'autres tâches telles que la montée d'escaliers (Laszlo et al., 1996), le ski (Zhao & van de Panne, 2005) ou la natation (Yang et al., 2004). Coros et al. (Coros et al., 2010) ont également amélioré la qualité du suivi des poses clefs en interpolant les trajectoires par des splines, permettant de traquer des trajectoires continues.

#### 3.1.3. Trajectoires issues de la capture de mouvement

Cette méthode consiste à directement utiliser des trajectoires issues de la capture de mouvement comme états cibles. L'avantage de l'utilisation de données issues de l'expérimentation est le fait que les trajectoires soient assurées d'être réalisables. De plus, le mouvement pourra avoir l'air plus naturel qu'un mouvement issu d'une génération procédurale ou de poses créées par un animateur. Cependant, l'utilisation de données expérimentales présente tout de même certains défauts. Tout d'abord, comme explicité précédemment, un modèle est une approximation de la réalité. Un mouvement acquis expérimentalement ne pourra donc pas parfaitement convenir à un modèle particulier. De même, les forces de réaction du sol calculées étant elles-mêmes issues d'une modélisation, leur impact sur le bipède ne sera pas le même qu'avec les forces réelles durant l'expérimentation. Enfin, la précision des outils de capture ne permet pas de mesurer l'ensemble des mécanismes de contrôle de l'équilibre à l'œuvre durant l'expérimentation. Se contenter de seulement suivre ces trajectoires aurait donc pour résultat l'accumulation d'erreurs menant à la perte d'équilibre du modèle. Cette méthode a donc elle aussi besoin de briques de contrôles supplémentaires pour corriger les erreurs et instabilités lors de la simulation. Pour cela il est possible de directement modifier les trajectoires articulaires (Sok et al., 2007) ou d'ajouter un couple articulaire provenant d'une brique de contrôle différente et dédiée à la conservation de l'équilibre (Zordan, 2002).

#### 3.1.4. Zero moment point (ZMP)

Le ZMP est défini comme le point où les moments tangentiels résultants des forces actives sur le pied au sol sont nuls (Vukobratović & Borovac, 2004). La désignation des forces actives inclue les forces externes (du reste du corps sur le pied), la gravité et la force de Coriolis en excluant l'inertie du pied et les forces de réaction du sol. Le principe du ZMP est qu'à l'état d'équilibre, quand le pied est statique sur le sol, le ZMP est situé sous la base de support et coïncide avec le CoP (Wieber, 2002). Au contraire, si le ZMP sort de la base de support, cela signifie que le pied se met à rouler sur le sol.

L'utilisation du ZMP lors d'une simulation se fait en plusieurs étapes. Tout d'abord, la trajectoire souhaitée du ZMP est planifiée à l'avance. La position du pied est ensuite déterminée de sorte à pouvoir respecter la condition de stabilité du modèle i.e. le ZMP doit rester dans la BoS définie par le pied au sol. La trajectoire du pelvis est alors calculée de manière à pouvoir correctement traquer le ZMP. Enfin, les trajectoires des angles articulaires sont déterminées par cinématique inverse.

Cette méthode de contrôle est notamment utilisée dans le domaine de la robotique (Al-Shuka et al., 2016), mais possède des limitations dans le cadre d'une marche davantage biofidèle. Pour convenir au principe de stabilité du ZMP, les pieds au sol doivent être posés à plat sur le sol ce qui impose des trajectoires non naturelles au bipède.

### 3.2. Contrôle de l'équilibre et de la vitesse

### 3.2.1. Contrôle de la pose du pied

De manière générale, l'équilibre et la vitesse du modèle sont contrôlés par la position du pied en fin de phase oscillante. Plus le pied est posé loin en avant, plus la vitesse du modèle diminuera lors du prochain pas et inversement. De même, plus le pied est posé latéralement, plus le modèle se décalera à l'opposé et inversement. Ajuster la position et le temps d'arrivée du pied est donc un bon moyen pour corriger les erreurs de vitesse, d'équilibre, ainsi que celles dues à des perturbations. La première étape d'une brique de contrôle gérant le membre inférieur en phase oscillante est de déterminer le point de contact souhaité. La seconde étape est alors de contrôler le membre inférieur de sorte à atteindre l'objectif défini via un calcul de cinématique inverse. La méthode généralement utilisée pour déterminer le point d'arrivée du pied consiste à utiliser le modèle de pendule inversé vu précédemment. Tsai et al. (Tsai et al., 2010) utilisent par exemple des données issues de captures de mouvement pour déterminer la trajectoire à suivre par le pied, tandis que Coros et al. (Coros et al., 2010) utilisent une interpolation linéaire entre le point de départ du pied et son point d'arrivée.

### 3.2.2. Forces virtuelles

En animation non basée sur la physique, il est possible d'assurer l'équilibre du modèle en appliquant des forces arbitraires et sans réalité physique sur celui-ci. Lors d'une simulation basée sur la physique ou durant le contrôle d'un robot, l'utilisation d'une telle méthode est impossible. Le principe des forces virtuelles consiste à simuler l'application d'une force en un point grâce à l'action d'une ou de plusieurs articulations (Figure 8). Dans ce cas, l'adaptation ne se fait pas par modification des trajectoires articulaires mais par ajout d'un couple articulaire. Coros et al. (Coros et al., 2010) utilisent ce principe pour appliquer une force au CoM en utilisant les articulations du membre inférieur en phase d'appui et du haut du corps en définissant les forces virtuelles telles que :

$$\vec{\tau}_F = J(p)^T \vec{F} \quad (9)$$

où  $\vec{\tau}_F$  est le vecteur de couple articulaire à appliquer,  $\vec{F}$  est la force que l'on souhaite appliquer et  $J(p)^T$  est la transposée de la jacobienne :

$$J(p)_{i,j} = \frac{\delta p_i}{\delta \theta_j}$$

La force  $F_v$  qu'ils appliquent au CoM est calculée telle que :

$$F_v = k_v (V_d - V)$$

où  $V$  est la vitesse du modèle,  $V_d$  est la vitesse cible et  $k_v$  une constante. Cela permet alors de contrôler plus finement la vitesse du modèle autrement qu'uniquement par la position de pose du pied.

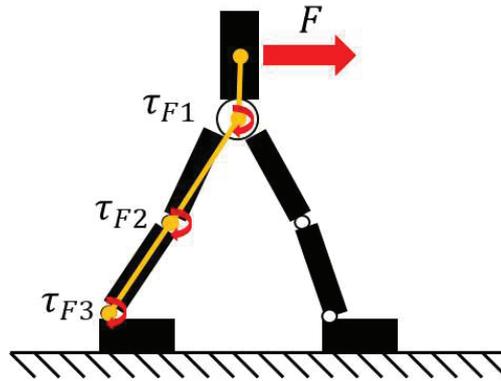


Figure 8 : Utilisation des couples articulaires pour simuler une force (Coros et al., 2010)

On remarque que la forme de l'équation (9) ressemble à celle de la cinématique inverse (8). Idéalement, il faudrait utiliser dans un cas comme dans l'autre l'inverse de la jacobienne. Cependant, la matrice jacobienne d'un bipède n'étant généralement pas carrée, elle est non inversible et la transposée est utilisée comme approximation de la matrice pseudo-inverse.

### 3.3. Suivi des trajectoires articulaires

Une fois les trajectoires spécifiées, le contrôleur doit déterminer les couples articulaires à appliquer aux articulations pour minimiser l'écart entre l'état du modèle et l'état désiré.

#### 3.3.1. Contrôleur proportionnel dérivé

La méthode la plus couramment utilisée est le contrôleur proportionnel dérivé (PDC) qui calcule le couple en fonction de la différence entre l'état cible et l'état actuel :

$$\tau = k_p(\theta_d - \theta) + k_v(\dot{\theta}_d - \dot{\theta}) \quad (10)$$

où  $\theta$  et  $\dot{\theta}$  sont respectivement les positions et vitesses angulaires des articulations et  $\theta_d$  et  $\dot{\theta}_d$  sont les positions et vitesses angulaires cibles. Ainsi, plus la différence entre l'état cible et actuel est grande, plus le couple calculé est élevé. La prise en compte du terme de vitesse est indispensable pour éviter au système d'être non amorti et d'osciller autour de la position cible, voire d'être instable. Cependant il est aussi possible d'utiliser le PDC avec  $\dot{\theta}_d = 0$ . Dans ce cas le contrôle est équivalent à un système de ressort amortisseur. Le rapport entre la différence d'état et le couple produit est alors déterminé par les valeurs de gains  $k_p$  et  $k_v$  dont le réglage est un élément clef pour le bon fonctionnement du contrôleur. En effet, des gains trop faibles ne permettront pas au modèle de suivre les trajectoires de référence. Au contraire, des gains trop élevés peuvent entraîner des instabilités numériques et répondre trop brutalement à des perturbations, diminuant ainsi la compliance du système. Pour obtenir un mouvement le plus stable possible, il convient donc de déterminer les gains les plus faibles possibles mais permettant tout de même de suivre les trajectoires souhaitées (Carensac et al., 2018). Cette tâche peut se faire

manuellement par un processus d'essai erreur ou par optimisation. Les gains en position et en vitesse doivent également être réglés ensemble. En effet, dans le cas où le gain en vitesse serait trop faible par rapport au gain en position, la position angulaire oscillerait autour de la valeur cible, similairement au cas où la composante de vitesse serait négligée, avant de tendre vers la valeur cible. Au contraire un gain en vitesse trop élevé provoquerait une convergence trop lente vers la position angulaire de référence. Théoriquement, la relation optimale entre les deux gains permettant la convergence la plus rapide vers la position de référence sans oscillation est :

$$k_v = 2\sqrt{k_p}$$

Dans le cas d'une simulation de la marche, cette relation n'est pas exacte à cause des changements constant d'états cibles et de la dynamique du système. Cependant, elle reste largement utilisée telle quelle.

### 3.3.2. Contrôleur proportionnel intégral dérivé

Un contrôleur proche du PDC est le contrôleur proportionnel intégral dérivé (PIDC) qui utilise un terme intégrant l'erreur entre la position actuelle et celle de référence :

$$\tau = k_p(\theta_d - \theta) + k_i \int (\theta_d - \theta) dt + k_v(\dot{\theta}_d - \dot{\theta})$$

où  $k_i$  est le gain d'intégrale. Grâce à ce terme en plus, si la position contrôlée tend vers une valeur différente de celle de référence, par exemple à cause de l'action de la gravité, l'accumulation de l'erreur permet d'augmenter la valeur du couple calculé et donc de se rapprocher de la valeur cible. Cependant, cette application n'est pas utilisée en simulation de la marche car le changement constant d'état cible rend ce terme inutile. Si le PIDC est généralement beaucoup plus utilisé que le PDC dans l'industrie pour l'asservissement de systèmes, il n'est en revanche pas utilisé dans la simulation de la marche.

### 3.3.3. Contrôleur proportionnel dérivé stable

Tan et al. (Jie Tan et al., 2011) ont proposé de modifier le PDC pour prendre en compte l'état du système au prochain pas de temps :

$$\tau^n = k_p(\theta_d^{n+1} - \theta^{n+1}) - k_v \dot{\theta}^{n+1}$$

où l'indice  $n$  et  $n+1$  indiquent le pas de temps actuel et le suivant. Pour déterminer l'état au prochain pas de temps ils utilisent une expansion en série de Taylor en ne gardant que les deux premiers termes :

$$\theta^{n+1} = \theta^n + \Delta t \dot{\theta}^n$$

$$\dot{\theta}^{n+1} = \dot{\theta}^n + \Delta t \ddot{\theta}^n$$

où  $\Delta t$  est le pas de temps. Le contrôleur est alors défini comme :

$$\tau = k_p (\theta_d^{n+1} - \theta^n - \Delta t \dot{\theta}^n) - k_v (\dot{\theta}^n + \Delta t \ddot{\theta}^n)$$

L'idée derrière cette formulation est de découpler la relation entre gain élevé et pas de temps. En effet, si l'on souhaite traquer une trajectoire de manière très précise, il faut utiliser des gains élevés. Mais comme vu précédemment, cela peut entraîner des instabilités numériques. Pour les éviter, il est possible de diminuer le pas de temps de la simulation mais cela se fait au prix du temps de calcul qui devient plus élevé. Cette formulation permet donc l'utilisation de gains élevés sans avoir à drastiquement diminuer le pas de temps.

### 3.3.4. Contrôle antagoniste

Les muscles contrôlant les articulations ne peuvent déployer de force que dans une seule direction. Pour pouvoir complètement contrôler une articulation, il faut donc au minimum 2 muscles antagonistes. Neff et al. (Neff & Fiume, 2002) s'inspirent de cela pour proposer un contrôle basé sur des ressorts linéaires antagonistes. Chaque articulation est contrôlée par deux ressorts dont la position de référence est fixe et correspond aux limites angulaires de l'articulation :

$$\tau = k_L (\theta_L - \theta) + k_H (\theta_H - \theta) - k_d \dot{\theta}$$

où  $\theta_L$  et  $\theta_H$  sont les positions de référence des ressorts. La position angulaire cible n'étant pas indiquée, c'est la variation des gains  $k_L$  et  $k_H$  qui, en faisant varier le point d'équilibre, permet de contrôler la position angulaire de l'articulation.

### 3.3.5. Champ de force non linéaire

Cette méthode est basée sur les travaux de Mussa-Ivaldi (Mussa-Ivaldi, 1997) qui a montré que des champs de force peuvent être utilisés pour générer des mouvements et qu'en superposant plusieurs champs, le bout d'une chaîne articulaire pouvait effectuer différents mouvements. Matarić et al. (Matarić et al., 1999) ont utilisé des superpositions de champs de force à un seul point d'équilibre pour déplacer un modèle où chaque champ de force est décrit par une relation non linéaire entre le couple et l'angle articulaire :

$$\tau = k_d (\theta_d - \theta) e^{\frac{k(\theta_d - \theta)^2}{2}} - k_v \dot{\theta}$$

où  $\tau$  est la dérivée d'un potentiel Gaussien centré sur  $\theta_d$ . Pour une faible erreur entre l'état cible et l'état actuel, le résultat est équivalent au PDC linéaire. Cependant, pour une erreur importante le couple articulaire calculé finira par chuter contrairement à celui calculé par un PDC, ce qui rend le contrôle moins intéressant si les adaptations tendent à trop éloigner le modèle de ses trajectoires de référence.

## 4. Contrôle optimal

Le principe du contrôle optimal est de déterminer un ensemble de valeurs d'actionneurs de sorte à réaliser la totalité d'un mouvement régi par un ensemble de contraintes, tout en minimisant une fonction objectif. Le problème peut être défini par :

$$\text{Trouver : } x \text{ Pour : minimiser } f(x) \text{ Contraint par : } h_i = 0; i = 1, \dots, m \quad (11) \quad g_j \leq 0, j = 1, \dots, k \quad (12)$$

### 4.1. Contraintes et objectifs

#### 4.1.1. Contraintes

Les contraintes représentent les propriétés dynamiques du modèle et les interactions avec l'environnement dans lequel il évolue. Le premier ensemble de contraintes représente la loi de la dynamique de Newton-Euler reliant les couples articulaires et les accélérations angulaires Eq. (11). Etant donnée la forme des équations du mouvement, ces contraintes sont appelées des contraintes d'égalité. Cet ensemble de contraintes est l'un des avantages de la méthode de contrôle optimal par rapport au contrôle cinématique puisqu'elle permet de prendre en compte les équations du mouvement dans la détermination des couples articulaires ; assurant une meilleure prédiction de l'effet de l'application d'un couple ou d'une force. Le second ensemble de contraintes, les contraintes d'inégalités, peuvent représenter les forces de frictions Eq. (12), les limites articulaires ou les couples maximaux.

#### 4.1.2. Objectifs

Le comportement du contrôleur est généralement défini par un ou plusieurs objectifs réunis dans une fonction que l'on cherche à minimiser. Ces objectifs ont généralement pour origine une mesure de performances biomécaniques. Des fonctions objectifs communes sont par exemple

L'effort dynamique :

$$f = \int_0^T \tau^2 dt$$

L'énergie mécanique :

$$f = \int_0^T \sum_i |\tau_i \dot{q}_i| dt$$

L'énergie métabolique dépendant de paramètres liés au modèle, et notamment aux caractéristiques des muscles, de leur activation et de leur travail :

$$f = \int_0^T \dot{E} dt$$

Les à-coups :

$$f = \int_0^T \dot{i}^2 dt$$

La stabilité :

$$f = \int_0^T S dt$$

où la définition de  $S$  peut être multiple comme le suivi du CoM ou du ZMP. Le comportement du bipède dépendant des objectifs utilisés et de leur poids dans la fonction à minimiser, le choix de ceux-ci est fondamental. Ackermann et van den Bogert (Ackermann & van den Bogert, 2010) ont par exemple démontré que des objectifs basés sur la mesure de la fatigue résultait en des comportements davantage biofidèles que des objectifs basés sur la consommation énergétique.

## 4.2. Stratégies de contrôle

### 4.2.1. Dynamique inverse

En dynamique inverse, le mouvement est connu à l'avance et les équations du mouvement sont utilisées pour déterminer les couples articulaires nécessaires pour le provoquer. Dans le domaine de la biomécanique, l'optimisation en dynamique inverse est alors notamment utile pour déterminer les forces musculaires à l'origine de ces couples articulaires. En effet, la redondance du système musculaire rend impossible la détermination de l'effet d'un muscle en particulier sur le mouvement. Utiliser une fonction objectif minimisant le coût énergétique ou la fatigue est alors un moyen de trouver une solution unique d'activation musculaire pour un mouvement. Cette méthode d'optimisation peut être à la fois utilisée en analyse du mouvement (Bessonnet et al., 2005; Koopman et al., 1995) et pour générer des trajectoires permettant de respecter les objectifs de la fonction à minimiser (Channon et al., 1992; Chevallereau & Aoustin, 2001; Kim et al., 2008; Ren et al., 2007; Saidouni & Bessonnet, 2003).

### 4.2.2. Dynamique directe

En dynamique directe, les couples articulaires ou les forces musculaires prescrites par le contrôleur sont utilisés pour intégrer les équations du mouvement. Le mouvement résultant dépend donc fortement de la fonction objectif (F. C. Anderson & Pandy, 2001; Bessonnet et al., 2002). Bien que des données cinématiques ne soient théoriquement pas nécessaires, elles peuvent être utilisées dans la fonction objectif comme erreur de suivi (Guess et al., 2014; Miller, 2014; Neptune et al., 2000; Thelen & Anderson, 2006). L'avantage de la dynamique directe est que les mouvements sont directement issus des couples articulaires ou des forces musculaires appliqués au modèle. En revanche, cela nécessite d'intégrer les équations du mouvement et de faire intervenir un modèle de contact du pied sur le sol, ce qui demande plus de ressources de calcul.

Pour l'optimisation en dynamique directe et inverse, le mouvement est optimisé sur l'ensemble du temps de la simulation. Cela est propice à la simulation de la marche ayant pour objectif l'analyse du mouvement et est donc majoritairement utilisé dans le domaine de la biomécanique. Au contraire, les domaines de l'infographie et de la robotique ont pour contrainte de réaliser un contrôle en temps réel et ne peuvent donc pas utiliser ces méthodes telles quelles. Pour contourner ces limitations, des méthodes basées sur l'optimisation mais adaptées pour être réalisées en temps réel telles que l'optimisation immédiate, l'optimisation hors ligne et l'optimisation à court terme ont été développées (Figure 9).

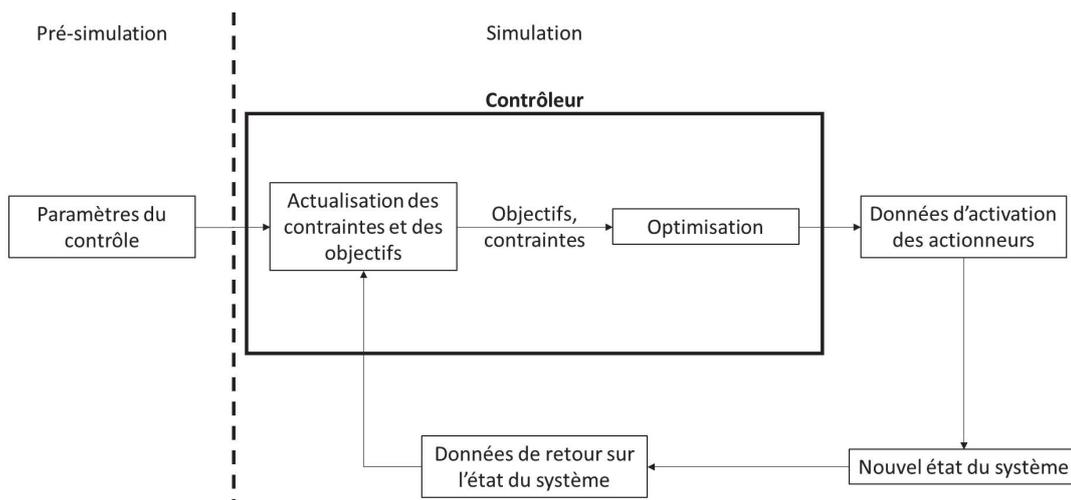


Figure 9 : Représentation schématique du contrôle de mouvement par optimisation en temps réel

#### 4.2.3. Optimisation immédiate

Cette méthode est utilisée pour des mouvements simples, en partie prédéfinis et suivis par le contrôleur. Le contrôleur n'a donc pas besoin d'optimiser le mouvement dans son entièreté. Un premier exemple est le contrôle d'un bipède en position statique. Abe (Abe, 2007) propose un contrôle utilisant simultanément trois briques de contrôle. Une première pour suivre une pose de référence, une deuxième pour tendre la main vers un point, et une troisième pour contrôler la position du CoM. Ce contrôle étant simple, l'optimisation peut alors à chaque pas de temps gérer les priorités à donner aux

briques sans se soucier du mouvement dans son ensemble. Un deuxième exemple est le contrôle de la marche via le suivi d'un mouvement de référence combiné à un contrôle de l'équilibre utilisant une optimisation. Macchietto et al. (Macchietto et al., 2009) utilisent le moment cinétique du modèle pour contrôler son équilibre pendant qu'il suit un mouvement prédéfini. L'optimisation a ici pour rôle de concilier les accélérations angulaires prescrites par les deux briques de contrôle pour n'en fournir qu'une seule et n'a donc là non plus besoin de s'intéresser au mouvement entier. Similairement, il est possible de combiner les briques de contrôle par optimisation avec des machines à états finis à la place du suivi de mouvement de référence (de Lasa et al., 2010; Jain et al., 2009).

#### 4.2.4. Optimisation hors-ligne

Le principe de cette méthode est de déterminer les trajectoires via une optimisation avant la simulation. Ces trajectoires sont alors suivies par un contrôleur n'ayant plus besoin que de gérer des perturbations mineures telles que les erreurs numériques. Les trajectoires optimisées peuvent être directement les angles et les vitesses articulaires (da Silva et al., 2008; Muico et al., 2009) ou les positions et les vitesses de la fin d'une chaîne cinématique (Wu & Popović, 2010). L'une des limitations de l'optimisation est qu'elle n'est pas robuste à des perturbations imprévues et qu'il faudrait alors relancer complètement le processus d'optimisation. Une manière de contourner cette limitation est de déterminer plusieurs trajectoires et d'en suivre une ou d'en mixer plusieurs en fonction de l'état du système (Muico et al., 2011).

#### 4.2.5. Optimisation à court terme

Une autre méthode pour utiliser l'optimisation en temps réel consiste à ne considérer qu'un intervalle de temps relativement court à partir du temps actuel. L'optimisation utilise alors l'état actuel du système comme état initial et détermine les trajectoires optimales sur cet intervalle de temps. L'optimisation est ensuite continuellement reconduite sur un nouvel intervalle (Da Silva et al., 2008; García et al., 1989; Qin & Badgwell, 2003). L'avantage de cette méthode est que même pour un intervalle de temps réduit, elle permet tout de même de prendre en compte l'évolution future du système, ce qui est notamment important pour gérer les discontinuités telles que les contacts du pied avec le sol.

## 5. Contrôle par réseau de neurones

Le principe de ce type de contrôle est d'utiliser un réseau de neurones pour déterminer comment activer les actionneurs en fonction de l'état du système. Chaque neurone reçoit plusieurs données d'entrée et renvoie une donnée de sortie. Ils sont organisés en plusieurs couches successives pouvant également être connectées de manière à former des boucles.

La méthode la plus courante consiste à déterminer la valeur de sortie des neurones en fonction des valeurs d'entrée, du poids qui leur est associé et d'une fonction de seuil (Bishop, 1994). Elle a été utilisée pour des simulations 2D (van de Panne, 1993) ou 3D mais sans donnée d'entrée liée à l'état du système (Reil & Husbands, 2002) ou sans produire de marche stable (Allen & Faloutsos, 2009). Il est également possible d'utiliser des fonctions utilisant des opérations logiques (et, ou, etc...), des décisions (si, alors, etc...) et ayant la capacité de garder des données en mémoire (de Garis, 1990).

La détermination des paramètres des neurones se fait par optimisation à l'aide d'une fonction objectif.

L'avantage de cette méthode est qu'elle ne demande pas de modéliser explicitement les relations entre l'état du modèle et les actionneurs. Cependant, concevoir le réseau requière une phase importante d'essai erreur. De plus, le réseau de neurone peut être assimilé à une boîte noire, et il est donc difficile de comprendre les mécanismes utilisés par le contrôleur (Figure 10).

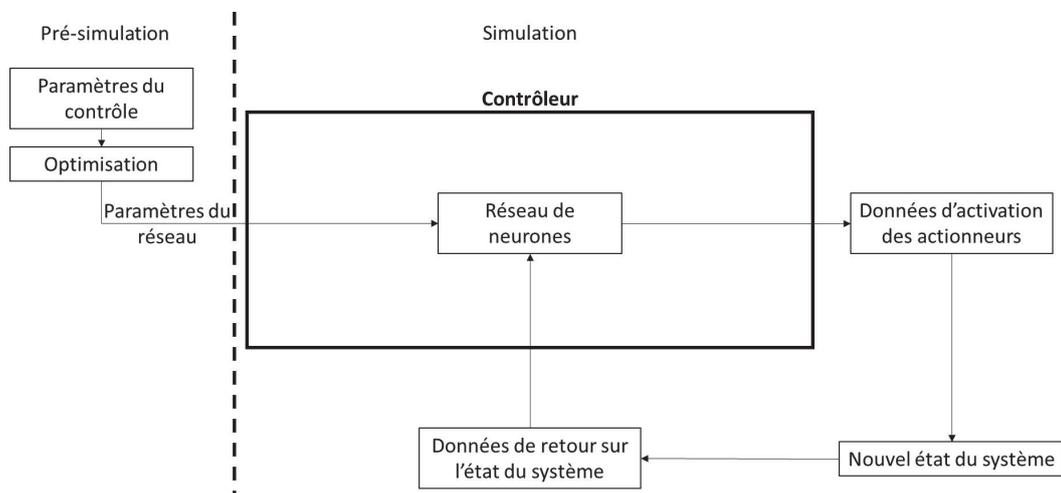


Figure 10 : Représentation schématique du contrôle de mouvement par réseau de neurones

## C. Résumé

Les modèles de bipèdes et contrôles des actionneurs utilisées pour réaliser la marche sont divers et leur utilisation dépend de l'objectif et des contraintes du projet.

Le modèle le plus simple en simulation numérique est le modèle de pendule inversé dont le pendant dans le domaine numérique est le modèle du compass gait. Ces modèles sont des simplifications importantes du corps et ne permettent que des contrôles limités tel que par le placement du pied ou la descente de pente. Des types de modèles plus fidèles sont les modèles squelettiques dans lesquels chaque segments et DoFs représentent un membre et une articulation du corps. Cela permet à la fois de réaliser des mouvements plus complexes et plus fidèles à ce que peut faire un être humain. Les modèles squelettiques sont contrôlés par des couples articulaires. Cela permet de simplifier le

contrôle par rapport à l'utilisation de lignes d'actions car un DoF n'est alors contrôlé que par un couple articulaire. Une variante du modèle squelettique est le modèle musculosquelettique dans lequel les DoFs sont contrôlés par des lignes d'action plutôt que par des couples. L'avantage d'utiliser des lignes d'actions est qu'elles permettent de modéliser des muscles ou groupes de muscles individuels et donc de mieux retranscrire la réalité du corps. En revanche, la redondance de lignes d'actions à chaque DoFs entraîne un nombre illimité de possibilité d'activation des lignes d'actions. Le contrôleur choisit pour ce type de modèle doit donc être adapté.

Les contrôles par marcheurs passifs et pendules inversés, si utilisés tels quels, sont généralement utilisés avec des modèles de pendules inversés ou de compass gait. Ils reposent également sur des simplifications importantes ou des situations restreintes. Le choix entre les autres types de contrôle se fait alors également en fonction des objectifs et contraintes du projet, généralement lié au domaine d'étude. Dans le domaine de la biomécanique, la priorité est généralement donnée à la biofidélité et les modèles les plus utilisées sont les modèles musculosquelettiques. La méthode de contrôle la plus répandue dans ce domaine est donc le contrôle optimal qui permet de déterminer les activations des lignes d'actions malgré leur redondance grâce à l'optimisation. Cependant, les optimisations sont généralement réalisées sur l'intégralité du mouvement souhaité ce qui impose un temps de calcul important. De plus, les simulations par contrôle optimal utilisent généralement un terme de suivi dans leur fonction objectif pour répliquer un mouvement existant, ce qui est idéale pour comprendre les mécanismes d'un mouvement particulier, mais pas pour adapter la cinématique du mouvement. Le contrôle par la cinématique articulaire est plus largement utilisé dans les domaines de l'infographie et de la robotique. Il est généralement utilisé avec des modèles squelettiques contrôlés par des couples articulaires. En effet, ce type de modèle correspond à la réalité d'un robot, et le fait d'utiliser un contrôle demandant moins de ressources de calcul permet d'atteindre plus facilement un contrôle en temps réel. Ce type de contrôle consiste à suivre une trajectoire initiale permettant de générer un mouvement proche de celui attendu, puis de le modifier via diverses boucles de contrôles agissant en temps réel sur la cinématique ou la dynamique du modèle en fonction de son état. Le contrôle par réseau de neurone est surtout développé dans le domaine de l'informatique. L'avantage de cette méthode est qu'elle ne demande pas de lier explicitement l'état du système au contrôle des actionneurs mais la conception et l'optimisation du réseau sont des tâches ardues et il est donc compliqué de l'adapter par la suite. Enfin, bien que les modèles et méthodes de contrôles sont généralement liés au domaine d'étude, cela n'est pas une règle absolue et il est possible de concevoir des contrôles s'inspirant de diverses méthodes tels que les méthodes de contrôle optimal en temps réel qui utilisent un mouvement de référence.

# III. Modèle et contrôleur

Comme présenté dans l'introduction, le but de cette thèse est de concevoir un simulateur de marche qui doit être, à terme, capable de simuler des pathologies lors de passages d'obstacles de la vie courante. Le simulateur a été développé en utilisant la bibliothèque de fonction d'OpenSim (Delp et al., 2007). OpenSim est un logiciel open source permettant de modéliser des systèmes biomécaniques, de réaliser des analyses musculosquelettiques (cinématique et dynamique inverse, optimisation dite « statique » pour estimer les forces musculo-tendineuses). Ce choix s'explique par l'utilisation répandue du logiciel dans la communauté biomécanique, incluant déjà une modélisation du corps humain et des solutions de calcul dynamiques (intégration des équations du mouvement, gestion du contact) rendant le simulateur facilement accessible à celle-ci. Bien que majoritairement utilisé pour l'analyse du mouvement par optimisation en dynamique inverse ou directe via son interface utilisateur, OpenSim propose également des bibliothèques de classes et de fonctions utilisables par des développeurs. Celles-ci peuvent alors être utilisées comme base pour créer un nouveau contrôleur. Ces bibliothèques comportent notamment une classe faisant office de solveur en intégrant les équations du mouvement sur la durée de la simulation et en proposant différentes méthodes d'intégrations. Une autre classe mise à disposition est une classe de contrôle qui après avoir déterminé la valeur des actionneurs à utiliser, les transmet automatiquement au solveur. Un autre avantage à l'utilisation d'OpenSim est l'existence de modèles squelettiques définis par les concepteurs du logiciel et par la communauté, et qu'il est possible de modifier pour correspondre aux caractéristiques que l'on souhaite attribuer au modèle.

Le simulateur devant à terme simuler des marches possiblement pathologiques en passant des obstacles, le choix du type de contrôle doit donc se faire en conséquence. Les modélisations et contrôleurs par marcheur passif ou par pendule inversé pur font appel à trop d'hypothèses simplificatrices pour atteindre la biofidélité escomptée et sont donc considérées inutilisables. Les types de pathologies et les obstacles à simuler étant divers, il faut que le contrôle soit robuste à ces modifications. La littérature présente les méthodes de contrôle à réseau de neurones suffisamment flexibles comme difficilement optimisables, ce qui pourrait compliquer le passage d'une marche simple à des marches pathologiques et en passage d'obstacle. De plus, le réseau de neurones étant assimilable à une boîte noire, l'implémentation des pathologies ou des nouveaux mouvements pourrait s'avérer peu intuitive par rapport aux autres méthodes de contrôle. Cette méthode n'a donc pas été retenue non plus pour ce projet. Dans le domaine de la biomécanique, le choix se porte généralement sur l'optimisation soit pour pouvoir étudier le mouvement au niveau musculaire, soit pour pouvoir suivre une fonction objectif précise. Cependant, les objectifs de ce projet ont orienté le choix de contrôle vers un contrôle par la cinématique articulaire. Tout d'abord, cela permet d'explorer son application au domaine de la biomécanique pour déterminer si les résultats d'une telle simulation sont suffisamment robustes pour l'application visée. Ensuite, si cela s'avère possible, un contrôleur de ce type correctement conçu pourrait alors théoriquement simuler différentes situations bien plus rapidement qu'une simulation optimisant l'ensemble du mouvement. En effet, les optimisations en dynamique directe les plus récentes qui utilisent des méthodes de contrôle optimal avec la formulation en « direct collocation » permettant

de diviser par 20 le temps de calcul, prennent tout de même environ 36 minutes pour simuler une foulée entière (Falisse et al., 2019). Un second avantage à l'utilisation du contrôle par cinématique articulaire est l'intuitivité qu'elle permet pour dégrader le contrôleur ou pour y ajouter des éléments. En effet, puisque ce type de contrôle est composé de différentes briques ayant chacune leur rôle, il est plus facile de cibler un élément particulier de la marche à dégrader pour prendre en compte une pathologie. Si dans tous les types de simulation il est possible de limiter les couples articulaires ou les forces musculaires pour simuler une faiblesse musculaire, dans un simulateur utilisant des briques de contrôle, il est par exemple possible de dégrader la position de la pose du pied pour simuler un manque d'équilibre ou de dégrader le suivi de la trajectoire du pied pour simuler un manque de coordination.

Etant donné le choix du contrôleur, le modèle utilisé est un modèle squelettique 3D actionné par des couples articulaires.

De plus, puisque le logiciel OpenSim est utilisé, le système d'axe utilisé dans le logiciel est utilisé pour l'ensemble du projet. Ainsi, l'axe X représente l'axe antéropostérieur, l'axe Y représente l'axe vertical, et l'axe Z représente l'axe médio-latéral.

## A. Définitions et notations

Le travail réalisé pour ce projet consiste majoritairement en du développement de script informatique dont la compréhension nécessite certaines notions de base. De même certaines notions en biomécanique ne sont pas décrites de la même manière dans les différents éléments de la littérature. Les notions et notations de bases utilisées dans ce projet sont explicitées ici pour être utilisées dans la suite de la description du simulateur.

### 1. Définitions et notations des paramètres de la marche

Pour ce projet, un cycle de marche est défini de la pose du pied à la pose subséquente du même pied (Figure 11). Le cycle est alors divisé en quatre phases distinctes. La première phase est appelée premier double appui et correspond au moment où les deux pieds sont en appui sur le sol après que le pied se soit posé sur le sol. La deuxième phase est appelée simple appui et correspond au moment où le pied controlatéral est en l'air tandis que le pied ipsilatéral est le seul appui du corps sur le sol. La troisième phase est appelée second double appui et correspond au moment où les deux pieds sont au sol après que le pied controlatéral se soit posé sur le sol. La quatrième phase est appelée phase oscillante et correspond au moment où le pied ipsilatéral n'est plus appuyé sur le sol. Le passage d'une phase à l'autre est appelé un évènement. Un cycle de marche est ainsi constitué de cinq évènements. Le premier évènement est la pose du pied sur le sol marquant le début du cycle. Il est noté « HS1 » pour Heel Strike. Le deuxième évènement est le lever du pied controlatéral noté « CTO » pour Contralateral Toe Off. Le troisième évènement est la pose du pied controlatéral noté « CHS » pour Contralateral Heel

Strike. Le quatrième évènement est le lever du pied noté « TO » pour Toe Off. Enfin, le cinquième évènement est la pose du pied marquant la fin du cycle et le début du suivant, noté « HS2 ». La marche étant symétrique, le cycle d'un côté du corps est décalé par rapport au côté opposé.

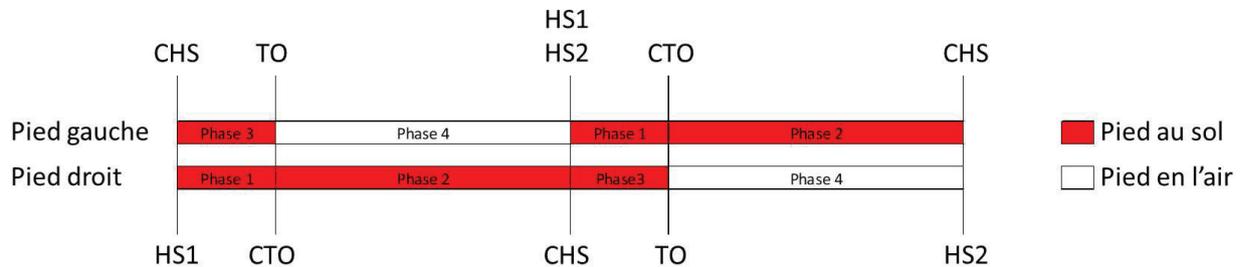


Figure 11 : Représentation des phases de marche et des évènements

## 2. Programmation orientée objet

### 2.1. Types de variables et de conteneurs

En C++, les types de variables doivent être précisés à leur création, et sont également précisés lors de leur présentation en annexe. Un *int* est utilisé pour stocker un nombre entier. Un *double* est utilisé pour stocker un nombre réel jusqu'à 15 décimales. Un *size\_t* contient un entier pouvant indiquer la taille d'un conteneur quelle que soit sa taille ou sa forme. Un *string* est utilisé pour stocker des chaînes de caractères. Un tableau est un conteneur dont la taille est invariable et doit être spécifié lors de sa déclaration. Enfin, un *vector* est également un conteneur mais dont la taille peut varier.

Les fonctions d'OpenSim font également appel à la notion de pointeur. Les variables et objets sont stockés dans des cases mémoire. Lorsqu'une variable ou un objet est passé comme argument dans une fonction, sa valeur est copiée dans une case mémoire différente et accessible par la fonction. Ainsi, si une opération est effectuée sur la variable dans la fonction, la variable hors de la fonction reste inchangée. Cependant, dans le cas où la variable ou l'objet à passer est un conteneur de grande dimension ou que l'on souhaite réaliser des opérations dessus, copier l'ensemble des valeurs du conteneur peut s'avérer une perte de temps et d'espace de stockage et les modifications ne seront pas directement appliquées à l'objet de base. Il est alors possible de passer en argument un pointeur vers la variable ou l'objet qui représente l'adresse de sa case mémoire. Ainsi, la fonction a directement accès à la case mémoire de la variable, les données n'ont pas besoin d'être copiées et la variable originelle peut être modifiée directement depuis la fonction.

## 2.2. Objets

Le langage C++ utilisé pour développer le simulateur est un langage de programmation orienté objet (POO) et le code développé profite largement de cette spécificité. Le principe de la POO est de définir des objets pouvant représenter des concepts ou des idées et de les manipuler. Un objet est défini par sa structure interne composée de variables et de fonctions.

Le moyen pour définir des objets est la « classe ». Une classe est constituée de membres. Ces membres peuvent être privés, auquel cas ils ne sont accessibles qu'à partir de la classe, ou publiques, auquel cas ils peuvent être accessibles hors de la classe. Les membres sont soit des données membres soit des fonctions membres. Les fonctions membres peuvent par exemple servir à modifier ou acquérir (retourner) la valeur d'une donnée membre ou à effectuer des opérations. Une fois définie, une classe peut être utilisée pour instancier des objets. Ces objets ont alors leurs propres instances des données membres et les fonctions membres de la classe permettent de les manipuler. Les fonctions membres peuvent nécessiter que des données supplémentaires leur soient transmises lorsqu'elles sont appelées. On parle alors de « paramètres formels » lorsque l'on fait référence au type de ces données et à leur nom, et l'on parle de « paramètres effectifs » lorsque l'on fait référence à la valeur des données transmises lors d'un appel particulier. Un autre avantage du système de classe est qu'une classe peut être utilisée comme parent pour définir une classe enfant. Cette nouvelle classe hérite alors des membres de la classe parent, tout en ayant la possibilité de se voir définir des membres supplémentaires.

Les différentes briques de contrôle utilisées dans le simulateur sont toutes des objets instanciés d'une classe spécifique. De plus, comme chaque classe n'est utilisée pour n'instancier qu'un seul objet, le nom de ces objets correspond au nom de la classe. Pour faire la différence lorsque l'on parle de la classe ou de l'objet instancié de cette classe, on se réfère à la convention de nommage suivante. Le nom d'une classe commence par une lettre majuscule (ex : *ReferenceTrajectories*) tandis que le nom d'un objet commence par une lettre minuscule (ex : *referenceTrajectories*). De même, le nom des fonctions, variables et données commencent par une minuscule. Pour différencier les fonctions des variables, les noms des fonctions commencent généralement par une action (ex : *get...*, *set...*). Cette convention est respectée dans le code développé et dans sa description ci-après.

## B. Modèle et solveur

### 1. Modèle squelettique

Le modèle est créé via un code C++ en utilisant les bibliothèques de fonctions d'OpenSim. Dans OpenSim, le modèle est un objet défini par une classe nommée *Model*. Les différents éléments constituant le modèle sont eux-mêmes définis par des classes. Les segments sont représentés par des objets définis par la classe *Body*, dont la construction prend comme paramètres le nom du segment, sa masse, la position de son CoM et son inertie par rapport au CoM. Les articulations sont représentées par des objets définis par la classe *CustomJoint*, prenant comme paramètres le nom de l'articulation, le segment parent, la position et l'orientation de l'articulation dans le repère du segment parent, le segment enfant, la position et l'orientation de l'articulation dans le repère du segment enfant, et un objet de la classe *SpatialTransform* défini par un objet de la classe *TransformAxis* qui décrit le ou les axes de rotations de l'articulation. Enfin, les couples articulaires sont représentés par des objets définis par la classe *TorqueActuator* prenant comme paramètres le segment parent, le segment enfant, l'axe du couple et l'information précisant si l'axe de rotation est exprimé dans le repère global ou dans le repère du segment parent. Dans ce projet, tous les axes de couples articulaires sont exprimés dans le repère du segment parent. En fin de code, le modèle est décrit dans un fichier OSIM qui est une extension spécifiquement utilisée par OpenSim.

La majorité des modèles utilisés pour l'analyse de la marche avec OpenSim sont des modèles se concentrant sur les membres inférieurs et dans lesquels les segments supérieurs sont regroupés en un seul segment rigide HAT (Head, Arms, Trunk). Afin de prendre en compte tous les éléments pouvant influencer sur la marche, il a été décidé pour ce projet de différencier les différents segments des membres supérieurs. Ainsi le modèle utilisé est un modèle à 16 segments et 31 DoF, dont six représentent la position et l'orientation du modèle dans le repère global et 25 représentent les articulations (Figure 12). Le segment de référence qui détermine la position et l'orientation du modèle est le pelvis. Chaque membre inférieur comporte cinq segments représentant les cuisses, les jambes, les talus, les pieds et les orteils. Le tronc et la tête sont réunis en un seul segment. Enfin, les bras et avant-bras sont chacun représentés par un segment. Les mains sont fusionnées avec les avant-bras. L'intégralité des articulations sont modélisées par des liaisons pivot. Les articulations des hanches, des lombaires et des épaules étant normalement des liaisons rotules, elles sont modélisées par un enchainement de trois liaisons pivots concourantes représentant respectivement la flexion extension, l'adduction abduction et la rotation interne externe des membres. Les propriétés des segments et des articulations sont résumées dans les Tables 1 et 2.

Segment	Masse (kg)	Position du CoM (m)	Matrice d'inertie** (kg.m <sup>2</sup> )
<b>Pelvis</b>	11.777	(-0.0707, 0, 0)	(0.1028, 0.0871, 0.0579)
<b>Fémur*</b>	9.3014	(0, -0.17, 0)	(0.1339, 0.0351, 0.1412)
<b>Tibia*</b>	3.7075	(0, -0.1867, 0)	(0.0504, 0.0051, 0.0511)
<b>Talus*</b>	0.1	(0, 0, 0)	(0.001, 0.001, 0.001)
<b>Calcaneum*</b>	1.25	(0.1, 0.03, 0)	(0.0014, 0.0039, 0.0041)
<b>Orteils*</b>	0.2166	(0.0346, 0.006, -0.0175)	(0.0001, 0.0002, 0.001)
<b>Tronc</b>	26.8266	(-0.03, 0.32, 0)	(1.4745, 0.7555, 1.4314)
<b>Humérus*</b>	2.0325	(0, -0.164502, 0)	(0.011946, 0.004121, 0.013409)
<b>Ulna*</b>	0.6075	(0, -0.120525, 0)	(0.002962, 0.000618, 0.003213)
<b>Radius*</b>	0.6075	(0, -0.120525, 0)	(0.002962, 0.000618, 0.003213)
<b>Main*</b>	0.4575	0, -0.068095, 0	(0.000892, 0.000547, 0.00134)

Table 1 : Caractéristiques des segments du modèle. \* : Les segments présentés sont ceux de droite. Les segments de gauche ont les mêmes caractéristiques à la différence que la position du CoM médio latéral est de signe opposé. \*\* : Les éléments présentés sont les éléments de la diagonale de la matrice d'inertie car les autres valeurs sont nulles.

Articulation	Segment parent	Position dans le segment parent (m)	Segment enfant	Axe de rotation
<b>Hanche*</b>	Pelvis	(-0.0707, -0.0661, 0.0835)	Fémur	(0, 0, 1)
				(1, 0, 0)
				(0, 1, 0)
<b>Genou*</b>	Fémur	(-0.00449986, -0.395821, 0)	Tibia	(0, 0, 1)
<b>Cheville*</b>	Tibia	(0, -0.43, 0)	Talus	(-0.10501355, -0.17402245, 0.97912632)
<b>Subtalaire*</b>	Talus	(-0.04877, -0.04195, 0.00792)	Calcaneum	(0.78717961, 0.60474746, -0.12094949)
<b>Métatarso phalangienne (MTP)*</b>	Calcaneum	(0.1788, -0.002, 0.00108)	Orteils	(-0.58095440, 0, 0.81393611)
<b>Lombaire</b>	Pelvis	(-0.1007, 0.0815, 0)	Tronc	(0, 0, 1)
				(1, 0, 0)
				(0, 1, 0)
<b>Epaule*</b>	Tronc	(0.003155, 0.3715, 0.17)	Fémur	(0, 0, 1)
				(1, 0, 0)
				(0, 1, 0)
<b>Coude*</b>	Fémur	(0.013144, -0.286273, -0.009595)	Ulna	(0.22604696, 0.02226900, 0.97386183)
<b>Radio-Ulnaire**</b>	Ulna	(-0.00672700, -0.01300700, 0.02608300)	Radius	Aucun
<b>Poignet**</b>	Radius	(-0.00879700, -0.23584100, 0.01361000)	Main	Aucun

Table 2 : Caractéristiques des articulations. Les orientations et la position dans le repère enfant ne sont pas indiquées car toujours nulles. \* : Les articulations présentées sont celles de droite. Les articulations de gauche ont les mêmes caractéristiques à la différence que la position des articulations dans la direction médio latérale et les composantes sur x et y des axes de rotation

*sont de signes opposés. \*\* : Ces articulations sont définies comme des liaisons encastrement et ne comptent donc pas dans le nombre de DoF. Pour la même raison, l'ulna, le radius et la main ne comptent que pour un seul segment.*

Une première spécificité de ce modèle comparé à ceux généralement utilisés dans OpenSim est la position de l'axe de rotation du genou. En général, la position de l'axe dans le repère du fémur varie en fonction de la valeur de l'angle du genou pour modéliser les déplacements antéro-postérieur et proximo-distal du tibia par rapport au fémur. Cependant, dans cette configuration, le couple articulaire n'est pas appliqué au niveau d'un axe de rotation fixe. Or, plusieurs éléments du contrôle sont simplifiés si l'axe de rotation reste fixe. Il a donc été décidé de donner la priorité aux éléments de contrôle et de simplifier le modèle en définissant un axe de rotation du genou fixe dans le repère de la cuisse.

Une seconde spécificité du modèle est la prise en compte de l'articulation métatarso-phalangienne. Dans beaucoup de modèles d'OpenSim, celle-ci est considérée comme une liaison encastrement. Dans les simulations en dynamique directe, pour éviter les problèmes liés à la modélisation du pied et des orteils en un seul segment, le modèle de contact n'agit que jusqu'au niveau de l'articulation métatarso-phalangienne, permettant aux orteils de pénétrer dans le sol. Encore une fois, dans un souci de prise en compte de tous les éléments pouvant influencer sur la marche, il a été décidé d'autoriser cette articulation à bouger.



Figure 12 : Modèle squelettique

## 2. Modèle de contact

### 2.1. Force normale

Pour modéliser les forces de contacts, OpenSim implémente le modèle de Hunt-Crossley calculant la force de réaction en fonction de la distance d'interpénétration des géométries de contact Eq. (1) (Sherman et al., 2011). Dans le cadre de notre simulateur, les seuls contacts sont entre le sol et les pieds. Ainsi une première géométrie modélise le sol par un volume infini dans les directions antéro-postérieures et média-latérales, et semi-infini dans la direction normale au sol. Pour les géométries liées au modèle du pied, OpenSim propose l'utilisation de sphères de contact. Le nombre, la position et les caractéristiques des géométries à utiliser pour modéliser un contact le plus fidèlement possible est un sujet de recherche à part entière. Il a donc été décidé pour ce projet d'utiliser un modèle préexistant.

Un modèle intégré de base dans les exemples d'OpenSim utilise trois sphères. La première est positionnée au niveau du talon et les deux autres aux extrémités médiales et latérales de l'articulation métatarso-phalangienne. Ce type de modèle, qui ne prend pas en compte l'interpénétration des orteils avec le sol, est celui utilisé lorsque l'articulation métatarso-phalangienne est considérée comme fixe pour permettre au pied de rouler autour de cette articulation. Etant donné que le modèle squelettique choisi prend en compte le mouvement des orteils, ce modèle ne convient pas, et il est nécessaire d'utiliser un autre modèle. Le modèle choisi est celui développé par Porsa et al. (Porsa et al., 2016) utilisant huit sphères de contact (Figure 13) dont cinq liées au pied et trois aux orteils et dont les caractéristiques ont été déterminées par un processus d'optimisation dans lequel l'objectif était de reproduire les forces verticales expérimentales avec une cinématique du modèle squelettique donnée. Les caractéristiques des géométries de contact optimisées sont mises à l'échelle du modèle si besoin mais les propriétés du contact sont, elles, invariables. Ces caractéristiques sont explicitées dans la table 3.

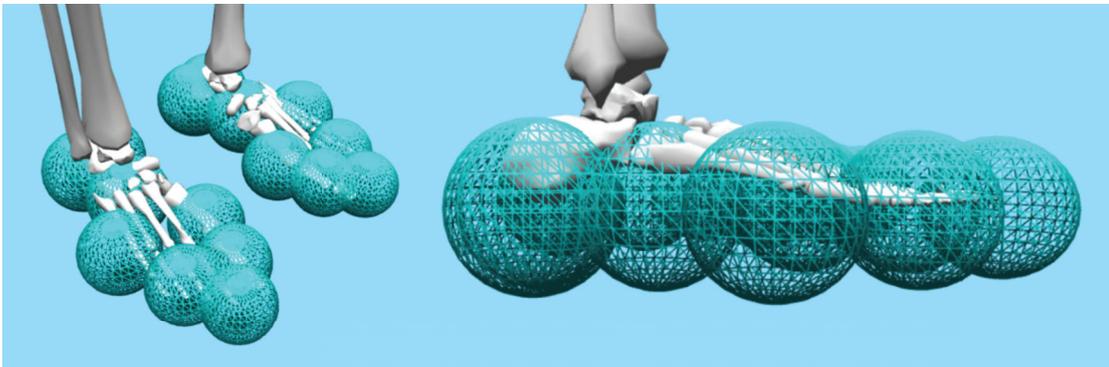


Figure 13 : Géométries de contact

Sphère	Segment	Rayon (m)	Position (m)	Raideur (N/m)	Amortissement (Kg/s)
1	Calcanéum	0.0465	(0.01, -0.01, 0.0075)	14740528.83	0.416320177
2	Calcanéum	0.0465	(0.01, -0.01, -0.02)	14740528.83	0.416320177
3	Calcanéum	0.045	(0.075, -0.01, -0.00175)	14984190.44	0.416320177
4	Calcanéum	0.0419	(0.125, -0.01, 0.05)	15528608.09	0.416320177
5	Calcanéum	0.0398	(0.16, -0.01, -0.035)	15933016.46	0.416320177
6	Orteils	0.0386	(0.02, -0.008, 0.03)	16178784.44	0.416320177
7	Orteils	0.038	(0.035, -0.008, -0.02)	16306011.44	0.416320177
8	Orteils	0.0365	(0.07, -0.008, 0.01)	16637693.09	0.416320177

Table 3 : Caractéristiques des géométries et forces de contact

## 2.2. Forces transversales

Pour calculer les forces transversales, OpenSim implémente le modèle de friction de Coulomb. Cependant, durant le développement du simulateur, il est apparu que ce modèle permettait au pied de glisser trop facilement sur le sol. En effet, la force de réaction étant calculée en fonction de la vitesse

relative des géométries de contact, il est possible d'atteindre une configuration dans laquelle la vitesse relative des géométries soit suffisamment faible pour que la force calculée n'empêche pas suffisamment le glissement. Jugeant ce phénomène non acceptable pour la simulation, il a été décidé de développer un modèle propre à ce projet.

La situation à éviter étant le glissement du pied sur le sol, il a été décidé d'utiliser l'hypothèse d'un contact sans glissement ou contact adhérent. Le modèle créé utilise ainsi des éléments viscoélastiques assimilables à des systèmes de ressorts-amortisseurs pour contrôler la position du pied sur le sol. Ce type de solution a par exemple été utilisé de manière similaire par Anderson et Pandy (Anderson & Pandy, 1999), mais un modèle propre au simulateur est conçu pour ce projet. Deux ensembles de trois éléments viscoélastiques sont associés à chaque pied. Le premier ensemble est placé au point médian des deux sphères de contact liées au talon. Le second est, quant à lui, placé au niveau de la sphère de contact la plus avancée des orteils. Dans les deux cas, le placement des ensembles dans le plan XZ correspond à la position du centre des sphères mais la position sur l'axe Y est située à hauteur du centre des sphères moins le rayon des sphères pour que le point d'application des forces soit au plus proche du sol (Figure 14). Chaque élément, bien que lié aux pieds, exerce une force ou un couple dans le repère global. Les ensembles sont composés d'un élément exerçant une force dans la direction antéropostérieure, d'un élément exerçant une force dans la direction média-latérale, et d'un élément exerçant un couple autour de l'axe normal du sol. La phase d'appui est divisée en trois phases qui déterminent la manière d'utiliser chaque élément.

### 2.2.1. 1<sup>ère</sup> phase

La première phase de contact correspond à la phase de double appui de la marche, lorsque les deux pieds sont en contact avec le sol. Puisqu'un ressort exerce une force en fonction de la variation de longueur qu'il subit, il est nécessaire de définir un point de référence pour les éléments viscoélastiques. La variation de position des éléments par rapport au point de référence peut alors faire office de longueur de déformation pour calculer la force de rappel. Le moyen le plus direct pour déterminer la position de référence est de la définir comme la position des éléments au moment du contact avec le sol. Cependant, cela poserait deux problèmes lors de la simulation.

Le premier problème concerne les éléments liés aux orteils. Après le contact du talon avec le sol, il y a toute une période où le pied roule sur le sol avant d'être à plat. Durant cette période, puisque le pied roule autour du talon, la position des éléments liés aux orteils évolue, principalement dans la direction antéropostérieure, mais aussi dans la direction média-latérale. Déterminer le point de référence des éléments liés aux orteils durant cette période aurait pour conséquence l'apparition d'une force uniquement liée à ce mouvement et donc non naturelle. Pour éviter ce problème, les éléments liés aux orteils ne produisent pas de force durant la première phase de la marche, et leur position de

référence n'est déterminée qu'au début de la deuxième phase, quand le pied est déjà à plat sur le sol et à partir duquel leur position n'est plus censée évoluer. La non-utilisation des éléments liés aux orteils durant cette première phase explique la nécessité d'un élément exerçant un couple autour de l'axe normal du sol. En effet, lorsque les éléments liés au talon et aux orteils exercent une force simultanément, l'action des deux forces média-latérales, et dans une moindre mesure les deux forces antéropostérieures, provoquent un couple empêchant le pied de tourner sur le sol. Ce couple de force n'étant pas présent lors de la première phase de la marche, un élément viscoélastique similaire à un ressort de torsion applique un couple au niveau du talon en fonction d'une orientation de référence définie lors du contact du talon avec le sol. L'orientation du pied est définie par l'angle entre l'axe antéropostérieur du repère global et la droite décrite par la projection des positions des deux ensembles d'éléments de contact sur le plan du sol (Figure 15).

Le second problème concerne les éléments liés au talon, et est également dû à la phase de roulement du pied après contact. Dans la continuité de l'hypothèse de contact sans glissement, le roulement du pied sur le sol devrait lui aussi se faire sans glissement et la position du point d'application des forces devrait donc évoluer dans le repère global. Fixer la position de référence de ces éléments au moment du contact génèrerait alors des forces contraignant les éléments à une position fixe, provoquant un phénomène où le talon tournerait sur le sol sans rouler. Pour éviter ce phénomène, la position de référence des éléments liés au talon est effectivement déterminée au moment du contact, mais évolue durant toute la première phase de contact en fonction de l'évolution de l'angle du pied par rapport au sol. L'angle du pied dans le plan sagittal est défini par l'angle entre l'axe vertical du sol et la projection de l'axe vertical du pied sur le plan sagittal. L'angle du pied dans le plan frontal est défini par l'angle entre l'axe médio latéral du sol et la projection de l'axe médio latéral du pied sur le plan sagittal (Figure 15). La position adaptée de la position de référence des éléments viscoélastiques dans la direction antéropostérieure est alors déterminée par :

$$X0_{Adapte} d_1 = X0_1 + (Zr0 - Zr) * r_{sphère}$$

où  $X0_{Adapte} d_1$  est la position de référence adaptée de l'élément,  $X0_1$  est la position de référence initiale de l'élément,  $Zr0$  est l'angle initial du pied par rapport au sol dans le plan sagittal,  $Zr$  est l'angle actuel du pied par rapport au sol dans le plan sagittal, et  $r_{sphère}$  est le rayon des sphères de contact liées au talon. L'indice 1 indique que les valeurs sont celles de l'élément lié au talon.

Enfin, pour éviter le problème de discontinuité de force au contact due à la vitesse non nulle à cet instant, un coefficient est appliqué à la force de réaction. Ce coefficient évolue linéairement de 0 à 1 entre le début de la phase d'appui et 5% de la durée d'un cycle de marche noté  $rTime$  pour le côté droit et  $lTime$  pour le côté gauche :

$$coef = \begin{cases} \frac{rTime}{5} & \text{si } rTime < 5 \\ 1 & \text{si } rTime > 0 \end{cases}$$

$$force X_1 = (k_x * (X0_{Adapte} d_1 - X_1) - 2 * \sqrt{k_x} * Xv_1) * coef$$

où  $force X_1$  est la force de réaction de l'élément antéropostérieur,  $k_x$  est la raideur de l'élément antéropostérieur,  $X_1$  est la position de l'élément dans la direction antéropostérieure et  $Xv_1$  est la vitesse de l'élément dans la direction antéropostérieure. La force médio latérale et le couple vertical sont calculés de la même manière :

$$force Z_1 = (k_z * (Z0 Adapte d_1 - Z_1) - 2 * \sqrt{k_z} * Zv_1) * coef$$

$$couple Y_1 = (k_y * (Yr 0 Adapte d_1 - Yr_1) - 2 * \sqrt{k_y} * Yrv_1) * coef$$

### 2.2.2. 2ème phase

La deuxième phase commence en même temps que la phase de simple appui et s'étend jusqu'à 60% de la phase de simple appui noté *swingTime*. Au début de cette phase, la position de référence des éléments liés au talon qui évoluait en fonction de l'angle du pied sur le sol, se fixe à sa position actuelle. La position de référence des éléments liés aux orteils ainsi que la position adaptée sont quant à elles définies comme étant leur position actuelle. Pendant toute une partie de la phase d'appui, le pied est à plat sur le sol. Cependant, à la fin de celle-ci, le pied commence à rouler autour de l'articulation métatarso-phalangienne provoquant le même phénomène au niveau du talon qu'au niveau des orteils lors de la première phase. Il est donc nécessaire de faire en sorte que l'élément antéropostérieur lié au talon n'exerce plus de force à ce moment-là. Pour ce faire, la force antéropostérieure exercée par l'élément est soumise à un coefficient évoluant linéairement en fonction de la durée de la phase de simple appui. Le coefficient appliqué à l'élément du talon évolue linéairement de 1 à 0 entre 0% et 60% de la phase de simple appui. A l'inverse, le coefficient appliqué aux éléments des orteils évolue de 0 à 1 sur la même période. Ainsi, une transition du point d'application des forces est opérée sans discontinuité du talon vers les orteils. Quant à la force exercée par l'élément médio latéral lié au talon, elle n'est pas soumise au coefficient pour permettre de générer le couple empêchant le pied de tourner sur le sol. De plus, comme explicité précédemment, les couples de forces exercés par les deux ensembles d'éléments permettent d'assurer la non-rotation du pied sur le sol. L'élément appliquant un couple lié aux orteils n'est donc pas utilisé pendant cette phase, et celui lié au talon diminue en même temps que la force antéropostérieure via l'application du même coefficient.

$$coef = \frac{-swingTime}{60} + 1$$

$$force X_1 = (k_x * (X0 Adapte d_1 - X_1) - 2 * \sqrt{k_x} * Xv_1) * coef$$

$$force Z_1 = k_z * (Z0 Adapte d_1 - Z_1) - 2 * \sqrt{k_z} * Zv_1$$

$$couple Y_1 = (k_y * (Yr 0 Adapte d_1 - Yr_1) - 2 * \sqrt{k_y} * Yrv_1) * coef$$

$$force X_2 = (k_x * (X0 Adapte d_2 - X_2) - 2 * \sqrt{k_x} * Xv_2) * (1 - coef)$$

$$force Z_2 = (k_x * (Z0 \text{ Adapte } d_2 - Z_2) - 2 * \sqrt{k_x} * Zv_2) * (1 - coef)$$

$$couple Y_2 = 0$$

### 2.2.3. 3<sup>ème</sup> phase

La 3<sup>ème</sup> phase comprend le reste de la phase d'appui. Seuls les éléments liés aux orteils et l'élément médio latéral lié au talon exercent une force et les valeurs de référence adaptées sont à nouveau mises à jour en fonction de l'angle du pied par rapport au sol :

$$force Z_1 = k_z * (Z0 \text{ Adapte } d_1 - Z_1) - 2 * \sqrt{k_z} * Zv_1$$

$$force X_2 = (k_x * (X0 \text{ Adapte } d_2 - X_2) - 2 * \sqrt{k_x} * Xv_2) * (1 - coef)$$

$$force Z_2 = (k_x * (Z0 \text{ Adapte } d_2 - Z_2) - 2 * \sqrt{k_x} * Zv_2) * (1 - coef)$$

Pour s'assurer que les forces de frictions soient bien nulles au moment du lever du pied, elles sont limitées pour ne pas être supérieures à la force de réaction normale calculée par le modèle de Hunt-Crossley notées  $rGrfY$  pour le pied droit et  $lGrfY$  pour le pied gauche :

$$force Z_1 = \begin{cases} force Z_1 \text{ si } |force Z_1| < rGrfY \\ \frac{force Z_1 * rGrfY}{|force Z_1|} \text{ si } |force Z_1| > rGrfY \end{cases}$$

$$force X_2 = \begin{cases} force X_2 \text{ si } |force X_2| < rGrfY \\ \frac{force X_2 * rGrfY}{|force X_2|} \text{ si } |force X_2| > rGrfY \end{cases}$$

$$force Z_2 = \begin{cases} force Z_2 \text{ si } |force Z_2| < rGrfY \\ \frac{force Z_2 * rGrfY}{|force Z_2|} \text{ si } |force Z_2| > rGrfY \end{cases}$$

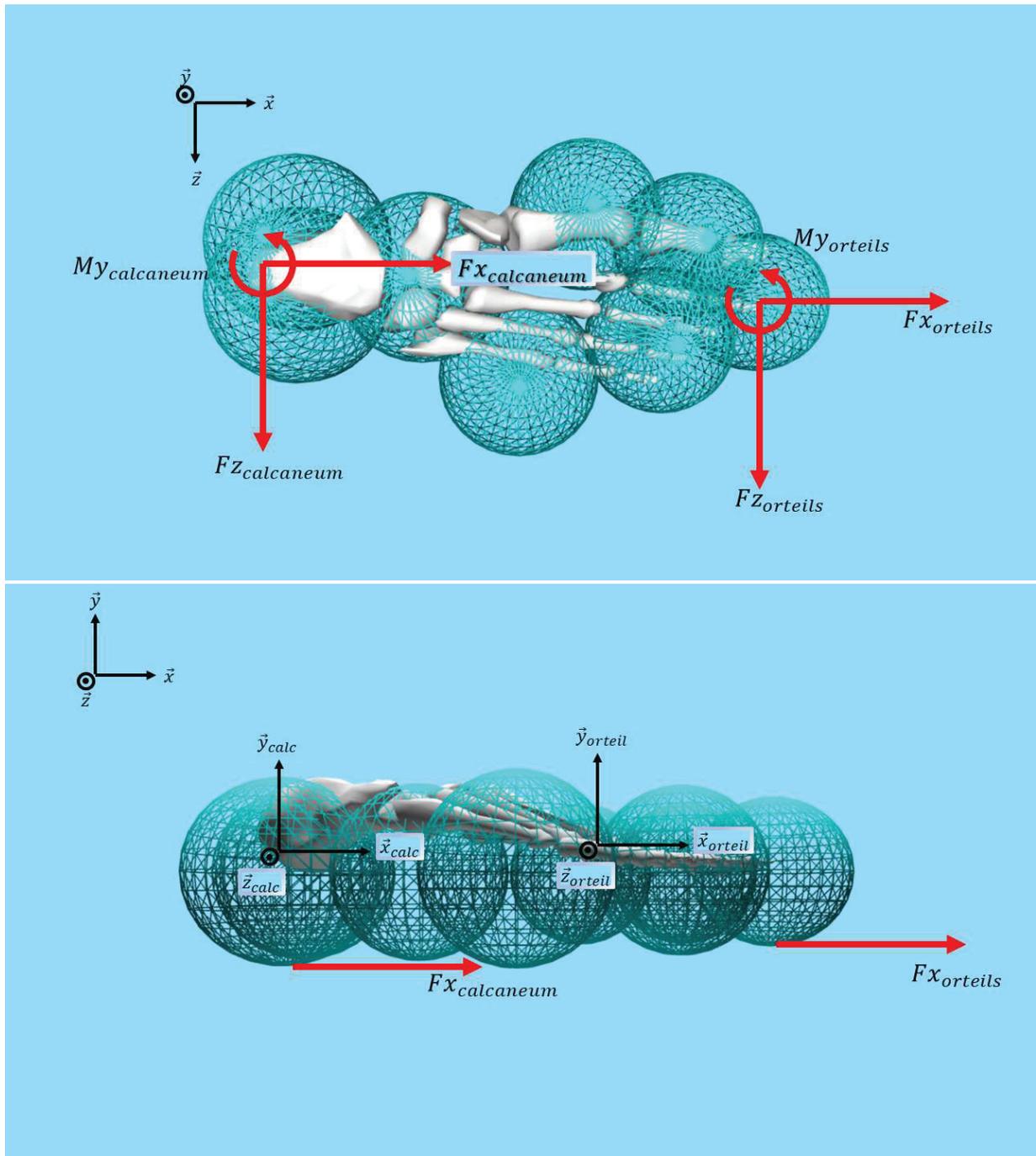


Figure 14 : Modélisation des forces de contacts horizontales.

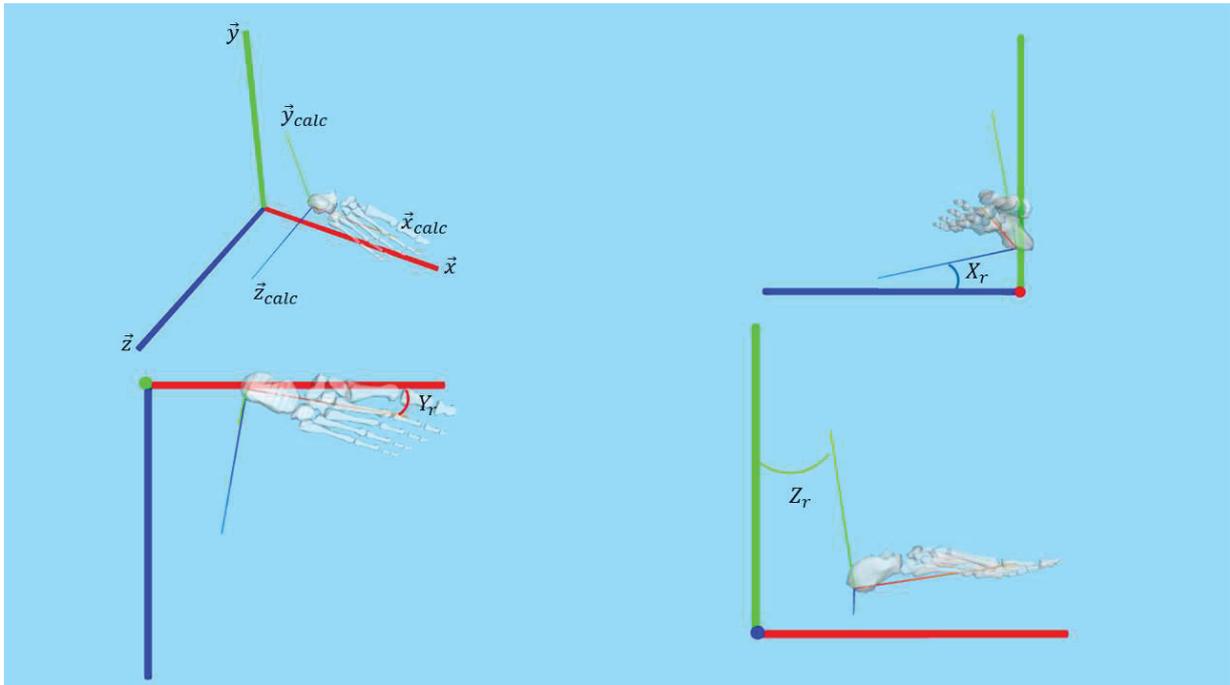


Figure 15 : Définition des angles entre les axes du pied et du repère global.

$Y_r$  est utilisé pour le calcul du moment de réaction.

$Z_r$  est utilisé pour calculer le roulement dans le plan sagittal.

$X_r$  est utilisé pour calculer le roulement dans le plan frontal.

## C. Détermination des trajectoires et temporalités de référence

Une simulation par contrôle de la cinématique articulaire a besoin de trajectoires de référence dont la détermination peut se faire de différentes manières. L'un des objectifs de ce projet étant la biofidélité du mouvement, il a été décidé d'utiliser des données expérimentales issues de la capture du mouvement. La méthode choisie fait appel à une base de données de captures de mouvement, préexistante pour la marche. Une autre base de données a été obtenue durant le projet pour le passage d'obstacle (cf. partie V). Les données sont utilisées pour déterminer des poses clefs des différentes trajectoires et pour définir un modèle statistique afin de déterminer leurs valeurs en fonction des caractéristiques de la simulation. Les trajectoires sont alors construites par interpolation entre ces poses clefs afin d'obtenir des trajectoires continues. En plus des trajectoires de référence, le modèle statistique est également utilisé pour déterminer des paramètres spatio-temporels de la marche.

## 1. Traitement de la base de données

La base de données utilisée provient de Moissenet et al. (Moissenet et al., 2019) réunissant 54 sujets. Il a été demandé à chaque sujet de marcher dans cinq conditions de vitesses différentes. Les trois premières conditions imposaient un intervalle de vitesse tandis que les deux dernières se faisaient à vitesse normale et rapide toutes deux choisies par les sujets. Chaque condition de marche a été répétée plusieurs fois pour disposer de plusieurs essais par condition.

### 1.1. Obtention des DoF expérimentaux

Les fichiers composant la base de données étant des fichiers *c3d*, la première étape consistait à obtenir les valeurs des DoF. Ces valeurs ont été obtenues grâce à l'outil de cinématique inverse d'OpenSim. Cet outil nécessite tout d'abord un modèle mis à l'échelle du sujet à traiter. Tous les modèles utilisés ont pour base le modèle décrit précédemment. Pour les besoins de la mise à l'échelle, le modèle est complété par des marqueurs virtuels correspondant aux marqueurs utilisés pour la capture de mouvement de la base de données (Figure 16).



Figure 16 : Marqueurs virtuels pour la mise à l'échelle et cinématique inverse

Les outils de mise à l'échelle et de cinématique inverse n'utilisant pas des fichiers *c3d* mais des fichiers *trc*, il est nécessaire de convertir les fichiers dans le bon format au préalable. La conversion a été effectuée grâce à un script python développé dans le cadre de ce projet. En fournissant à l'outil de mise à l'échelle le modèle de base, un essai de capture de mouvement statique au format *trc* et un fichier *xml* décrivant les propriétés du sujet et le placement des marqueurs, l'outil OpenSim de mise à l'échelle détermine des coefficients de mise à l'échelle pour chaque segment de sorte à minimiser la distance entre les marqueurs du fichier de capture de mouvement et les marqueurs virtuels du modèle. Les coefficients sont alors utilisés pour mettre à l'échelle la longueur mais aussi la masse de chaque segment. En utilisant le modèle mis à l'échelle, un fichier au format *trc* du mouvement à étudier et un fichier *xml* précisant les conditions à appliquer au calcul, l'outil OpenSim de cinématique inverse détermine alors à chaque pas de temps la configuration des DoF optimale pour minimiser la différence entre les marqueurs du fichier de capture de mouvement et les marqueurs virtuels du modèle en formulant le problème par la méthode des moindres carrés pondérés. Pour ce projet, tous les poids sont définis identiques. Les DoF dont la trajectoire est obtenue sont les articulations mais aussi les six DoF indiquant la position et l'orientation du pelvis dans le repère global. L'avantage d'utiliser l'outil de cinématique inverse d'OpenSim plutôt qu'un programme déterminant directement les angles articulaires via les fichiers *c3d* est que, de cette façon, le modèle cinématique (nombre de DoF, direction des axes) correspond directement au modèle qui sera utilisé lors des simulations.

## 1.2. Trajectoires et paramètres spatio-temporels expérimentaux

Les trajectoires articulaires ne sont pas les seules nécessaires au simulateur. En effet, celui-ci fait également appel aux trajectoires du CoM et du pied en phase oscillante. Pour simplifier le traitement des données, les valeurs temporelles des trajectoires sont exprimées en pourcentage du cycle de marche, excepté la trajectoire du pied en phase oscillante exprimée en pourcentage de la phase oscillante. De plus, certains paramètres spatio-temporels sont également nécessaires tels que la durée de la foulée en seconde, la longueur du pas en mètre et la durée des différentes phases de la marche en pourcentage du cycle de marche. Les sujets ayant effectué plusieurs essais par condition, les trajectoires et valeurs de paramètres sont définis comme la moyenne de ceux des différents essais, ce qui permet d'obtenir un seul ensemble de trajectoires et de paramètres par sujet et par condition. La détermination des différentes trajectoires et paramètres spatio-temporels se fait via un script python développé pour le projet.

### 1.2.1. Evènements

Avant de déterminer les trajectoires et paramètres spatiaux-temporels il est nécessaire d'obtenir les informations temporelles sur les évènements du cycle de marche. La temporalité de ces évènements est directement incluse dans les fichiers *c3d*. Pour chaque essai, les évènements correspondant à deux foulées, l'une commençant par le membre inférieur droit et l'autre par le membre inférieur gauche, ont pu être récupérés.

Dans les données récupérées, la position temporelle des évènements est exprimée en indice d'images. Connaissant les indices du premier et du dernier évènement d'une foulée, les évènements sont exprimés en pourcentage du cycle de marche avec le HS1 correspondant à 0% et le HS2 correspondant à 100%.

### 1.2.2. Trajectoires articulaires

Les trajectoires articulaires sont directement obtenues via le résultat de la cinématique inverse en ne gardant que la partie du mouvement comprise entre le premier et le dernier évènement. De la même manière que les évènements, la position temporelle de chaque pas de temps des trajectoires sont exprimées en indice d'images. Elles sont donc interpolées pour être exprimées en pourcentage du cycle de marche.

### 1.2.3. Trajectoire du CoM

La position du CoM du modèle n'est pas explicitée dans le résultat de la cinématique inverse. Il faut donc le déterminer autrement. La méthode utilisée consiste à la calculer via la moyenne des CoM de chaque segment. En effet, puisque l'outil de cinématique inverse utilisé fait appel au modèle décrit précédemment, ses caractéristiques, dont la position des segments entre eux et la position de leur CoM dans leur repère propre, sont connues.

Les matrices de passage entre chaque segment sont tout d'abord définies. Les angles entre les segments sont directement obtenus via le résultat de la cinématique inverse. Les distances entre l'origine des repères sont quant à elles définies par les distances du modèle de base qui sont donc connues au préalable, ainsi qu'avec les coefficients de mise à l'échelle obtenus via l'outil de mise à l'échelle d'OpenSim. Pour illustrer le processus, voici par exemple les calculs utilisés pour déterminer la position globale du CoM de la cuisse droite.

Les matrices de passage sont déterminées par l'opérateur homogène :

$${}^0_sT = \begin{matrix} \vec{X}_s & \vec{Y}_s & \vec{Z}_s & \vec{OO}_s \\ \vec{X}_0 \\ \vec{Y}_0 \\ \vec{Z}_0 \\ i \end{matrix}$$

Où  ${}^0_sR$  est la matrice de rotation entre les segments 0 et S, et  $\vec{OO}_s$  est la distance entre les origines des segments dans le repère du segment parent. La première matrice de passage entre le repère global et le pelvis se fait par rotation autour de l'axe z (pelvis\_tilt / inclinaison sagittale) :

$$T_{o_{pelvisz}} = \begin{bmatrix} \cos(\text{pelvis}_{tilt}) & -\sin(\text{pelvis}_{tilt}) & 0 & t_x \\ \sin(\text{pelvis}_{tilt}) & \cos(\text{pelvis}_{tilt}) & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Où  $t_x$ ,  $t_y$  et  $t_z$  sont les coordonnées cartésiennes de l'origine du repère du pelvis dans le repère global. La deuxième et la troisième matrice de passage entre le repère global et le pelvis se font par rotation autour de l'axe x (pelvis\_list / inclinaison frontale) et y (pelvis\_rot / rotation axiale) :

$$T_{pelvisz_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\text{pelvis}_{list}) & -\sin(\text{pelvis}_{list}) & 0 \\ 0 & \sin(\text{pelvis}_{list}) & \cos(\text{pelvis}_{list}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{pelvisx_y} = \begin{bmatrix} \cos(pelvis_{rot}) & 0 & \sin(pelvis_{rot}) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(pelvis_{rot}) & 0 & \cos(pelvis_{rot}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La première matrice de passage entre le pelvis et la cuisse se fait par rotation autour de l'axe z (hip\_flex) :

$$T_{pelvisy_{rfemurz}} = \begin{bmatrix} \cos(hip_{flex}) & -\sin(hip_{flex}) & 0 & pelvis_{RfemurX} * pelvis_{coef} \\ \sin(hip_{flex}) & \cos(hip_{flex}) & 0 & pelvis_{RfemurY} * pelvis_{coef} \\ 0 & 0 & 1 & pelvis_{RfemurZ} * pelvis_{coef} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Où  $pelvis_{RfemurX}$ ,  $pelvis_{RfemurY}$  et  $pelvis_{RfemurZ}$  sont les distances sur l'axe X, Y et Z entre les origines du repère du pelvis et de la cuisse dans le modèle de base. En multipliant ces valeurs par le coefficient de mise à l'échelle du pelvis, on obtient la distance entre les origines dans le modèle mis à l'échelle. La deuxième et la troisième matrice de passage entre le repère du pelvis et la cuisse se fait par rotation autour de l'axe x (hip\_add) et y (hip\_rot) :

$$T_{rfemurz_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(hip_{add}) & -\sin(hip_{add}) & 0 \\ 0 & \sin(hip_{add}) & \cos(hip_{add}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{rfemurx_y} = \begin{bmatrix} \cos(hip_{rot}) & 0 & \sin(hip_{rot}) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(hip_{rot}) & 0 & \cos(hip_{rot}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Il est alors possible de déterminer la matrice de passage entre le repère global et le repère de la cuisse :

$$T_{O_{femur}} = T_{O_{pelvis}} \cdot T_{pelvisz_x} \cdot T_{pelvisx_y} \cdot T_{pelvisy_{rfemurz}} \cdot T_{rfemurz_x} \cdot T_{rfemurx_y}$$

Enfin, la position du CoM de la cuisse dans le repère global est calculée à partir de sa position dans le repère local :

$$\left( \overrightarrow{O_{global} CoM_{rfemur}} \right)_{global} = T_{rfemurx_y} \cdot \left( \left( \overrightarrow{O_{rfemur} CoM_{rfemur}} \right)_{rfemur} * femur_{coef} \right) \quad (13)$$

Où  $\left(\overline{O_{rfemur} CoM_{rfemur}}\right)_{rfemur}$  est la position du CoM de la cuisse exprimée dans le repère de la cuisse et  $femur_{coef}$  est le coefficient de mise à l'échelle de la cuisse. Les CoM restants sont ensuite déterminés de la même manière. Une fois les positions des CoM de chaque segment connues dans le repère global, le CoM total est déterminé en calculant la moyenne de tous les CoM des segments. La pondération est elle-même déterminée en fonction de la masse du segment par rapport à la masse totale du modèle.

#### 1.2.4. Trajectoire du pied

La trajectoire du pied dans le repère global lors de la phase oscillante est déterminée de la même manière que celle du CoM. Le point choisi pour situer le pied dans l'espace est le point médian entre les centres des deux sphères de contact représentant le talon. En effet, puisque le talon est la première partie du pied à toucher le sol lors du posé du pied, c'est la position de ce point qui sera contrôlée lors de la simulation pour s'assurer que le pied se pose à l'endroit déterminé par le contrôleur.

#### 1.2.5. Données spatio-temporelles

La première donnée recueillie est la longueur des foulées. Elle est déterminée grâce à la position du talon calculée précédemment et est définie comme la différence entre la position du talon en fin et en début de foulée sur l'axe antéropostérieur.

La deuxième donnée recueillie est la durée des foulées en seconde. La fréquence de capture utilisée lors des expérimentations de capture de mouvement étant connue, la durée des foulées est déterminée par celle-ci et par le nombre d'images contenues entre le premier et le dernier évènement de la foulée.

Enfin, la vitesse du sujet est directement déterminée en fonction de la distance parcourue par le CoM durant le temps d'une foulée.

## 2. Détermination des poses clef

Comme expliqué précédemment, les trajectoires de référence utilisées lors des simulations sont le résultat d'une interpolation entre différentes poses clefs. Les poses clefs sont composées d'une valeur indiquant sa position dans le temps et d'une valeur indiquant l'angle ou la position de la trajectoire à ce moment-là. Elles doivent donc être choisies pour faire en sorte que l'interpolation respecte les caractéristiques des trajectoires de base. Ainsi, elles peuvent être choisies pour leur temporalité (concordance avec un évènement) ou pour une caractéristique de la trajectoire devant être respectée

(extremum, valeur nulle, etc...). Le choix des poses clés à utiliser pour chaque trajectoire a été fait empiriquement de sorte qu'elles soient pertinentes pour tous les sujets ou au moins un maximum d'entre eux. En effet, la variabilité inter sujet peut provoquer des situations dans lesquelles une trajectoire présente une caractéristique pouvant faire office de pose clé chez certains sujets mais pas chez d'autres. Ce phénomène peut également être causé par la différence de vitesse de marche comme le montre la Figure 17 présentant des différences de caractéristiques telles que la position et la valeur du minimum ou la pente en début de cycle.

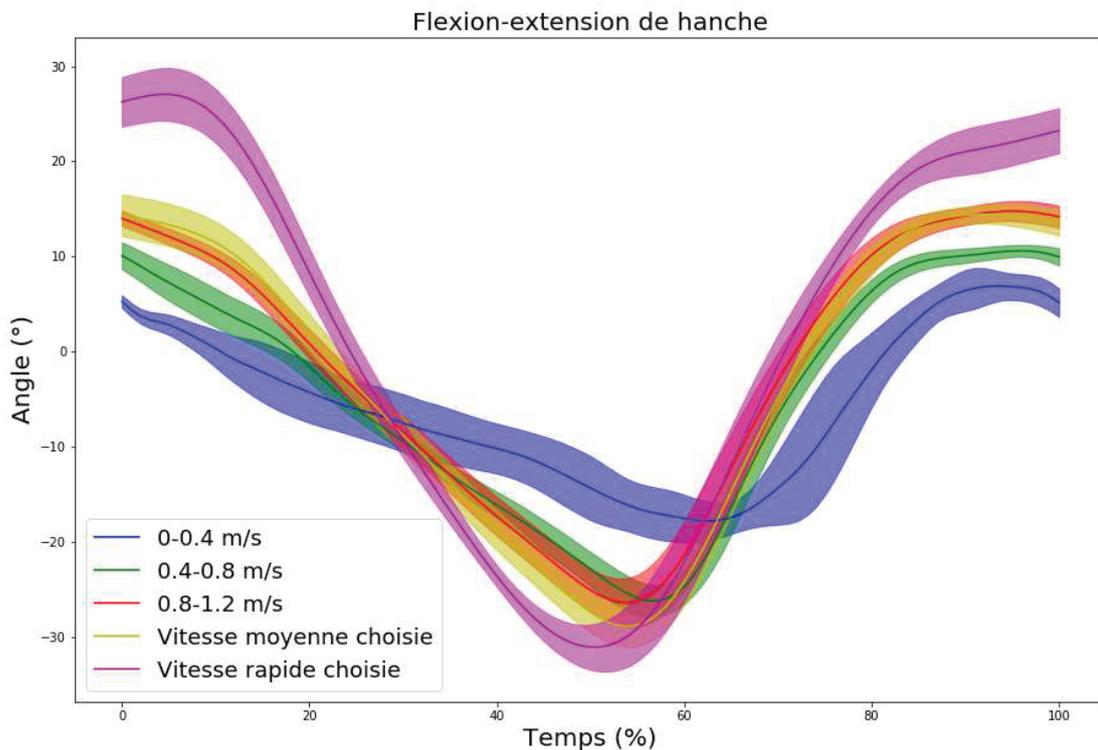


Figure 17 : Exemple de différences de trajectoires d'un sujet en fonction de la vitesse de marche. On remarque que selon la vitesse, la pente en début de cycle peut être de signe différent et donc provoquer un maximum local.

En plus de ces poses clés définies par des caractéristiques des trajectoires, des poses clés complémentaires sont définies à position fixe et régulière.

Les trajectoires de référence sont interpolées par des splines cubiques. Cette méthode d'interpolation est utilisée car elle permet d'assurer la continuité des premières et deuxièmes dérivées. Ces continuités sont importantes dans le cadre d'une simulation de la marche puisque les premières et secondes dérivées des trajectoires représenteront les vitesses et accélérations des angles articulaires, du CoM ou du talon. Or, non seulement des vitesses et accélérations non continues ne seraient pas biofidèles, mais cela provoquerait également des discontinuités et potentiellement des instabilités lors du suivi des trajectoires par le contrôleur. Ce problème de continuité s'applique aussi au passage d'un

cycle de marche à l'autre. En effet, les trajectoires n'étant calculées que pour un cycle de marche, lors d'un changement de cycle, les poses de référence passent immédiatement de la fin de la trajectoire au début de celle-ci. Pour éviter les discontinuités, il faut donc s'assurer que les valeurs de la trajectoire en début et fin de cycle soient égales, mais que leur première dérivée le soit également. Les valeurs des dérivées en début et fin de trajectoires sont donc calculées au même titre que les poses clefs pour être déterminées par le modèle statistique en même temps que les poses clefs. La fonction permettant d'interpoler des splines implémentée dans OpenSim ne permettant pas de contraindre les dérivées en début et fin de courbe, une librairie open source est utilisée à la place (ttk592, 2015/2022). Chaque interpolation nécessite au minimum trois points et les dérivées en début et fin de courbes peuvent être imposées.

La solution par défaut envisagée pour chaque trajectoire est de placer les poses clefs aux moments des évènements. Par la suite, les caractéristiques des trajectoires sont étudiées pour déterminer si une pose clef doit y être associée. Si c'est le cas, la pose est ajoutée à celles correspondant aux évènements ou en remplacement de l'une d'elles si elles sont presque confondues. Les choix des poses clefs pour chaque trajectoire sont explicités ci-après.

## 2.1. Pelvis

Les trajectoires du pelvis n'étant pas directement contrôlées, elles sont tout de même interpolées afin de déterminer la solution initiale de la simulation et n'ont donc pas besoin d'autant de précisions que les trajectoires suivantes. Ainsi, les splines sont uniquement constituées des poses liées aux évènements.

## 2.2. Flexion-extension de hanche

La première caractéristique propice à la définition d'une pose de la trajectoire de flexion-extension de hanche est son minimum aux environ de 50% du cycle de marche. Cela correspond approximativement au CHS. Dans cette configuration, la position du minimum est jugée plus pertinente

pour la trajectoire que la valeur de l'angle lors de l'évènement. La pose clef correspondant à cet évènement est donc remplacée par le minimum. Cependant, placer la pose clef du minimum au bon instant n'assure pas que le minimum de la trajectoire interpolé se situe au bon instant. En effet, l'interpolation fera passer la trajectoire par ce point mais rien ne contraindrait la dérivée de la courbe en ce point à être nulle. Le résultat serait alors que le minimum de la courbe se situe plus bas que la pose clef et que sa position soit décalée. Puisque la fonction utilisée pour l'interpolation permet de définir la dérivée en début et fin de courbe, la solution pour remédier à ce problème est de diviser la trajectoire de la flexion-extension de hanche en deux interpolations différentes. Le point final de la première interpolation est la pose clef du minimum avec une dérivée imposée à 0. Le premier point de la seconde interpolation est alors la même pose clef avec la même condition de dérivée nulle, assurant à la fois la continuité de la trajectoire et de la vitesse et le respect de la valeur et de la position du minimum de la courbe. Cette solution provoque néanmoins un autre problème. En effet, en réalisant une telle transition, bien que la trajectoire et la première dérivée soient continues, la deuxième dérivée ne l'est pas. Par conséquent, la vitesse de référence peut subir des variations plus importantes que sans cette transition. Il y a donc un choix à faire entre le respect de la pente et la continuité des dérivées de la trajectoire de référence. Ce phénomène étant également présent lors du changement de phase, le contrôleur doit de toute façon pouvoir s'adapter à cette instabilité, et il est donc décidé de conserver cette solution pour contraindre la pente de la courbe. La pose liée au CTO est gardée comme point intermédiaire puisqu'elle correspond au moment où la pente de la courbe commence à changer de manière significative. Garder ce point plutôt qu'en prendre un plus tard permet donc de garder la forme globale de la trajectoire. La première interpolation est ainsi constituée du HS1, du CTO et de l'angle minimal.

Un point pouvant sembler important dans la deuxième partie de la trajectoire est le maximum local situé dans les 20 derniers pourcents du cycle. Cependant, en fonction de la condition de marche, ce maximum peut ne pas exister (Figure 18), auquel cas un point où la pente varie plus rapidement semble exister. Une première approche pour trouver une pose clef pertinente à ce moment sans être placé à un instant décidé arbitrairement a été de s'intéresser à la première dérivée de la trajectoire mais n'a abouti à aucune solution concluante. Finalement, la solution retenue a été de s'intéresser à la seconde dérivée de la trajectoire et de définir la pose clef comme étant le minimum de l'accélération après 80% du cycle de marche. De plus, l'évènement du TO étant situé à un moment où la trajectoire est relativement constante (environ 65% du cycle), elle est considérée comme non nécessaire à la détermination de la trajectoire. Ainsi, la seconde interpolation est constituée de l'angle minimal, du minimum d'accélération après 80% du cycle, et du HS2 (Figure 19).

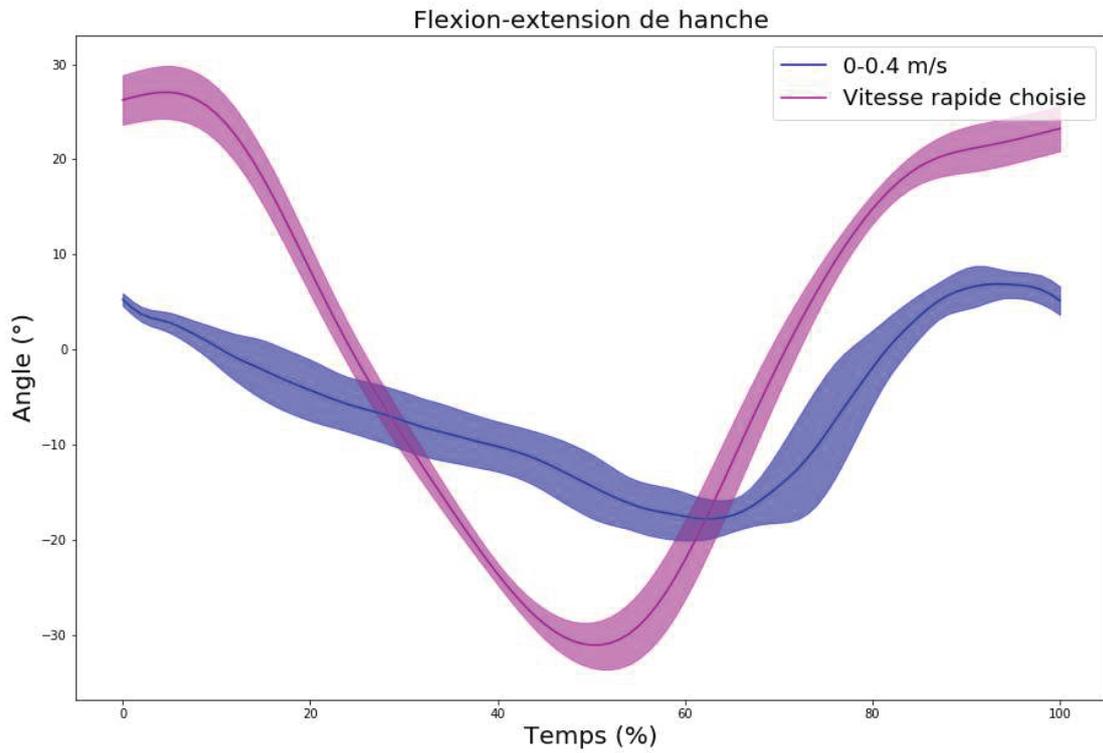


Figure 18 : Exemple de trajectoire de flexio- extension de hanche en fonction de la vitesse de marche

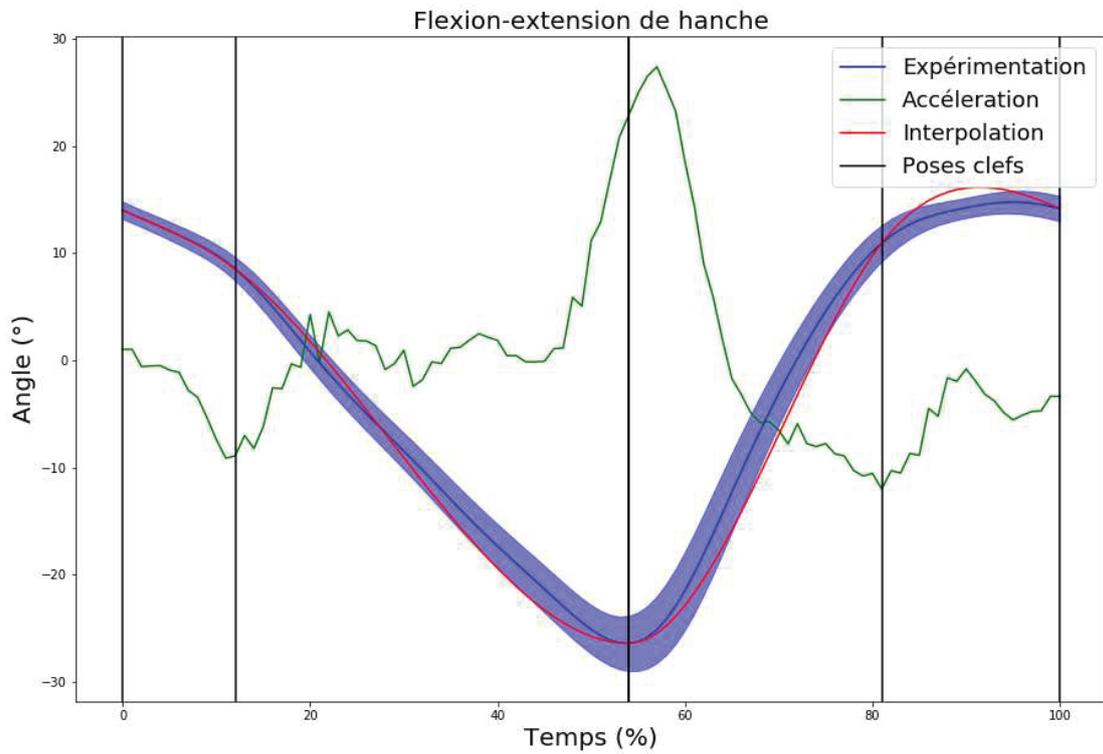


Figure 19 : Trajectoire expérimentale et interpolée, et poses clefs de la flexion extension de hanche

### 2.3. Adduction-abduction de hanche

La trajectoire d'adduction-abduction de hanche varie beaucoup en fonction des conditions de marche, rendant la sélection des poses clés plus compliquée. Par exemple, le maximum observé entre 20 et 40% du cycle dans un cas est remplacé par un plateau entre 20 et 40% du cycle dans un autre (Figure 20). Le comportement de la courbe varie également en phase oscillante où parfois un minimum est clairement visible alors que d'autres fois la trajectoire semble former un plateau ou diminuer pendant la majeure partie de la phase. Etant donnée la difficulté pour définir des poses clés liées à des caractéristiques de la trajectoire, seules les poses liées aux évènements sont utilisées pour la décrire (Figure 21). La prise en compte des variations des caractéristiques de la courbe est alors assurée par les poses complémentaires décrites dans la partie III.C.2.17.

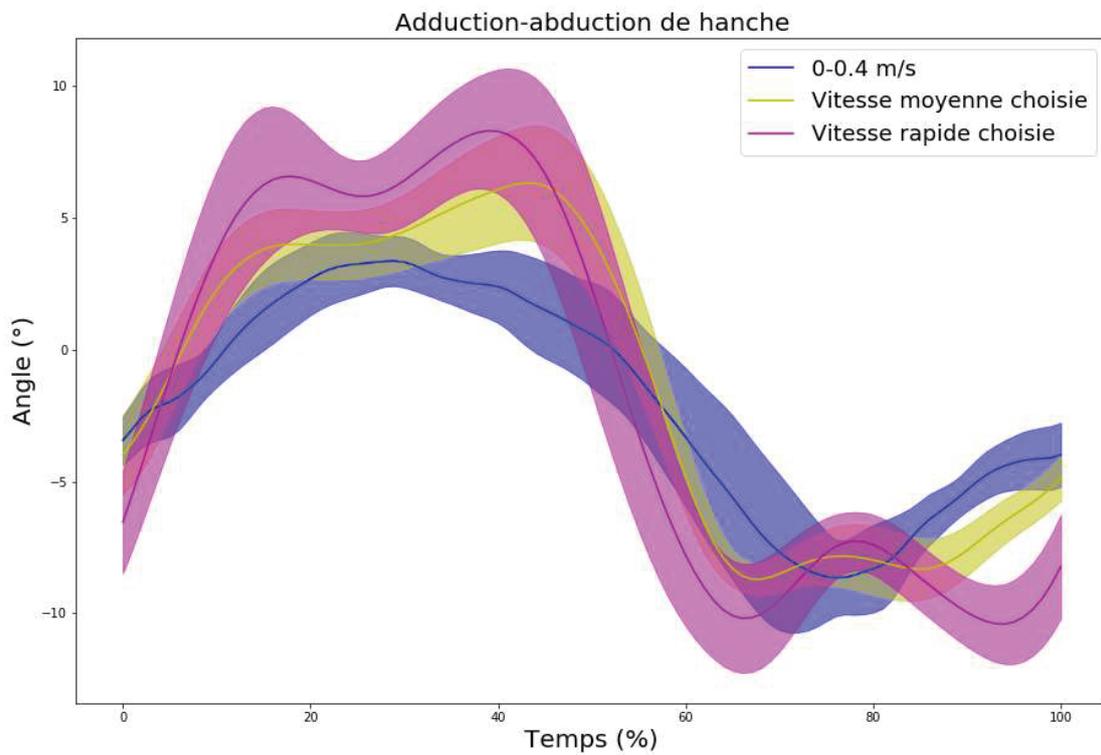


Figure 20 : Exemple de trajectoire d'adduction abduction de hanche avec et sans plateau en phase d'appui et en phase oscillante

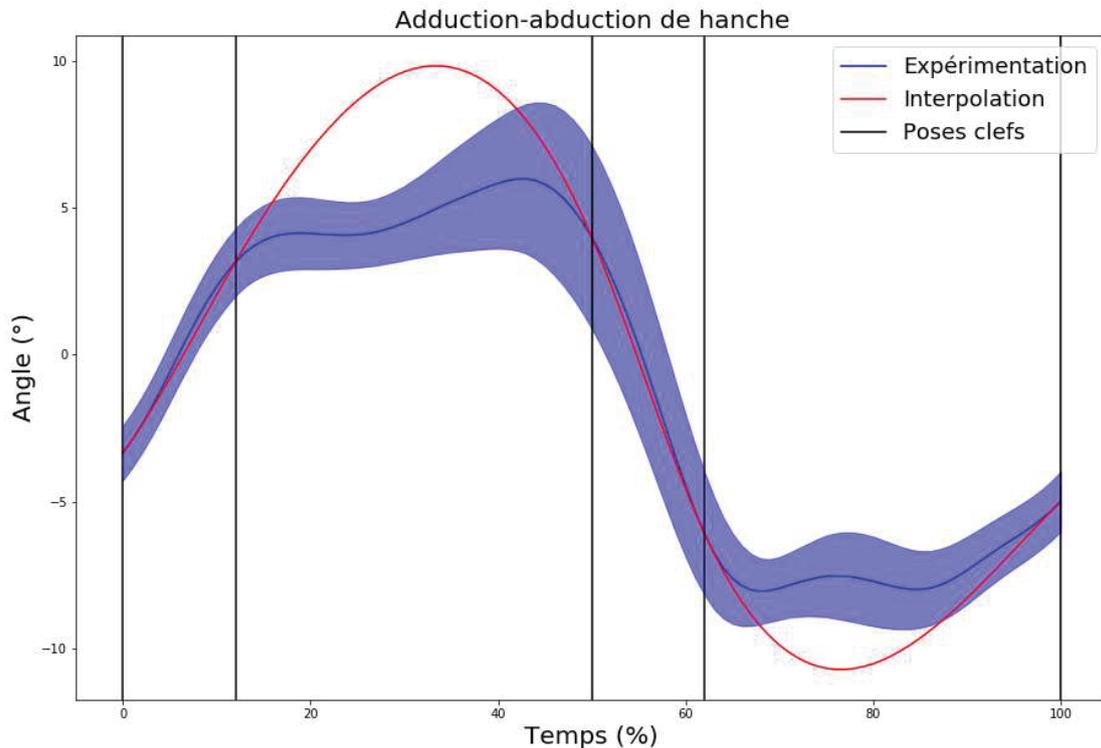


Figure 21 : Trajectoire expérimentale et interpolée, et poses clefs de l'adduction abduction de hanche

#### 2.4. Rotation interne-externe de la hanche

Les trajectoires de la rotation interne-externe de la hanche varient également beaucoup en fonction de la vitesse mais sont aussi beaucoup moins lisses que les autres trajectoires. En effet, la présence de maximums et minimums locaux est très irrégulière d'un essai à l'autre, rendant ici aussi la détermination de poses clefs difficile (Figure 22). Cependant, un point important à contrôler pendant la simulation est l'orientation du pied au moment de sa pose sur le sol. Cette orientation étant en grande partie contrôlée par la rotation de la hanche, une attention particulière doit être apportée à la fin de la trajectoire. Pour ce faire, une pose clef est définie comme étant l'angle minimal de la rotation après le CHS. Ce minimum étant positionné en fin de trajectoire, il permet de s'assurer que l'interpolation en fin de cycle est la plus biofidèle possible. En début de développement, le but était originellement d'utiliser le moins de poses clefs possibles. La pose de l'angle a donc remplacé la pose liée au CTO. L'interpolation de cette trajectoire est donc composée du HS1, du CTO, du CHS, de l'angle minimal après le CHS, et du HS2 (Figure 23).

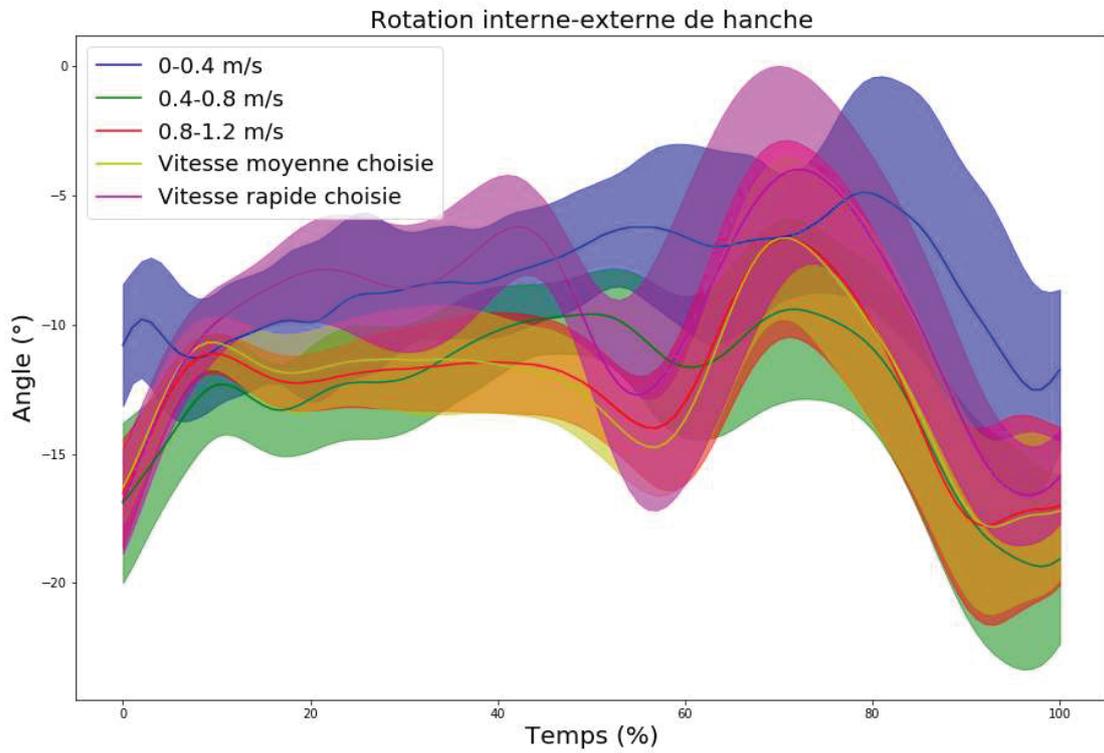


Figure 22 : Exemple de différents extremums de la rotation interne externe de hanche en fonction de la vitesse de marche

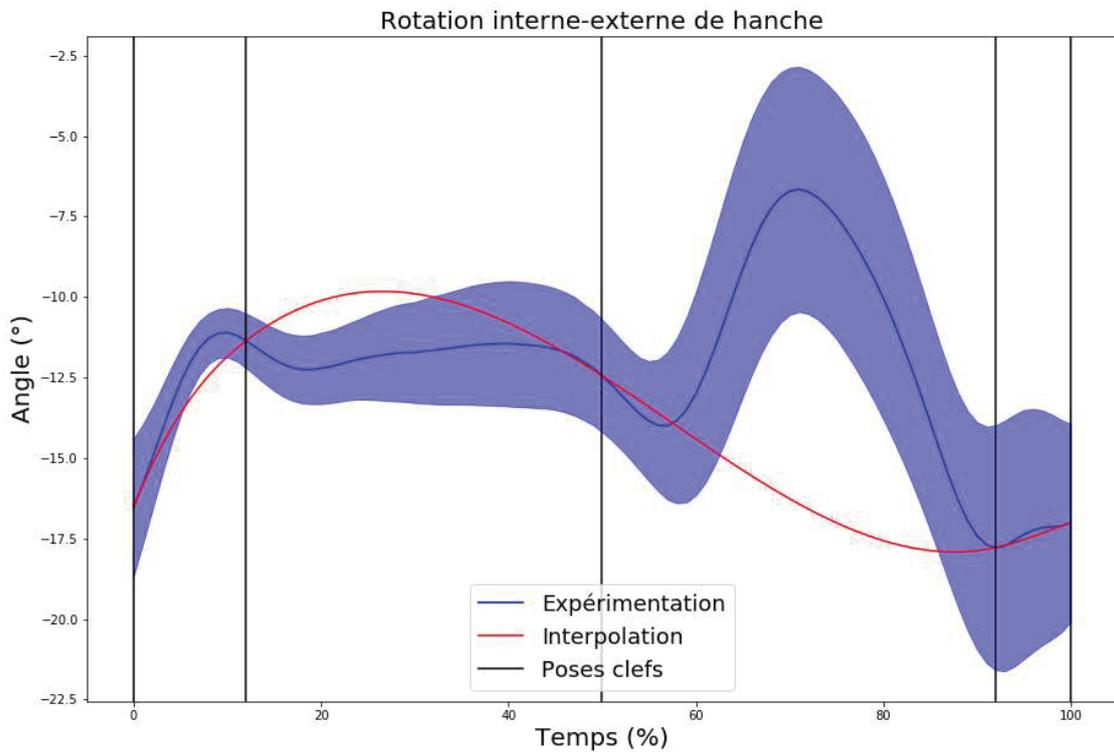


Figure 23 : Trajectoire expérimentale et interpolée, et poses clefs de la rotation interne externe de hanche

## 2.5. Flexion-extension du genou

Un premier point qui semble intéressant pour caractériser la trajectoire du genou est le premier minimum local situé entre 15 et 20% du cycle de marche. Malheureusement, la présence de ce minimum dépend de la vitesse de marche et il n'est pas toujours présent (Figure 24). Cependant, lorsqu'il est présent, l'évènement du CTO a tendance à en être proche. Cet évènement est donc conservé et permet de prendre en compte le minimum local lorsqu'il est présent. Les poses suivantes sont déterminées de manière similaire à celles de la flexion-extension de hanche. La trajectoire du genou présente un minimum global à environ 70% du cycle de marche. Le même raisonnement que pour la flexion-extension de hanche est alors appliqué. La trajectoire est composée de deux interpolations dont la fin de la première et le début de la seconde correspondent au minimum de la trajectoire. De plus, pour éviter que l'interpolation ne commence à viser le minimum trop tôt, la pose liée au CHS est conservée puisque la flexion-extension du genou amenant à ce minimum ne commence pas avant. Dans le cas où la trajectoire du genou commence par un plateau, celui-ci est alors défini par trois points, assurant sa présence. Dans le cas où la trajectoire commence par un minimum local, la présence de la pose au CHS assure alors que le premier minimum soit effectivement local. La première interpolation est donc composée du HS1, du CTO, du CHS et de l'angle minimal de la trajectoire.

Pour la deuxième interpolation, la situation est également similaire à la flexion-extension de hanche. Dans certains cas, un maximum local existe avant la fin du cycle alors que d'en d'autres, la pente varie rapidement mais sans s'inverser. La même solution est donc appliquée et une pose clef est définie au moment du minimum d'accélération entre l'angle minimal de la trajectoire et la fin du cycle. De plus, là aussi, la pose liée au TO est située dans une partie où la trajectoire est relativement constante et n'est donc pas prise en compte. La seconde interpolation est donc composée de l'angle minimal de la trajectoire, du minimum d'accélération après cela, et du HS2 (Figure 25).

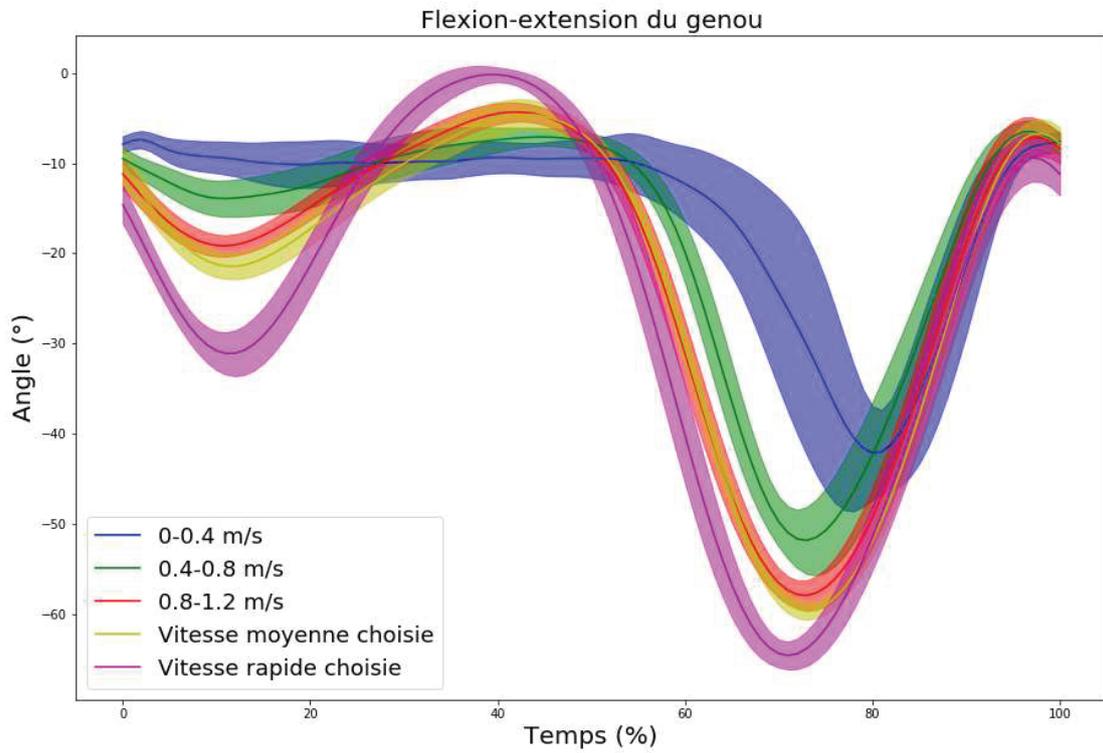


Figure 24 : Exemple de différences de trajectoires de flexion-extension du genou d'un sujet en fonction de la vitesse de marche

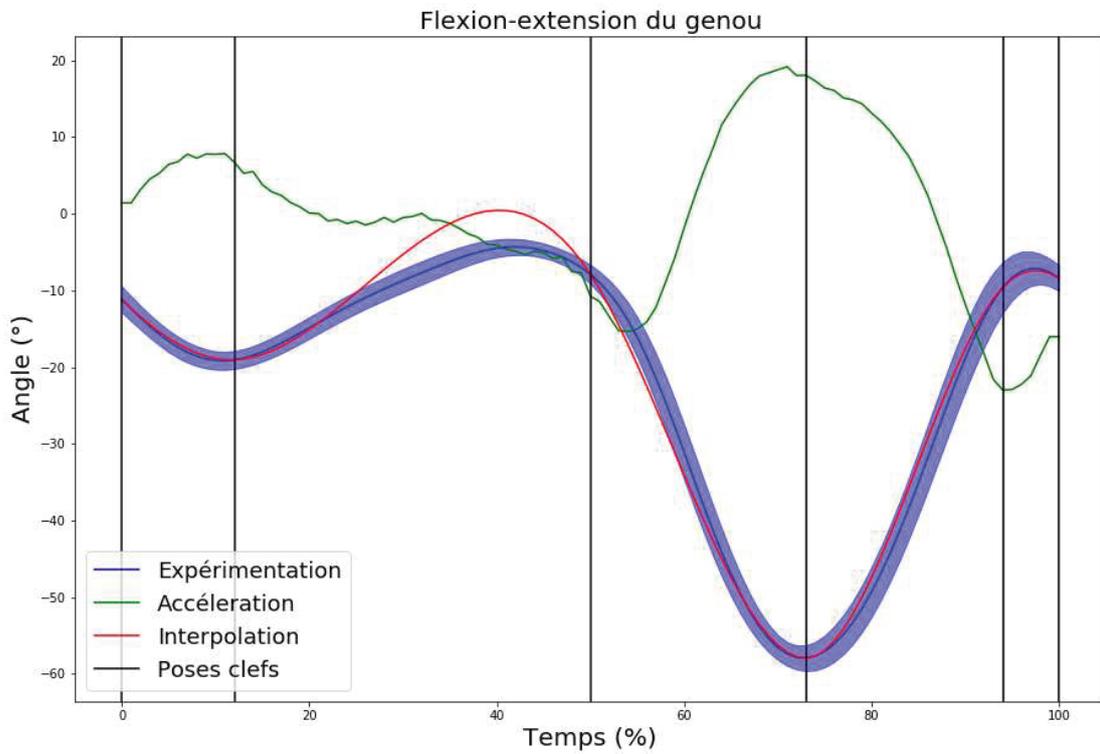


Figure 25 : Trajectoire expérimentale et interpolée, et poses clefs de la flexion extension du genou

## 2.6. Dorsi-planti flexion de cheville

La trajectoire de la cheville est constituée d'un enchaînement de minimums et maximums locaux, et ce, de manière régulière quelle que soit la vitesse (Figure 26). Les poses clés retenues sont alors le premier minimum, le maximum avant le second minimum, le second minimum et le second maximum. Pour assurer une meilleure concordance entre l'interpolation et les trajectoires expérimentales, il a été décidé de séparer cette trajectoire en deux interpolations afin de pouvoir contraindre la dérivée à l'un des extremums. Les deux extremums qu'il peut être intéressant de contraindre sont le premier maximum et le second minimum. En effet, la période entre ces deux poses correspond au moment où la cheville propulse le corps en fin de phase d'appui. Le suivi d'une trajectoire biofidèle étant plus important lors de la phase d'appui, le premier maximum a été choisi pour avoir sa dérivée contrainte. En effet, contraindre le second minimum aurait en partie amélioré la trajectoire de la cheville lors de la phase oscillante, ce qui est moins nécessaire. Ainsi, la première interpolation comprend le HS1, le premier minimum et le premier maximum tandis que la seconde interpolation comprend le premier maximum, le second minimum, le second maximum et le HS2 (Figure 27).

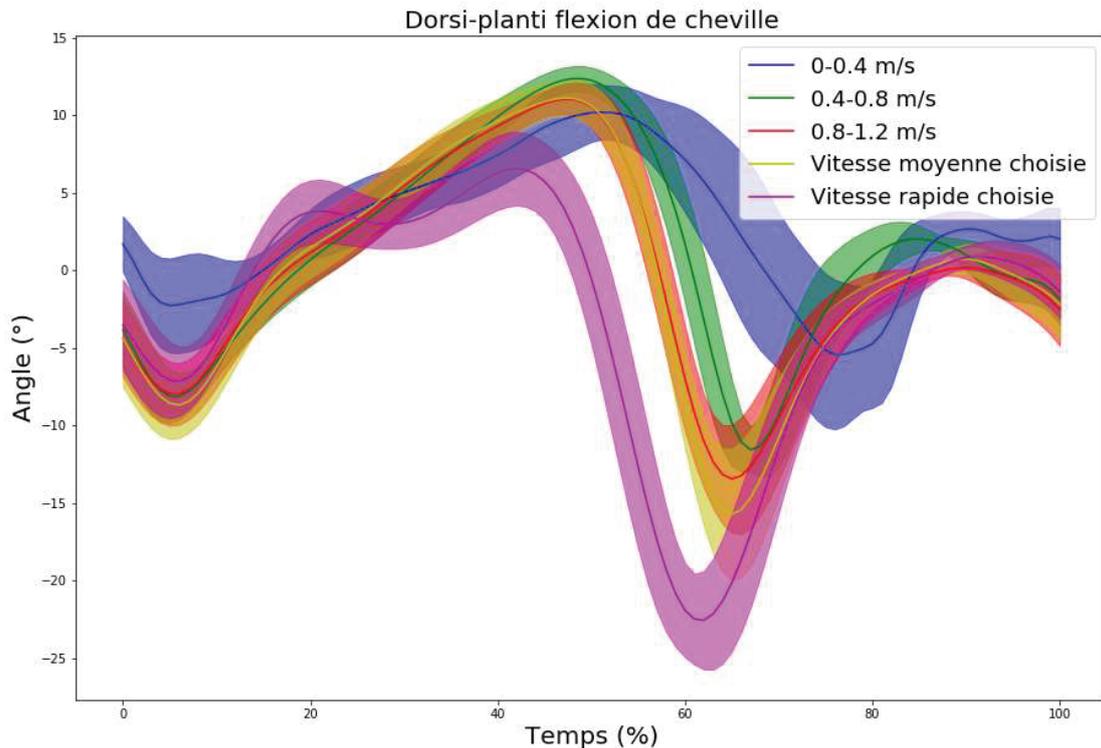


Figure 26 : Exemple de différences de trajectoires de dorsi-planti flexion de cheville d'un sujet en fonction de la vitesse de marche

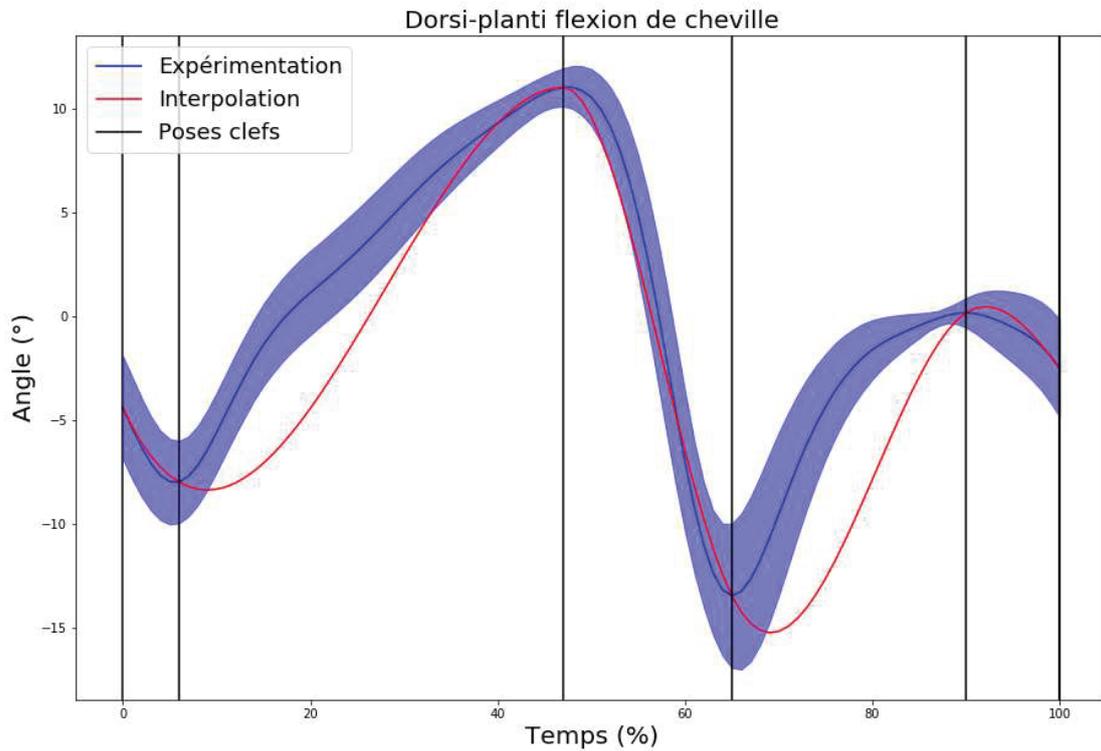


Figure 27 : Trajectoire expérimentale et interpolée, et poses clefs de la dorsi-planti flexion de cheville

## 2.7. Inversion-éversion de l'articulation subtalaire

La première caractéristique visible pour cette trajectoire est le maximum entre 50 et 60% du cycle de marche et le minimum local qui suit. Cependant, avant le maximum, la trajectoire varie selon la vitesse et le sujet (Figure 28). Avant la pente menant au maximum, la valeur de l'angle peut diminuer de manière constante, diminuer puis former un plateau ou même décrire un arc de cercle. Deux poses clefs sont utilisées pour décrire la trajectoire avant le maximum. La première est la pose liée au TO. En effet, dans les cas où la trajectoire décrit un plateau, le début du plateau semble coïncider avec le TO. La seconde pose fait office d'exception et est définie arbitrairement à 40% du cycle de marche. En effet, la multitude de formes que peut prendre la trajectoire en début de cycle rend compliquée la détermination de la position du début de la pente menant au maximum. En observant plusieurs essais de différents sujets à différentes vitesses, 40% semble une bonne approximation de ce moment et est donc choisi comme pose clef. Ainsi, l'interpolation comporte le HS1, le CTO, l'angle à 40% du cycle, l'angle maximal et le HS2 (Figure 29).

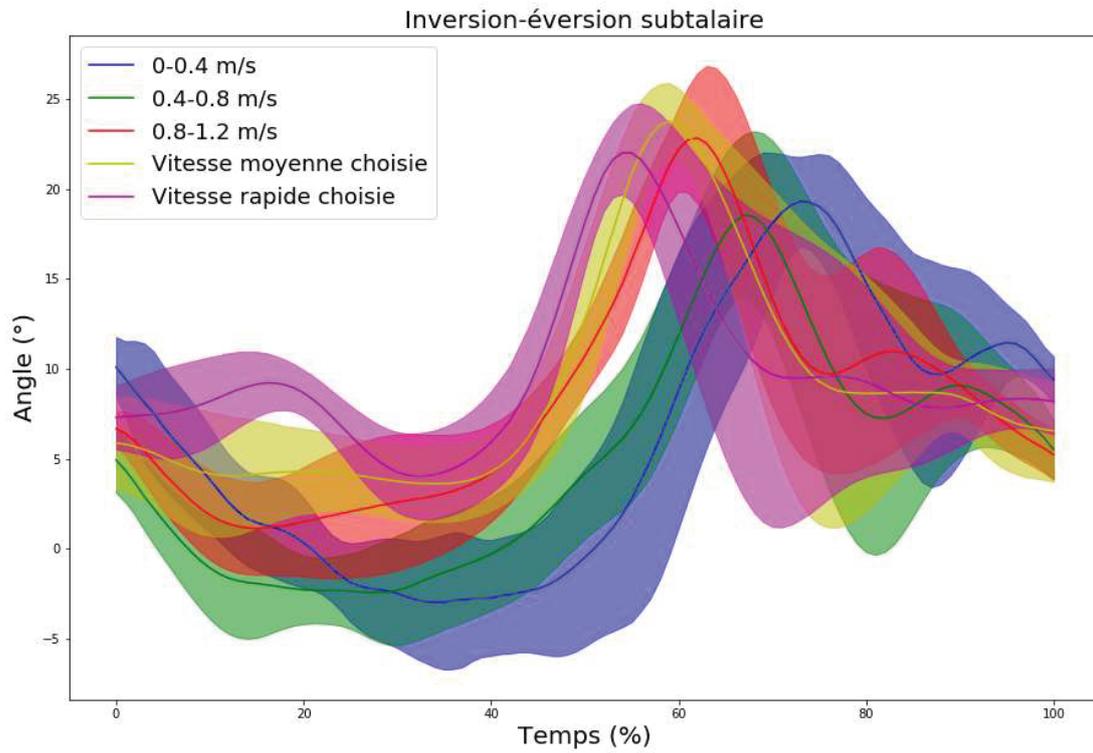


Figure 28 : Exemple de différences de trajectoires d'inversion éversion-subtalaire d'un sujet en fonction de la vitesse de marche

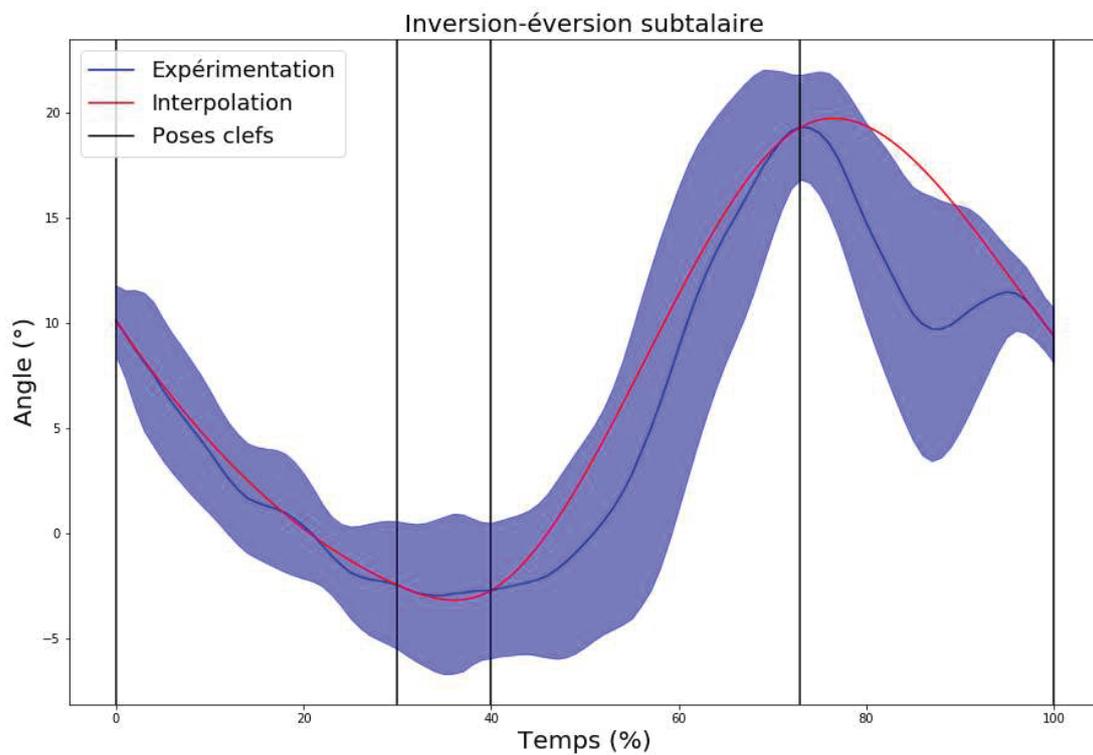


Figure 29 : Trajectoire expérimentale et interpolée, et poses clefs de l'inversion-éversion subtalaire

## 2.8. Flexion-extension de l'articulation métatarso-phalangienne

Dans ce projet, cette articulation est considérée comme un élément passif n'étant pas utilisé pour contrôler le modèle. Elle est donc modélisée par un ressort et un amortisseur de torsion suivant une valeur et une vitesse angulaire de référence nulle, permettant une meilleure modélisation du roulement du pied sur le sol que si les orteils n'étaient pas pris en compte dans le contact. De ce fait, sa trajectoire de référence est une droite constante de valeur nulle et ne fait donc pas l'objet d'une définition de poses clefs.

## 2.9. Flexion-extension de l'articulation lombaire

La plupart des conditions de vitesse permettent de distinguer des extremums locaux qui pourraient servir de poses clefs. Cependant, ces extremums ont tendance à être absents lors d'une marche à faible vitesse et ne sont donc pas utilisables (Figure 30). En revanche, les extremums, lorsqu'ils sont présents, semblent coïncider avec les événements. En effet, les événements CTO, CHS et TO se situent généralement respectivement entre 8 et 30%, vers 50%, et entre 58 et 80% du cycle en fonction de la vitesse. De plus, il est cohérent que des variations importantes de la trajectoire de flexion-extension lombaire soient causées par des changements d'appui. Les poses clefs liées aux événements sont donc gardées telles quelles sans ajout (Figure 31).

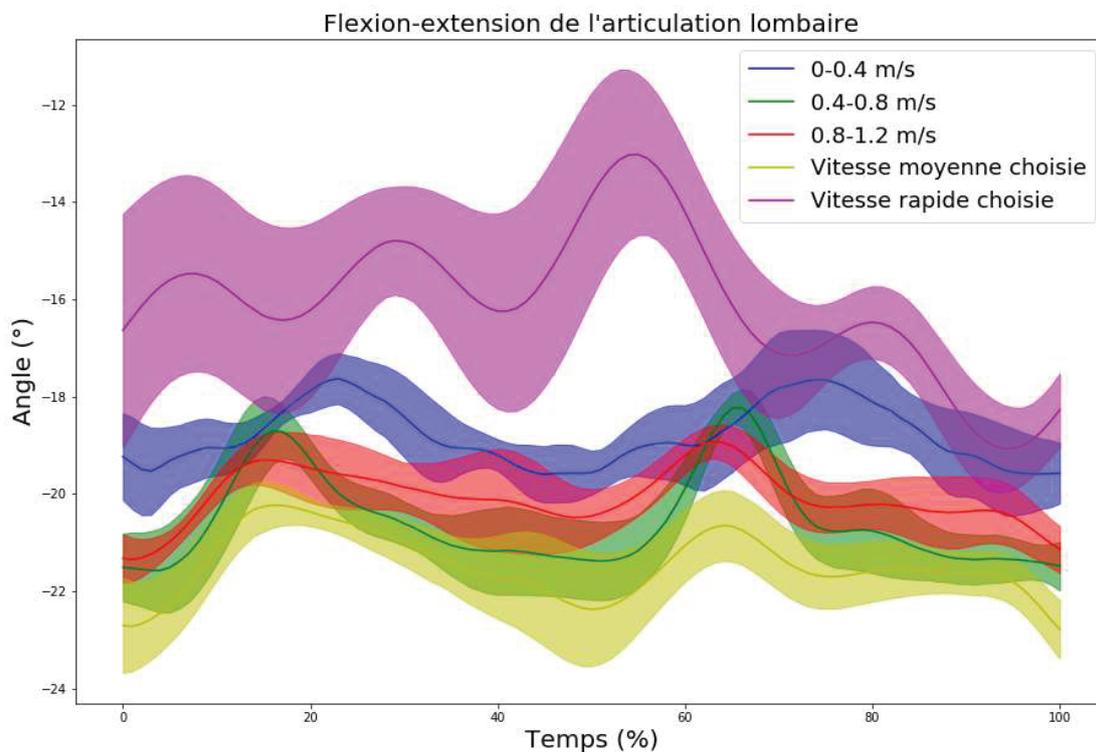


Figure 30 : Exemple de différences de trajectoires de flexion-extension lombaire d'un sujet en fonction de la vitesse de marche

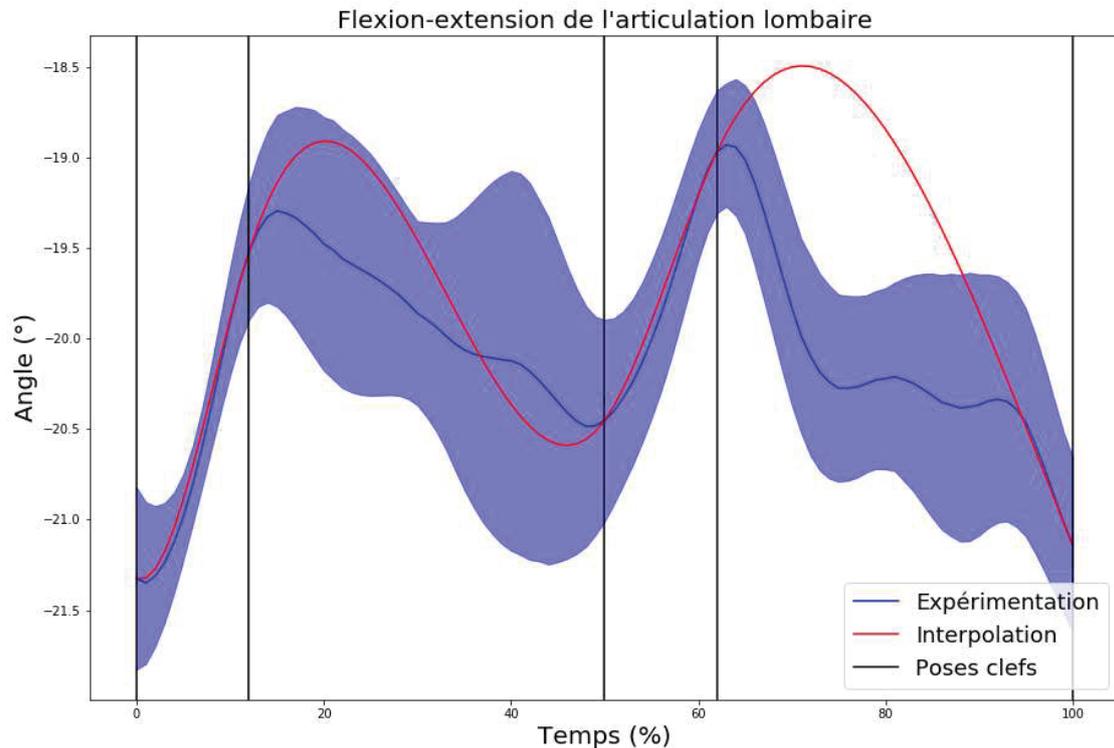


Figure 31 : Trajectoire expérimentale et interpolée, et poses clefs de la flexion-extension lombaire

## 2.10. Inclinaison latérale et rotation axiale droite-gauche de l'articulation lombaire

Lors de la simulation, ces deux DoF ne sont pas contrôlés en fonction d'une trajectoire de référence mais par rapport à l'orientation du pelvis, de sorte que le torse reste droit dans l'espace global. En effet, le pelvis peut voir son orientation varier relativement rapidement lors de la marche, notamment lors d'une pose du pied, alors que le tronc garde une orientation neutre. Pour garder cette orientation, les lombaires compensent les variations d'orientation du pelvis. Si cet effet est retranscrit dans les trajectoires expérimentales, l'approximation des interpolations risque, elle, de ne pas le faire, provoquant potentiellement des instabilités si le tronc est entraîné par les rotations du pelvis ou si le tronc empêche les rotations du pelvis. Contrôler l'orientation du torse directement en fonction de l'angle du pelvis permet donc d'éviter ces instabilités.

Les couples du pelvis sur le tronc permettent au tronc de rester droit malgré les variations d'inclinaison du pelvis. Cependant, l'action de ces mêmes couples agit également sur le pelvis de manière opposée et a donc tendance à intensifier ces variations. Les variations de l'orientation du pelvis dans le plan frontal et vertical sont modérées et ce phénomène n'est donc pas considéré comme gênant. En revanche, les variations dans le plan sagittal peuvent être plus importantes au point que ce phénomène puisse engendrer une inclinaison du pelvis trop importante pour être corrigée. De ce fait, la flexion-extension lombaire n'est pas contrôlée de cette façon.

### 2.11. Flexion-extension, adduction-abduction et rotation interne-externe de l'épaule et flexion-extension du coude

Les bras sont contrôlés dans la simulation pour prendre en compte leur action sur la dynamique du système mais ne sont pas directement utilisés pour contrôler l'équilibre du bipède. Leurs trajectoires n'ont pas besoin d'être aussi précises que celles des membres inférieurs et sont donc uniquement décrites par les poses liées aux évènements.

### 2.12. Position antéropostérieure du CoM

La trajectoire du CoM dans la direction antéropostérieure semble évoluer de manière constante sans présenter de caractéristique pouvant faire office de poses clefs. Une première approximation serait donc de définir sa trajectoire de référence par une fonction linéaire. Cependant, la trajectoire n'est pas parfaitement linéaire et ce genre d'approximation ferait donc perdre des informations sur les variations de vitesse au cours du cycle de marche. La vitesse de référence serait alors constante, ce qui ne correspond pas à la réalité. Pour garder ces informations, la trajectoire est décrite par les poses clefs liées aux évènements puisque là aussi, les changements d'appui ont une influence sur la trajectoire (Figure 32).

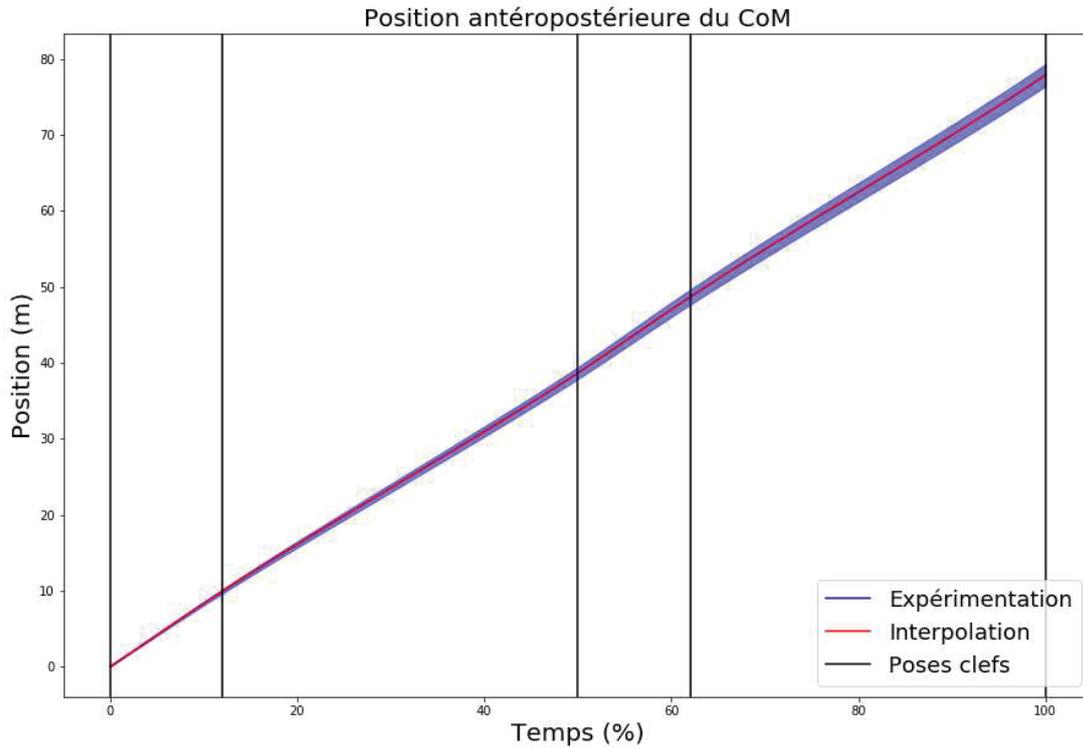


Figure 32 : Trajectoire expérimentale et interpolée, et poses clefs de la trajectoire antéropostérieure du CoM

### 2.13. Position médio-latérale du CoM

La trajectoire du CoM dans la direction médio-latérale est semblable à une sinusoïde et est approximée comme telle lors de la simulation. Pour garder une direction de marche constante, le CoM doit osciller de part et d'autre de l'axe antéropostérieur avec la même amplitude. Cependant, lors d'une expérimentation, cette égalité n'est jamais parfaitement respectée sur une seule foulée, ce qui peut influencer la valeur des poses clefs pour le modèle statistique. Directement approximer la trajectoire par une sinusoïde permet donc de ne pas prendre en compte ces imperfections dans le modèle statistique. Pour cette trajectoire, la seule donnée extraite de la base de données est l'amplitude de la trajectoire calculée comme étant la différence entre le maximum et le minimum de la trajectoire (Figure 33).

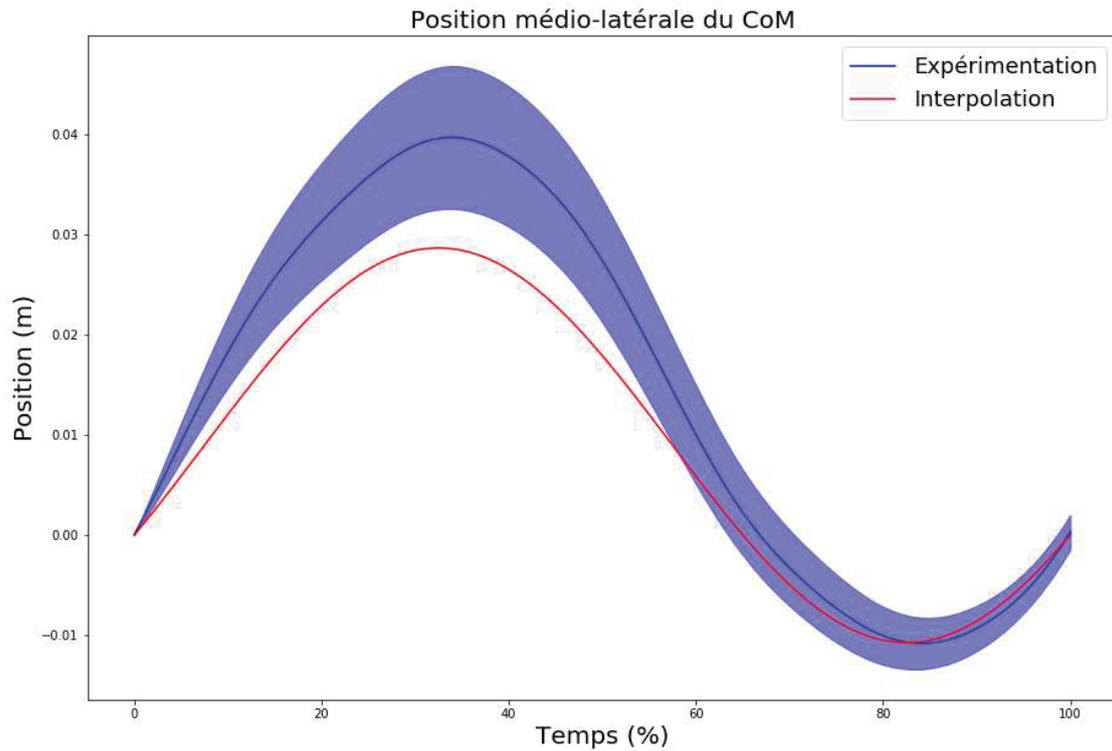


Figure 33 : Trajectoire expérimentale et interpolée, et poses clefs de la trajectoire médio-latérale du CoM

#### 2.14. Position antéropostérieure du talon en phase oscillante

La trajectoire antéropostérieure du talon en phase oscillante ne présente pas de point caractéristique pour la décrire mais possède tout de même une forme bien particulière et régulière quelle que soit la condition de marche. Il a donc été décidé pour décrire cette trajectoire de placer des poses clefs de manière régulière. Les poses sont à 0, 25, 50, 75 et 100% de la phase oscillante. Les poses à 0 et 100% permettent de donner la pente générale de la trajectoire tandis que les poses à 25, 50 et 75% permettent de prendre en compte ces inflexions (Figure 34).

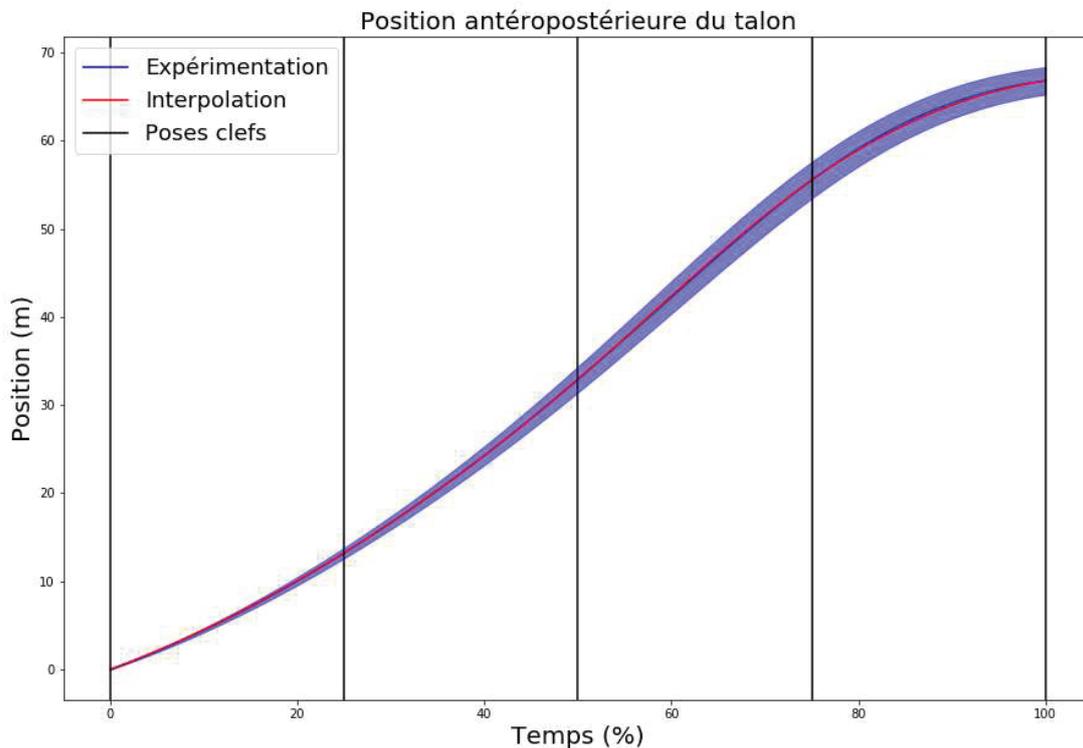


Figure 34 : Trajectoire expérimentale et interpolée, et poses clefs de la trajectoire antéropostérieure du talon en phase oscillante. Dans cette figure, 0% correspond au TO et 100% au HS2.

## 2.15. Position verticale du talon en phase oscillante

La trajectoire verticale du talon est elle aussi similaire quelle que soit la condition de marche mais présente des points caractéristiques utilisables pour déterminer des poses clefs. La première pose correspond au maximum de la courbe peu après le début de la phase oscillante. La seconde correspond au minimum local à environ 70% de la phase oscillante. Et la troisième correspond au maximum local à environ 90% de la phase. La prise en compte du minimum et du maximum local est importante pour éviter que la trajectoire ne décrive une cloche sur l'ensemble de la phase, ce qui ne correspondrait pas à une trajectoire naturelle du talon. Cette trajectoire peut toutefois être problématique lors de la simulation. En effet, la trajectoire au moment du minimum local fait passer le talon à quelques millimètres du sol. Cela n'est pas dérangeant pour un humain non pathologique puisque le contrôle du corps est suffisamment précis et robuste pour que le pied passe à cette hauteur sans toucher le sol et que, le cas échéant, le pied ne fasse que glisser sur le sol sans que les forces de réaction du sol ne soient suffisamment importantes pour perturber la marche. Cependant, dans le cas d'une simulation, le contrôle est moins précis et robuste et il est possible qu'une modification des caractéristiques soit une perturbation suffisante pour que le pied touche le sol à ce moment-là, même si la condition de base permettait au pied de passer sans toucher le sol. Dans ce projet, les événements étant détectés par la présence ou non des forces de réaction du sol, un contact du pied à ce moment ferait passer la

simulation dans la phase de marche suivante beaucoup trop tôt pour que le modèle reste stable. L'une des causes possibles est que la dérivée de l'interpolation ne soit pas nulle à la pose clef du minimum, ce qui implique que le minimum local est décalé en amont ou en aval de ce point, mais surtout, que sa valeur est inférieure à celle de la pose clef. Pour éviter cela, la trajectoire est décrite par deux interpolations dont la fin de la première et le début de la seconde correspondent au minimum local pour pouvoir assurer une dérivée nulle en ce point. La première interpolation est donc constituée de la valeur à 0% de la phase oscillante, du maximum et du minimum local. La seconde interpolation est constituée du minimum local, du maximum local et de la valeur à 100% de la phase oscillante (Figure 35).

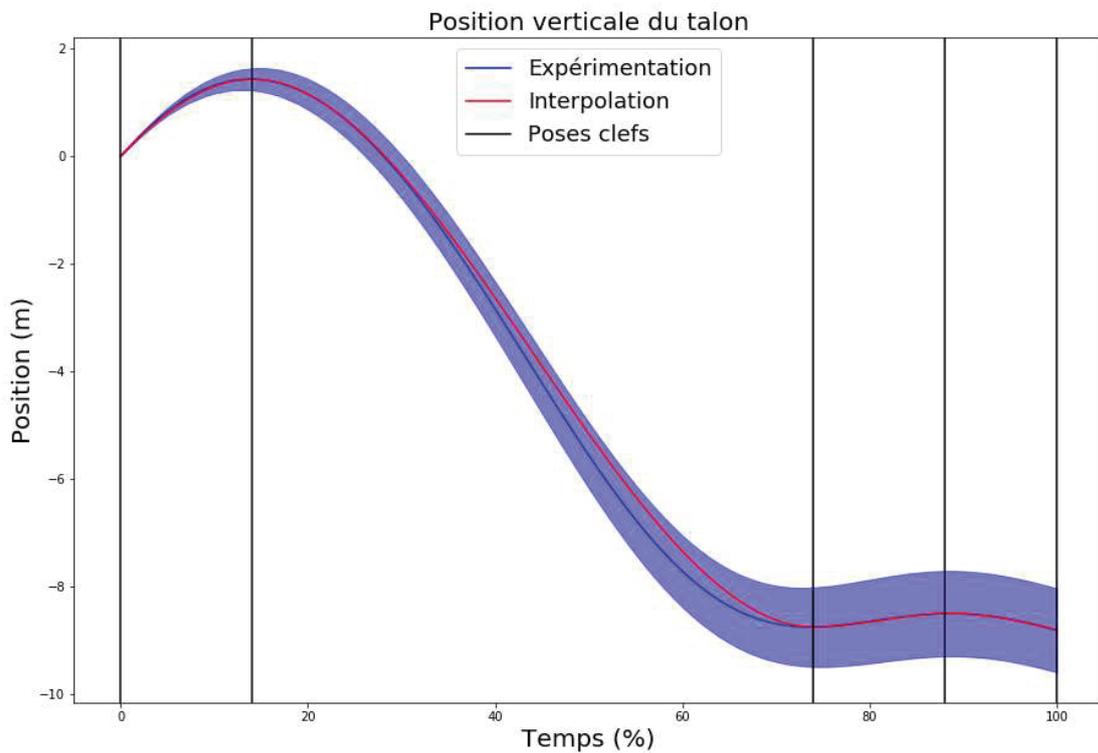


Figure 35 : Trajectoire expérimentale et interpolée, et poses clefs de la trajectoire verticale du talon en phase oscillante. Dans cette figure, 0% correspond au TO et 100% au HS2.

## 2.16. Position médio-latérale du talon

La trajectoire du talon peut présenter une différence au niveau du sens dans lequel elle évolue en début de phase oscillante selon la vitesse de marche. Cependant, la caractéristique la plus importante de la trajectoire est le sens dans lequel se dirige le talon en fin de phase puisque cela détermine la direction de la force de réaction du sol au moment du contact. L'information concernant la pente de la trajectoire étant utilisée lors de la définition des splines, il a été déterminé que la forme globale de la trajectoire pouvait être décrite par trois poses clefs à 0, 50 et 100% de la phase oscillante. Les trajectoires pour les talons droit et gauche étant symétriques, pour la détermination du modèle statistique, les trajectoires du talon gauche sont inversées pour correspondre à celles du talon droit. Lors

de la simulation, le modèle statistique fournit la trajectoire du talon droit, qui est alors inversée si le membre inférieur en phase oscillante est le membre inférieur gauche (Figure 36).

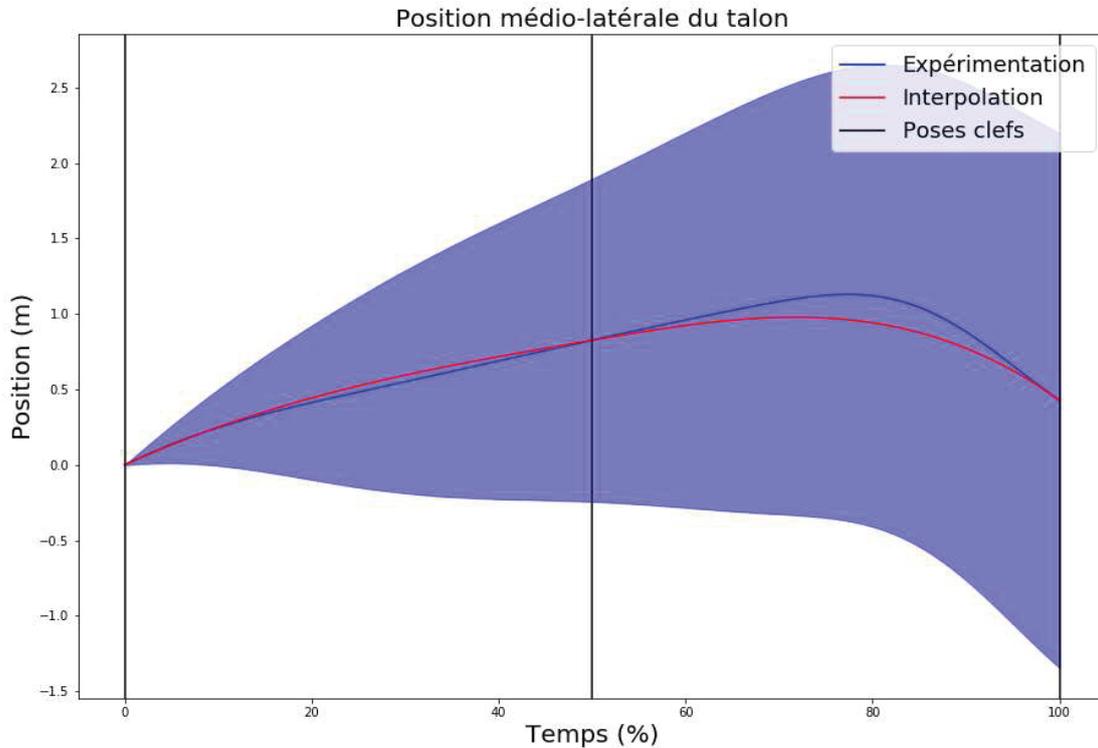


Figure 36 : Trajectoire expérimentale et interpolée, et poses clés de la trajectoire médio-latérale du talon en phase oscillante. . Dans cette figure, 0% correspond au TO et 100% au HS2.

## 2.17. Poses complémentaires

Pour les trajectoires des angles articulaires, en plus des poses clés principales, des poses clés supplémentaires sont définies à intervalles réguliers pour augmenter la précision de l'interpolation. Cela est notamment utile pour les trajectoires ayant des caractéristiques difficilement définissables telles que l'angle d'adduction-abduction de hanche. Plusieurs intervalles ont été étudiés pour déterminer le meilleur rapport entre la précision apportée et la régularité des trajectoires. En effet, plus l'intervalle entre les poses complémentaires est étroit, plus l'interpolation passe par des points décrivant la trajectoire et plus les caractéristiques de cette dernière sont respectées. Cependant, lors de la simulation, les poses clés sont déterminées par un modèle statistique, et sont donc des approximations

dont la valeur est indépendante de la valeur des poses adjacentes. Il n'est donc pas impossible que le modèle statistique détermine deux poses adjacentes temporellement plus proches, et avec des valeurs plus éloignées qu'elles ne devraient l'être. Cette configuration pourrait alors résulter en une variation importante de la pente de l'interpolation qui n'est pas présente dans une trajectoire naturelle (Figure 37).

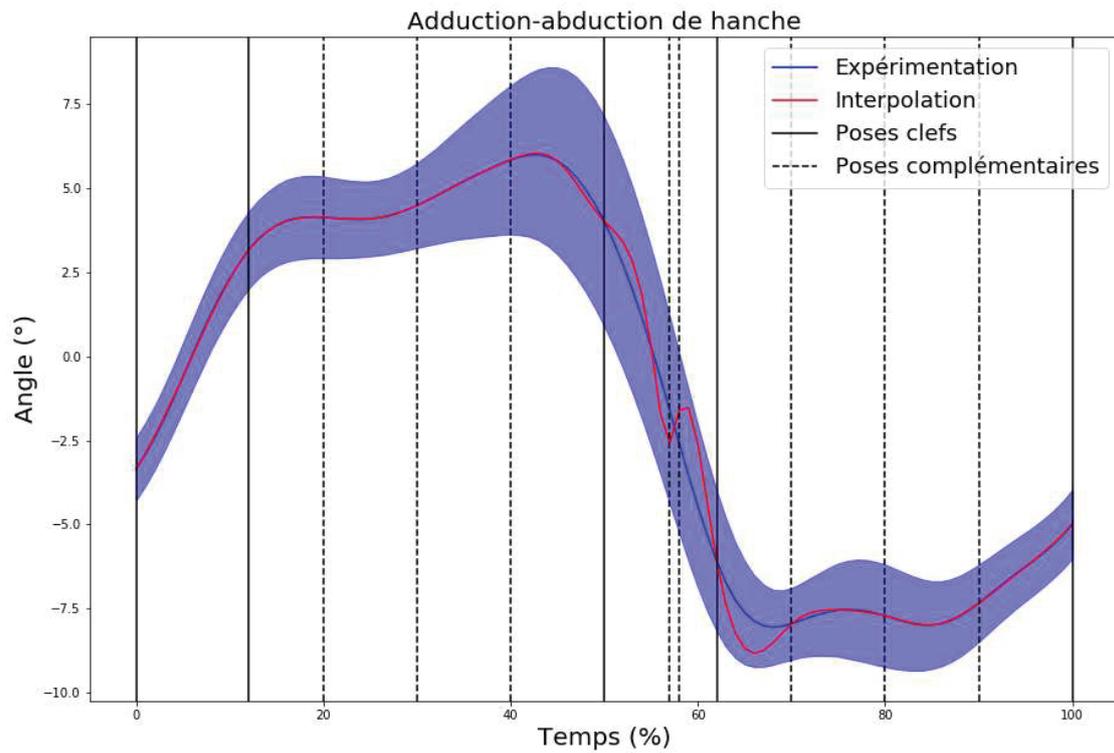


Figure 37: Risque à l'utilisation de poses clés trop proches

Trois intervalles ont été envisagés, plaçant les poses complémentaires tous les 5, 10 et 20% du cycle de marche. Lors des simulations, il s'est avéré que même un espacement de 5% ne provoquait pas l'effet décrit précédemment. C'est donc cet intervalle qui a été retenu (Figure 38).

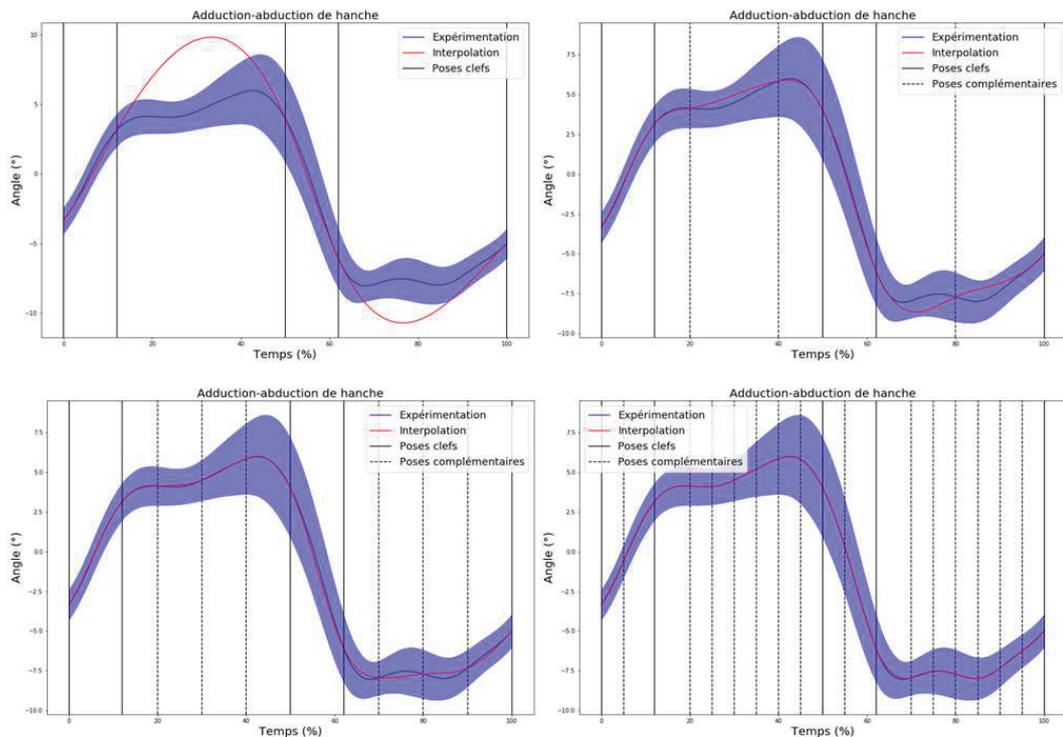


Figure 38 : Adduction-abduction de hanche avec et sans poses complémentaires

## 2.18. Force de réaction antéropostérieure

Nous verrons plus tard que les forces de réaction du sol sont nécessaires à l'optimisation des paramètres du contrôle et doivent donc être déterminées au même titre que les angles articulaires qui sont, eux, directement contrôlés.

La force de réaction antéropostérieure a une forme constante quels que soient le sujet ou la condition de marche. Deux points particuliers sont facilement localisables et permettent de décrire la courbe : le minimum et le maximum de la force. Ces deux extremums étant important, ils sont chacun définis comme point de liaison entre deux splines différentes pour pouvoir déterminer la pente de l'interpolation à ces endroits. En plus de ces points, la position à laquelle la force devient positive est également un point intéressant pour caractériser la forme de la courbe. Puisque pour les forces de réaction la dérivée de l'interpolation n'est pas utilisée, diviser l'interpolation en plus de deux splines ne pose pas de problème. Ainsi, pour encore plus de précision, le point où la force devient positive est également défini comme point d'intersection entre deux splines pour contrôler la pente de la courbe à cet endroit. La force antéropostérieure est donc interpolée via la succession de quatre splines. La bibliothèque permettant de créer les splines ayant besoin de trois poses par spline, une pose complémentaire est créée à mi-distance entre chaque pose clef déterminée précédemment et sont nommées q1, q2, q3 et q4. La première interpolation est décrite par la valeur à 0% du cycle, q1 et le

minimum. La deuxième interpolation est décrite par le minimum,  $q_2$  et la valeur nulle. La troisième interpolation est décrite par la valeur nulle,  $q_3$  et le maximum. Et enfin, la quatrième interpolation est décrite par le maximum,  $q_4$  et la valeur au lever du pied (Figure 39).

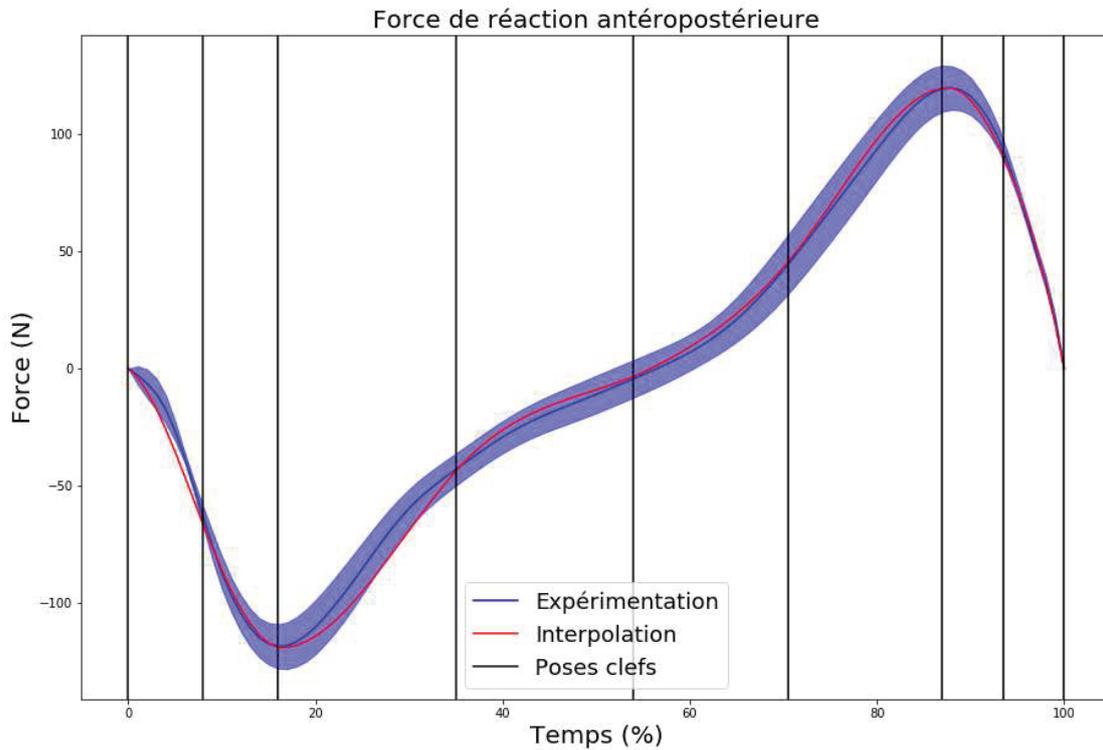


Figure 39 : Trajectoire expérimentale et interpolée, et poses clefs de la trajectoire de la force de réaction antéropostérieure

## 2.19. Force de réaction verticale

La force de réaction verticale a également une forme constante quel que soit le sujet ou la condition de marche. Deux points caractéristiques permettent de caractériser la courbe : les deux maximums locaux correspondant respectivement au levé et à la pose du pied controlatéral. Ces deux points étant des extremums, ils sont définis comme point d'intersection entre deux splines pour pouvoir déterminer leur pente. Ainsi, la courbe est déterminée par trois splines successives. Comme pour la force antéropostérieure, des poses clefs complémentaires ( $q_1$ ,  $q_2$ ,  $q_3$ ) sont définies à mi-distance des poses clefs précédentes pour compléter la détermination des splines, celle entre les deux maximums permettant, de plus, de décrire le minimum local. La première interpolation est décrite par la valeur à 0% du cycle,  $q_1$  et le premier maximum. La deuxième interpolation est décrite par le premier maximum,  $q_2$  et le second maximum. Enfin, la troisième interpolation est décrite par le second maximum,  $q_3$  et la valeur au lever du pied (Figure 40).

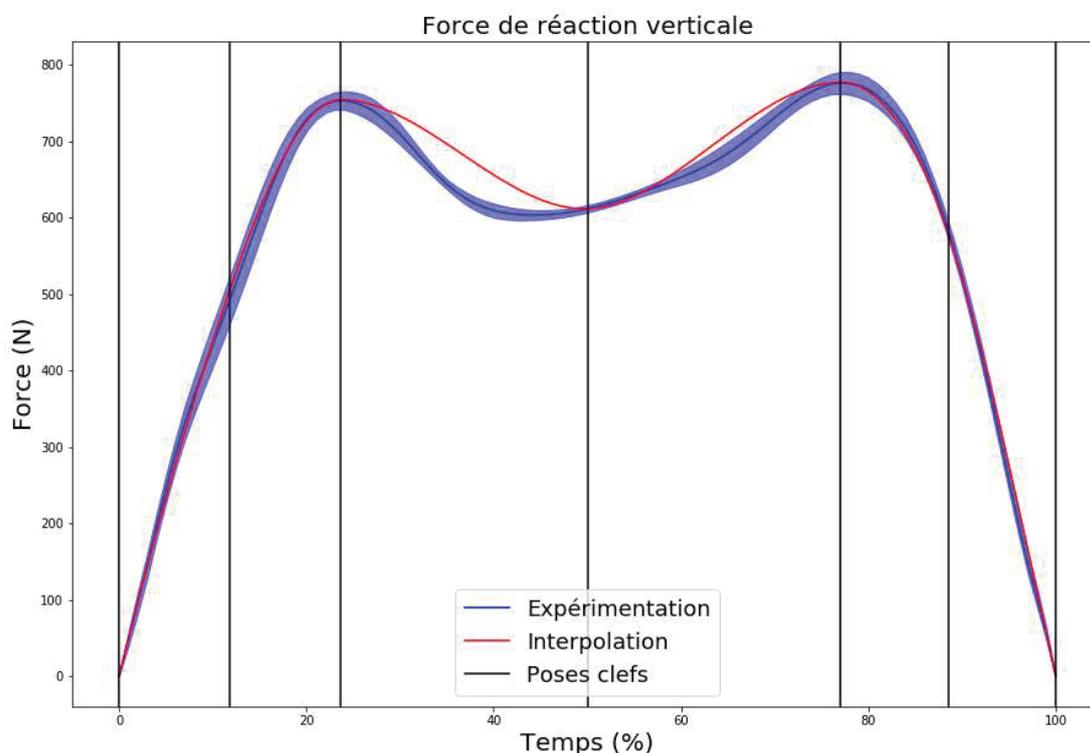


Figure 40 : Trajectoire expérimentale et interpolée, et poses clefs de la trajectoire de la force de réaction verticale

## 2.20. Force de réaction médio-latérale

La force de réaction médio-latérale a également une forme constante quels que soient le sujet ou la condition de marche, à part pour la pente de la courbe en fin de phase d'appui. Les trois points qui semblent intéressants pour décrire la courbe sont le minimum et les deux maximums locaux. Seulement, le minimum et le premier maximum local sont temporellement proches et les considérer tous les deux comme point d'intersection entre deux splines multiplierait le nombre de pose clefs sur une durée réduite. Ainsi, ces trois points sont définis comme des poses clefs, mais seuls les deux maximums sont également définis comme point d'intersection entre deux splines. Le minimum n'étant pas défini comme un point d'intersection, la pente de la courbe à ce moment n'est pas directement contrainte alors qu'il s'agit d'une partie importante de la courbe. Une pose intermédiaire  $q_1$  est alors définie à mi-distance entre le minimum et le premier maximum. Contrairement aux autres poses intermédiaires, elle n'est pas indispensable à l'interpolation de la spline mais est utilisée pour contraindre la forme de la courbe. En revanche, deux poses intermédiaires ( $q_2$ ,  $q_3$ ) entre les autres poses clefs entre le second maximum et le lever du pied sont bien définies pour permettre l'interpolation des splines. La première interpolation est donc décrite par la valeur à 0% du cycle, le minimum local,  $q_1$  et le premier maximum local. La deuxième interpolation est décrite par le premier maximum local,  $q_2$ , et le second maximum local. Enfin, la

troisième interpolation est décrite par le second maximum local,  $q_3$ , et la valeur au lever du pied (Figure 41).

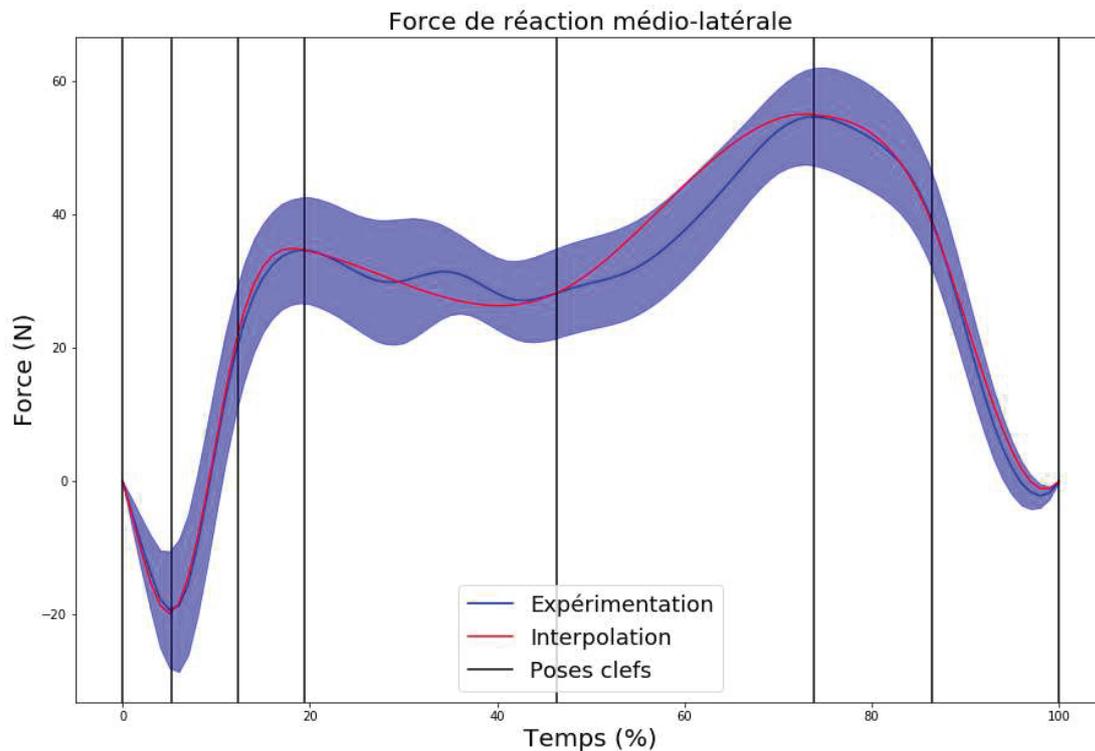


Figure 41 : Trajectoire expérimentale et interpolée, et poses clés de la trajectoire de la force de réaction médio-latérale

### 3. Modèle statistique

L'objectif de l'utilisation d'un modèle statistique pour déterminer les trajectoires de référence est de faire en sorte que celles-ci soient elles-mêmes adaptées à la morphologie de la personne simulée et à la vitesse de marche, afin de diminuer le besoin d'adaptation lors de la simulation.

Une fois toutes les poses clés déterminées, un code C++ compile la position des événements, la valeur des paramètres spatio-temporels, la position et la valeur des poses clés et les caractéristiques de tous les essais dans un même fichier texte pour servir à la détermination du modèle statistique. Les caractéristiques des essais sont la vitesse de marche, la longueur des membres inférieurs, l'IMC, l'âge et le sexe des sujets (Moissenet et al., 2019). La vitesse de marche étant influencée par la longueur des membres inférieurs, la vitesse utilisée dans le modèle statistique est adimensionnée par rapport à la longueur des membres inférieurs  $L_{membre}$  et à la valeur de l'accélération de la pesanteur  $g$  :

$$V_{adim} = \frac{V_{marche}}{\sqrt{L_{membre} * g}}$$

Le modèle statistique est alors déterminé comme un modèle de régression linéaire. Le principe de ce modèle est de prédire la valeur d'une variable dépendante en fonction d'une ou plusieurs variables indépendantes en déterminant un coefficient à appliquer à chaque variable indépendante. Pour cette application, les variables dépendantes sont les évènements (HS1, CTO, CHS, TO, HS2) et les poses clés, tandis que les variables indépendantes sont les caractéristiques des essais citées précédemment. Les coefficients sont alors calculés via un script python utilisant la fonction *linear\_model.LinearRegression.fit* de la bibliothèque *sklearn*, puis stockés dans un fichier texte utilisable par le simulateur.

$$Pose = Coef_0 + Coef_1 * V_{adim} + Coef_2 * L_{membre} + Coef_3 * IMC + Coef_4 * age + Coef_5 * sexe$$

Par la suite, un second modèle statistique est utilisé pour déterminer la longueur de la foulée. En effet, puisqu'il existe un lien entre la vitesse de marche, la longueur de la foulée et la durée de la foulée, il est commun, une fois la vitesse de marche cible décidée, de fixer une valeur initiale de la longueur ou la durée de la foulée pour ensuite déterminer l'autre. Le choix d'utiliser un modèle statistique différent vient du fait que ces valeurs ne sont liées qu'à la vitesse de marche et à la longueur des membres inférieurs. En prenant en compte uniquement ces paramètres, il est alors possible de définir une relation plus précise entre ceux-ci et l'une des deux variables que l'on souhaite déterminer. La meilleure relation ayant été déterminée a pour variable indépendante :

$$x = V_{marche}$$

et pour variable dépendante :

$$y = \frac{L_{foulée}}{L_{membre}}$$

La relation est définie par une relation polynomiale d'ordre 5 en utilisant la fonction *polyfit* de la bibliothèque python *numpy* (Figure 42) et se définit par :

$$y = \sum_{i=0}^5 coef_i * x^i \quad (14)$$

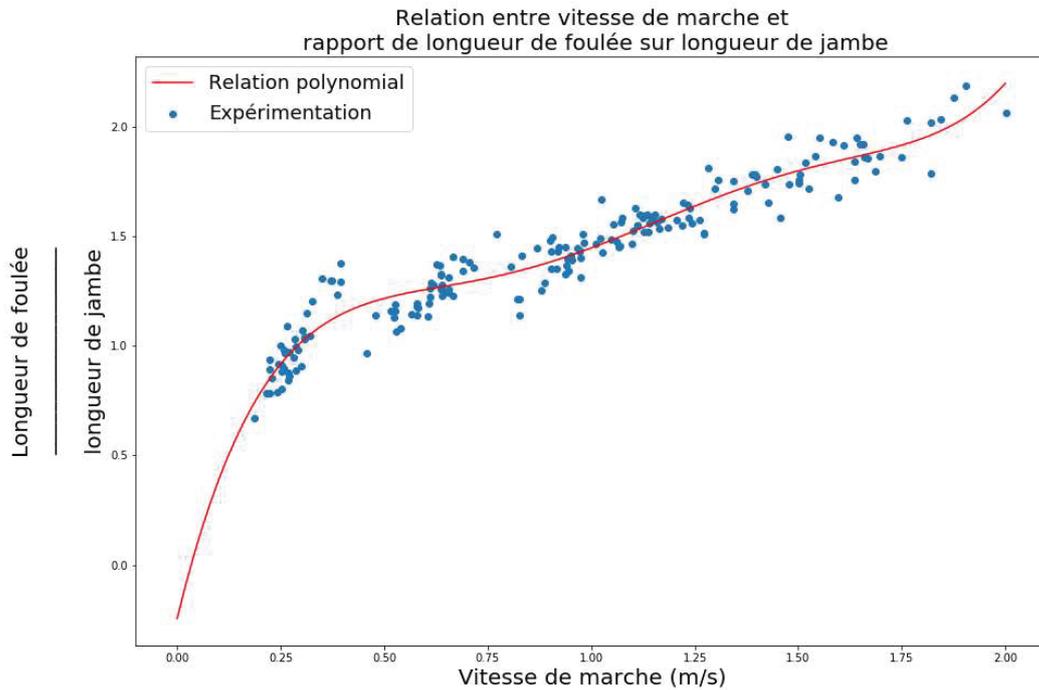


Figure 42 : Comparaison entre la relation polynomiale et les valeurs expérimentales

## D. Contrôle

Le contrôleur est développé dans le langage C++ car il s'agit du langage dans lequel sont écrites les bibliothèques de fonctions OpenSim à utiliser. Les différents éléments du contrôleur sont répartis dans différents fichiers. Les différentes parties du contrôleur font l'objet d'une classe à part entière, localisée dans un fichier dédié afin de simplifier l'organisation du code.

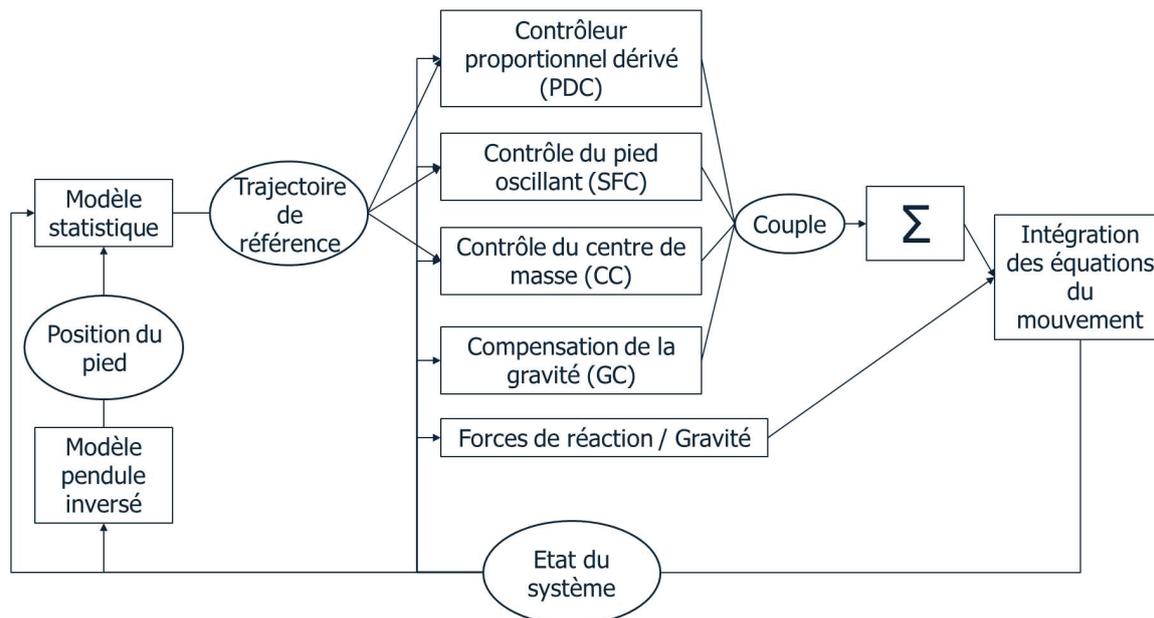


Figure 43 : Construction du contrôleur

Dans ce contrôle, un principe dérivé du modèle de pendule inversé est utilisé pour déterminer la position d'arrivée médio-latéral du pied en fin de phase oscillante. Un modèle statistique est alors utilisé pour déterminer un ensemble de trajectoire de référence. Quatre briques de contrôle sont alors appelées pour déterminer les couples articulaires à appliquer en fonction de l'état du système et de l'état de référence déterminé par le modèle statistique. La première brique est un PDC qui a pour rôle de suivre les trajectoires articulaires de référence. La seconde brique contrôle la position et la vitesse du talon en phase oscillante via l'application de forces virtuelles sur le talon. La troisième brique contrôle la position antépostérieure et médio-latéral du CoM et la vitesse de marche moyenne, également via l'application de forces virtuelles au niveau du CoM. Enfin, la quatrième brique a pour rôle de compenser la force de gravité sur les segments non porteurs afin de faciliter le contrôle de ces segments par les autres briques de contrôle. Les couples issus de ces briques sont alors additionnés pour déterminer le couple total à appliquer à chaque DoF, les couples totaux pouvant eux-mêmes être adaptés lors de changements de phases afin d'éviter toutes discontinuités. Les couples articulaires ainsi que les forces de réaction du sol et de gravité sont ensuite utilisés pour intégrer les équations du mouvement et déterminer le nouvel état du système, qui sera alors utilisé comme retour d'information à utiliser dans le modèle statistique et par les briques de contrôles.

## 1. Fichier principal

Le fichier principal contenant le point d'entrée du programme, la fonction *main*, est le premier fichier lu par l'ordinateur lors de la simulation. Il est utilisé pour décrire les caractéristiques de la simulation, créer les différentes instances de classe du contrôleur en fonction des conditions, et pour lancer la simulation.

### 1.1. Caractéristiques du modèle

La première étape consiste à préciser les caractéristiques du modèle pour la détermination des trajectoires de référence. Le modèle de base pour développer le simulateur a comme caractéristiques une vitesse de marche de 1 m/s, une longueur membre inférieur  $L_{membre}$  de 0.9139 m, un IMC de 21.75, un âge de 25 ans et est de sexe masculin. Cela correspond aux caractéristiques du modèle OpenSim de base décrit précédemment.

Il est ensuite possible de spécifier, en mètre, la longueur des différents segments du modèle pour faire varier la morphologie. Ces valeurs sont alors utilisées comme paramètres pour instancier une classe nommée *ScaleSetDefinition* écrite pour ce projet. Le fichier principal appelle alors une fonction de cette classe qui calcule pour chaque segment un coefficient de mise à l'échelle en fonction de la longueur indiquée et de la longueur de base :

$$Coef = \frac{\text{Longueur indiquée}}{\text{Longueur de base}}$$

Pour mettre le modèle à l'échelle, OpenSim utilise une fonction utilisant comme paramètre un fichier *xml* dans lequel sont indiqués les différents coefficients à appliquer. La fonction appelée crée donc un fichier *xml* avec les coefficients calculés. Les coefficients étant également nécessaires dans d'autres parties du simulateur, une seconde fonction de la classe est appelée pour remplir un tableau, défini dans le fichier principal, par les valeurs des coefficients. Ce tableau peut alors être utilisé comme paramètre d'une classe ou d'une fonction si nécessaire.

### 1.2. Importation et finalisation du modèle

La deuxième étape consiste à importer le modèle de base. Comme expliqué précédemment, le modèle est défini par une classe nommée *Model*, et qu'il est possible d'instancier directement avec un modèle existant en lui indiquant le chemin du fichier *osim* du modèle correspondant. De plus, c'est également via cette classe que la gravité est prise en compte, et il faut donc définir la valeur de celle-ci pour qu'elle soit appliquée correctement durant la simulation.

Une fois le modèle importé, il est nécessaire de le compléter avec le modèle de contact. En effet, la taille et la position des sphères de contact sur les pieds ne sont pas prises en compte lors de la mise à l'échelle du modèle, et il faut donc directement les définir en prenant en compte les coefficients de mise à l'échelle des segments des pieds. Le semi-volume de contact du sol est représenté par un objet de la classe *ContactHalfSpace* prenant comme paramètres sa position, son orientation, le segment auquel il est attaché et son nom. Les sphères de contact sont quant à elles représentées par des objets définis par la classe *ContactSphere* prenant comme paramètres le rayon et la position du centre de la sphère, le segment auquel elle est attachée, et son nom. C'est notamment lors de l'instanciation des sphères que le coefficient de mise à l'échelle est appliqué aux valeurs de rayon et de position des sphères. Les forces de contact sont représentées par des objets définis par la classe *HuntCrossleyForce* prenant comme paramètres la raideur, l'amortissement, le nom de la force et les géométries de contact à partir desquels la force est calculée. Cette classe a également pour données membres la friction statique, dynamique et visqueuse, ainsi que la valeur de la vitesse de transition pour calculer les forces de friction, mais puisque ces dernières sont calculées via un autre modèle, ces données ne sont pas définies. Pour des raisons de clarté du code, les actionneurs liés aux éléments viscoélastiques sont définis à ce moment, même si leur position est, elle, prise en compte par la mise à l'échelle du modèle. Les forces correspondantes à ces éléments viscoélastiques sont représentées par des objets définis par la classe *PointActuator* prenant comme paramètres le segment sur lequel est appliqué la force, le nom de la force, la position du point, l'information précisant si le point est exprimé dans le repère du segment ou dans le repère global, la direction de la force et l'information précisant si la direction est exprimée dans le repère du segment ou dans le repère global. Enfin, les éléments viscoélastiques représentant un ressort/amortisseur de torsion sont représentés par le même type d'actionneurs que les couples articulaires.

### 1.3. Tableau d'actionneurs

A plusieurs instants de la simulation, le contrôleur a besoin d'accéder aux objets définissant les forces, que ce soit pour obtenir la valeur calculée pour les forces de Hunt-Crossley ou pour appliquer la valeur calculée pour les autres. Afin de faciliter l'accès aux actionneurs, quatre tableaux, un pour les 25 couples articulaires, un pour les 16 forces de Hunt-Crossley, un pour les huit forces de frictions, et un pour les quatre couples de friction, sont définis et remplis par des pointeurs vers les objets les définissant pour être ensuite utilisés comme paramètres à passer à la classe de contrôle.

### 1.4. Calcul des poses clefs

L'étape suivante consiste à utiliser le modèle statistique pour déterminer les paramètres spatio-temporels et les poses clefs des trajectoires de référence. Cette étape est réalisée grâce une classe écrite pour ce projet nommée *StatisticalModel* prenant comme paramètres les caractéristiques du modèle précédemment définies.

Elle a comme données membres les paramètres de la simulation, directement initialisés lors de l'instanciation de la classe, 26 nombres entiers permettant de stocker le nombre de poses clefs principales par DoF, nécessaires lors de la création des *vector* contenant les poses clefs, ainsi que 28 *vector* permettant de stocker les poses clefs et les paramètres spatio-temporels qui seront transmis au contrôleur.

La classe possède ensuite une première fonction membre permettant de calculer les valeurs à déterminer par le modèle statistique.

Enfin, 28 fonctions sont définies pour retourner les *vector*, tableaux et valeurs réelles dans lesquels sont stockées les valeurs déterminées par le modèle statistique.

Cette classe est instanciée en un objet nommé *statisticalModel*. La première fonction est alors directement appelée pour calculer les valeurs nécessaires.

Tout d'abord, connaissant la longueur des membres inférieurs puisqu'elle fait partie des caractéristiques du modèle, la valeur de la vitesse est adimensionnée de la même manière que lors de la détermination du modèle statistique.

Ensuite, la fonction accède au fichier texte contenant tous les coefficients, en extrait les valeurs, et les stocke dans un tableau 2D, *coefficients*, où chaque ligne représente une valeur à déterminer et chaque colonne est un des coefficients. Les valeurs à déterminer sont alors calculées grâce aux coefficients puis stockées dans un tableau.

Le second modèle statistique est alors utilisé pour déterminer la longueur de pas. Le ratio de la longueur de la foulée par rapport à la longueur de membre inférieur est calculé via le modèle puis multiplié par la longueur du membre inférieur pour obtenir la longueur de la foulée. Elle est tout d'abord utilisée pour déterminer la durée de la foulée, stockée dans une donnée nommée *strideDuration*, puis est divisée par deux pour obtenir la longueur de pas. La durée de la foulée est également utilisée pour déterminer la durée de la phase oscillante :

$$swingDuration = \frac{strideDuration * (HS2 - \dot{c})}{100}$$

A ce stade, tous les paramètres spatio-temporels et toutes les poses clefs ont été déterminés mais sont stockés dans un unique tableau sans être triés. Pour plus de clarté, les valeurs sont triées et séparées dans des tableaux différents. Les paramètres spatio-temporels font l'objet d'un unique tableau comprenant la vitesse de marche en m/s, la temporalité des événements en pourcentage du cycle de marche, le temps de la foulée en seconde, la durée de la phase oscillante en seconde, la longueur de la foulée, la longueur de pas et la valeur  $\omega_0 = \sqrt{g/L_{membre}}$ . L'amplitude de la trajectoire du CoM dans la direction médio-latérale est un unique réel double précision. Concernant les poses clefs, chaque trajectoire ou partie de trajectoire fait l'objet d'un unique *vector* contenant la position des poses, puis la valeur des poses et enfin la valeur de la dérivée en fin de trajectoire.

Pour chaque trajectoire, le tri des poses clefs dans les *vector* se fait en plusieurs étapes. Tout d'abord, un tableau 2D est créé pour ne contenir que les poses de cette trajectoire, poses principales et

complémentaires incluses, les colonnes représentant la position et la valeur des poses. Elles sont alors triées chronologiquement. En effet, avec la fonction utilisée pour réaliser l'interpolation, les poses doivent obligatoirement être dans l'ordre chronologique. Les poses passent ensuite par un tri pour déterminer lesquelles garder et lesquelles supprimer. Dans la partie présentant la détermination des poses clefs, le potentiel problème d'avoir des poses trop rapprochées temporellement a été abordé. Ce potentiel problème est également présent entre les poses principales et complémentaires puisque certaines des poses principales peuvent être quasi-confondues avec des poses complémentaires. Pour éviter ce cas de figure les poses complémentaires situées à moins de 5% du cycle de marche d'une pose principale sont supprimées et les poses restantes sont alors stockées dans un *vector*. Une fois les poses clefs à garder triées dans l'ordre chronologiques elles sont stockées dans les *vector* prévus à cet effet.

## 1.5. Instanciation des classes

L'étape suivante consiste à instancier les classes du contrôleur. Chaque classe développée pour ce projet est utilisée pour instancier un objet ayant un rôle spécifique dans le contrôleur et est écrite dans un fichier dédié pour plus de clarté. Les classes sont introduites ci-après mais leur fonctionnement sera décrit dans leur ordre d'utilisation dans le contrôleur lors de la description du contrôle. De plus, une description plus précise des données membres est disponible en annexe.

### 1.5.1. Classe pour la détermination des trajectoires de référence

La première classe à être instanciée se nomme *ReferenceTrajectories*. Elle fournit les trajectoires de référence à chaque pas de temps, et les adapte lors des changements de phases (Annexe A). Elle possède entre autres comme données membres des conteneurs pour stocker les poses clefs et les dérivées des trajectoires avant et après adaptation. Elle est instanciée en un objet nommé *referenceTrajectories*. Les données relatives aux poses clefs avant adaptation, les dérivées en fin de trajectoire, les paramètres spatio-temporels, l'amplitude de la trajectoire du CoM médio-latérale et les coefficients de mise à l'échelle sont directement initialisés avec les valeurs fournies par *statisticalModel* dans le fichier principal et ne seront plus modifiées lors de la simulation. Les autres données sont quant à elles initialisées et modifiées lors de la simulation en fonction de l'état du système.

### 1.5.2. Classe du contrôleur proportionnel dérivé

La deuxième classe à être instanciée est nommée *PDCController* (PDC). Elle détermine les couples articulaires à appliquer pour suivre les trajectoires articulaires de référence grâce à un contrôleur proportionnel dérivé (Annexe B). La classe est utilisée dans le programme principal pour instancier un objet nommé *pdcontroller*.

### 1.5.3. Classe du contrôle du talon en phase oscillante

La troisième classe à être instanciée est nommée *SwingFootControl* (SFC). Elle contrôle la position du pied en phase oscillante pour suivre sa trajectoire de référence via l'application de forces virtuelles sur le talon (Annexe C). La classe est utilisée pour instancier un objet nommé *swingFootControl*.

### 1.5.4. Classe du contrôle du CoM

La quatrième classe à être instanciée est nommée *ComControl* (CC). Elle contrôle la vitesse du modèle et une partie de son équilibre via l'application de forces virtuelles sur le CoM (Annexe D). La classe est utilisée pour instancier un objet nommé *comControl*.

### 1.5.5. Classe de compensation de la gravité

La cinquième classe à être instanciée est nommée *GravityCompensation* (GC). C'est cette classe qui permet de calculer les couples articulaires nécessaires pour compenser l'effet de la gravité sur un segment et donc permettre aux autres briques de contrôle d'utiliser des gains moins élevés pour contrôler ces segments (Annexe E). La classe est utilisée pour instancier un objet nommé *gravityCompensation*. Une première fonction membre est alors directement appelée pour calculer le poids de chaque segment.

### 1.5.6. Classe de calcul des forces de réaction horizontales

La sixième classe à être instanciée est nommée *GroundReactionForces*. C'est dans cette classe qu'est défini le modèle de contact permettant de calculer les forces de frictions (Annexe F). La classe est utilisée pour instancier un objet nommé *groundReactionForces*.

### 1.5.7. Classe du contrôleur global

La dernière classe à être instanciée est nommée *GlobalController*. Elle est définie comme une classe enfant de la classe *Controller* présente dans la bibliothèque d'OpenSim. La classe *Controller* est notamment en charge de transmettre la valeur des couples à appliquer aux actionneurs. La classe *GlobalController*, développée pour ce projet étant une classe enfant de la classe *Controller*, elle a

également ce rôle mais est également utilisée pour stocker les données sur l'état du système et appeler les fonctions des différentes classes de contrôle à chaque pas de temps (Annexe G). La classe est utilisée pour instancier un objet nommé *controller*. Elle possède entre autres comme données membres des pointeurs vers les objets des différentes classes de contrôle et vers des tableaux d'actionneurs. Les tableaux et objets précédemment instanciés sont donc passés à l'objet *controller* pour que celui-ci ait accès aux différentes briques de contrôles et aux actionneurs du modèle pendant la simulation

## 1.6. Conditions initiales

L'étape suivante consiste à déterminer et appliquer l'état initial du système, c'est-à-dire la valeur et la vitesse des DoF. Les DoF initiaux sont déterminés en appelant des fonctions de l'objet *referenceTrajectories* de sorte à ce qu'elles renvoient l'état cible du système à 90% du cycle de marche ( $rTime = 90\%$  et  $lTime = 40\%$ ). Initialement, la simulation commençait au début du cycle de marche du membre inférieur droit, c'est-à-dire au contact du pied droit avec le sol. Cependant, au temps 0 de la simulation, le modèle de contact ne calcule pas encore de forces, et il faut donc un certain temps après le début de la simulation pour que la distance d'interpénétration des géométries de contact du pied en phase d'appui se stabilise. Durant cette période, les forces de réaction de Hunt-Crossley ne sont donc pas encore correctement déterminées et il en est de même avec les forces tangentielles. Commencer la simulation au moment du contact du pied droit avec le sol résulte donc en des forces de réaction initiales mal déterminées à un moment critique du cycle. Pour résoudre ce problème, la simulation commence à 90% du cycle de marche du membre inférieur droit. Cela permet au modèle de contact du pied gauche d'être stable avant que le pied droit ne touche le sol. Les seules exceptions pour les articulations concernent la flexion latérale et rotation axiale droite-gauche lombaire. Ces DoF étant contrôlés de sorte à compenser les angles de flexion et de rotation du pelvis, leur état initial est également déterminé comme l'opposé des angles du pelvis.

Une fois ces DoF initiaux déterminés et la pose initiale connue, la hauteur du pelvis est déterminée. Pour cela, la position moyenne du point le plus bas des sphères de contact situé au niveau de l'articulation métatarsophalangienne est obtenue via une fonction de l'objet *controller*. La hauteur du pelvis est alors ajustée pour faire en sorte que cette valeur soit 0, c'est-à-dire que les géométries soient en contact mais sans interpénétration. La trajectoire de la hauteur du CoM n'ayant pas été interpolée, la vitesse du CoM sur l'axe vertical a été déterminée pour le modèle de base par un processus d'essais-erreurs.

Une fois la pose initiale et la hauteur du modèle connue, l'angle de l'articulation métatarsophalangienne du pied gauche doit également être déterminé. En effet, son angle cible étant de  $0^\circ$ , si cette valeur était gardée, les orteils seraient, à ce moment du cycle, enfoncés dans le sol et produiraient une force de réaction non désirée. La même fonction que précédemment est utilisée pour connaître la position du point le plus bas des sphères liées aux orteils, et l'angle du DoF est ajusté pour que cette valeur soit 0, de manière similaire à précédemment.

Concernant la position du modèle dans la direction antéropostérieure, elle est indiquée par la position du repère du pelvis. Cependant, pour l'analyse des résultats, c'est la position du CoM qui est étudiée. La position du pelvis est donc ajustée pour que la position du CoM dans la direction antéropostérieure en début de simulation soit à 0 et la vitesse du pelvis est ajustée pour que la vitesse du CoM corresponde à celle du modèle statistique. La même opération est réalisée pour la direction médio-latérale. La seule différence vient du fait que la sinusoïde représentant la trajectoire médio-latérale du CoM étant centrée sur 0, la position du CoM n'est pas ajustée sur 0, mais sur la valeur de la trajectoire à 90% du cycle de marche.

## 1.7. Manager

Le manager est une classe fournie par OpenSim qui gère l'exécution de la simulation. C'est également via cette classe que la méthode d'intégration est choisie. OpenSim propose différentes méthodes d'intégration à savoir : Euler explicite, Runge-Kutta de second ordre, Runge-Kutta de troisième ordre, Runge-Kutta-Feldberg, Runge-Kutta-Merson, Euler semi-explicite et Verlet. Le choix de la méthode d'intégration s'est fait au cours du développement du simulateur pour déterminer laquelle était la plus stable. Le choix final s'est porté sur la méthode Euler semi-explicite.

C'est également via le manager que la durée de simulation et le pas de temps est décidé. Lors du développement du simulateur, la fréquence de simulation a été fixée à 5000 Hz pour limiter au maximum les instabilités numériques.

## 2. Détermination des matrices de passage et jacobiennes

### 2.1. Matrices de passage

Dans différentes parties de la simulation, les matrices de passage du système sont requises. C'est notamment le cas pour calculer la position des points liés aux pieds dans le repère global. Ces matrices de passage ont une forme invariable définie par les caractéristiques du modèle, mais doivent être mises à jour à chaque pas de temps pour prendre en compte les nouvelles valeurs des DoF. La forme de ces matrices étant invariable, plutôt que de les recalculer à chaque pas de temps en réalisant les produits matriciels successifs, leur forme est déterminée une unique fois dans un script python différent.

La détermination des matrices se fait dans un script python en calcul symbolique en utilisant la bibliothèque *sympy*. Chaque matrice de passage est définie en utilisant le nom du tableau stockant les valeurs de DoF dans le contrôleur, et en prenant en compte les coefficients de mise à l'échelle. L'on peut prendre comme exemple la matrice de passage entre le repère du pelvis et du fémur :

$$T_i = \begin{bmatrix} \cos(\text{DoFValues}[6]) & -\sin(\text{DoFValues}[6]) & 0 & \text{PelvisRfemurx} * \text{scaleCoef}[0] \\ \sin(\text{DoFValues}[6]) & \cos(\text{DoFValues}[6]) & 0 & \text{PelvisRfemury} * \text{scaleCoef}[0] \\ 0 & 0 & 1 & \text{PelvisRfemurz} * \text{scaleCoef}[0] \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Où *PelvisRfemurx*, *PelvisRfemury* et *PelvisRfemurz* représentent la distance entre les origines du pelvis et du fémur droit dans le repère du pelvis. Une fois toutes les matrices écrites, les matrices de passage qui seront nécessaires au contrôleur sont calculées puis écrites telles quelles dans le contrôleur. Puisque les valeurs des angles utilisent directement le nom des variables utilisées dans le contrôleur, aucune modification supplémentaire n'est nécessaire.

## 2.2. Jacobiennes

Le contrôleur a également besoin de différentes jacobiniennes pour les calculs de forces virtuelles. Comme expliqué précédemment, les éléments des jacobiniennes décrivent la variation de la position d'un point par rapport à une rotation autour d'un axe. La matrice utilisée est la matrice transposée de la jacobienne. Elle comporte autant de lignes que de DoF utilisés pour appliquer la force virtuelle, et trois colonnes représentant les trois dimensions. Pour déterminer les éléments, chaque ligne de la matrice jacobienne transposée est calculée par produit vectoriel de l'axe de rotation du DoF et de la distance entre le DoF et le point d'application de la force :

$$J(p)_i^T = (\vec{a}_i \times (p - b_i))^T$$

Où  $\vec{a}_i$  est l'axe de rotation du DoF *i*, *p* est la position du point d'application de la force virtuelle, et *b<sub>i</sub>* est la position du DoF *i*, le tout exprimé dans le repère global. Là encore, la forme des matrices est invariable et est donc déterminée par calcul symbolique pour être écrite telle quelle dans le contrôleur.

La première étape consiste à déterminer les matrices de passage. Les axes des DoFs peuvent alors être déterminés à partir de celles-ci, comme par exemple pour l'axe de flexion du fémur droit :

$$rFemurzAxis = T_i[:3, :3] * [0 \ 0 \ 1]^T$$

Où  $T_i[:3, :3]$  sont les trois premières lignes et colonnes de la matrice de passage, soit la matrice de rotation, entre le fémur et le repère global. La position des DoF et des points d'applications sont alors également déterminés grâce aux matrices de passage, puis utilisés pour déterminer les matrices jacobiniennes.

## 3. Contrôle global

L'objet *controller* stocke les données des caractéristiques de la simulation à chaque pas de temps, appelle les fonctions nécessaires à la mise à jour des trajectoires de référence si besoin, et utilise les objets de contrôle pour unifier les différentes valeurs de contrôle et les transmettre aux actionneurs. La fonction principale de cette classe, héritée de la classe *Controller*, est appelée à chaque pas de temps par la classe *Manager* et permet de transmettre les valeurs de couples aux actionneurs. La première tâche de cette fonction est d'appeler la fonction qui s'occupe de mettre à jour les variables concernant l'état du système.

### 3.1. Mise à jour de l'état de la simulation

Les premières données mises à jour concernent le temps. Le temps actuel de la simulation en seconde est directement obtenu via une fonction d'OpenSim. A partir du temps actuel et du temps au pas de temps précédent, le pas de temps est calculé pour servir au calcul des vitesses. Enfin, le pas de temps en pourcentage du cycle de marche est calculé pour servir au calcul des vitesses à partir des trajectoires qui sont elles-mêmes exprimées en pourcentage.

$$timeStepPercent = \frac{timeStepSecond * 100}{strideDuration}$$

Les deuxièmes données mises à jour concernent l'état du CoM dont les valeurs de positions et de vitesse sont directement obtenues via une fonction d'OpenSim. Quant au XCoM, il est calculé grâce aux valeurs de position et de vitesse du CoM obtenues précédemment.

Les troisièmes données mises à jour concernent les valeurs des DoF, également obtenues via une fonction d'OpenSim.

Les quatrièmes données mises à jour concernent les forces de Hunt-Crossley. Pour chacun des pieds, les forces des huit sphères liées à ce pied sont sommées et le résultat, représentant la force de réaction totale du pied, est stocké dans *rGrfY* ou *lGrfY*.

Les cinquièmes données mises à jour concernent la position et la vitesse du talon dans le repère global. Pour les déterminer, un script python est utilisé pour calculer les matrices de passage entre le repère global et les segments des pieds en calcul symbolique. Les matrices sont alors codées telles quelles dans la classe *GlobalController*, et la valeur de chaque élément est recalculée à chaque pas de temps en fonction des nouvelles valeurs des DoF. Une fois les valeurs des matrices mises à jour, il est alors possible de les utiliser pour déterminer la position globale d'un point des segments du pied Eq. (13). Avant de mettre à jour les positions des talons, les positions stockées sont copiées pour garder en mémoire la position au pas de temps précédent. Les nouvelles positions sont alors stockées, et leurs vitesses sont déterminées par différenciation grâce aux valeurs du pas de temps précédent et à la durée du pas de temps.

Les sixièmes données mises à jour concernent la position, la vitesse et l'orientation des points d'application des forces de réaction tangentielles. Les positions et les vitesses sont calculées de la même

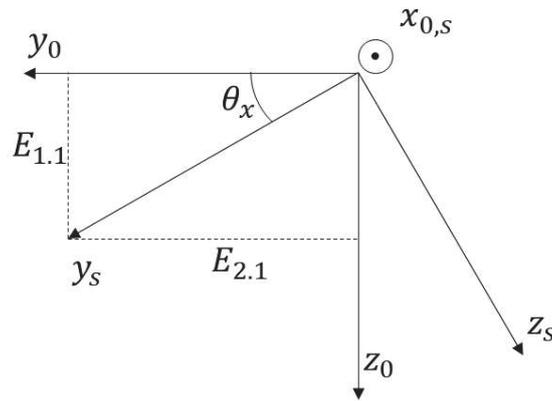
manière que pour les talons. Les orientations sont quant à elles déterminées via les éléments des matrices de passage. En effet, la matrice de rotation, incluse dans la matrice de passage, permet d'exprimer les composantes du repère d'un segment dans le repère global et inversement tel que :

$$\begin{matrix} \vec{X}_s & \vec{Y}_s & \vec{Z}_s & \vec{OO}_s \\ & & & \vec{X}_0 \\ & & & \vec{Y}_0 \\ {}_s^0T & = & \vec{Z}_0 \\ & & & \hat{i} \end{matrix}$$

$$\begin{matrix} \vec{X}_s & \vec{Y}_s & \vec{Z}_s \\ [{}^0_sR] & = & \begin{matrix} \vec{X}_0 \\ \vec{Y}_0 \\ \vec{Z}_0 \end{matrix} \begin{bmatrix} E_{0,0} & E_{0,1} & E_{0,2} \\ E_{1,0} & E_{1,1} & E_{1,2} \\ E_{2,0} & E_{2,1} & E_{2,2} \end{bmatrix} \end{matrix}$$

$$\vec{X}_s = E_{0,0} \cdot \vec{X}_0 + E_{1,0} \cdot \vec{Y}_0 + E_{2,0} \cdot \vec{Z}_0$$

Connaissant les composantes des vecteurs du repère du segment dans le repère global, il est alors possible de déterminer l'angle entre les repères autour d'un axe.



$$\theta_x = \sin^{-1}\left(\frac{E_{2,1}}{\sqrt{E_{2,1}^2 + E_{1,1}^2}}\right)$$

Les positions, orientations et vitesses des points sont alors stockées.

### 3.2. Détection des évènements

La simulation est rythmée par la succession de phases de marche pendant lesquelles le contrôle est différent. Le contrôle de certains membres étant différent selon la phase du cycle dans laquelle ils se trouvent, il est nécessaire que le contrôleur détecte les changements de phase. Cette détection se fait grâce aux forces de Hunt-Crossley calculées par les géométries de contact et dont les données ont été acquises précédemment. La phase du cycle de marche est alors stockée dans une variable ayant comme valeurs possibles 0, 1, 2, 3 et 4. Les valeurs de 1 à 4 correspondent respectivement au premier double appui, simple appui, second double appui et à l'oscillation du membre inférieur droit. La marche étant symétrique, les membres inférieurs et supérieurs droits suivent l'ordre  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ , pendant que les membres gauches suivent l'ordre  $3 \rightarrow 4 \rightarrow 1 \rightarrow 2$ . Les DoFs liés au pelvis et à l'articulation lombaire n'ayant pas de symétrie comme les membres inférieurs et supérieurs, ils suivent le même ordre que les membres de droite. La phase 0 correspond quant à elle à la phase dans laquelle le modèle se situe en début de simulation, le temps que le modèle de contact se stabilise. En effet, puisque les forces de réaction calculées par le modèle de contact dépendent de la distance et de la vitesse d'interpénétration des géométries de contact, il existe un état d'équilibre dans lequel l'interpénétration des géométries est stable. Du fait des erreurs dues à la simulation numérique et des hypothèses simplificatrices du modèle de contact, l'état du modèle ne pourra jamais parfaitement suivre cet état mais pourra tout de même s'approcher suffisamment pour produire une simulation stable. Puisque cet état n'est pas connu en début de simulation, les forces de contacts calculées ne correspondent pas aux forces attendues. Commencer la simulation en début de nouvelle foulée alors que le contact du pied sur le sol représente justement un moment critique dans la modélisation du contact compliquerait alors le contrôle du modèle. Pour éviter cette difficulté, la simulation débute donc légèrement avant la pose du pied droit pour laisser le temps au contact du pied gauche de se stabiliser avant le contact du pied droit. Du point

de vue des membres de droite, la phase 0 correspond à la phase 4 mais est numérotée comme phase 0 pour indiquer au contrôleur que le modèle doit être contrôlé différemment durant cette phase. En effet, les trajectoires de référence du talon en phase oscillante étant déterminées en fonction de l'état du modèle au TO, n'ayant pas eu lieu pour la phase 0, le modèle est alors uniquement contrôlé par le PDC.

La phase 1, correspondant à la première phase de double appui du membre inférieur droit et débutant à la pose du pied droit sur le sol, est détectée lorsque la somme des forces de réaction calculée par les deux sphères de contact liées au talon droit devient non nulle. À ce moment-là, la phase de marche prend la valeur 1, et une donnée concernant les transitions de phase prend la valeur « 4->1 ». Cette donnée a par défaut la valeur 0. Lorsqu'un changement de phase est détecté, elle prend une valeur particulière pour le pas de temps actuel, permettant au contrôleur d'appeler les fonctions nécessaires lors d'un changement de phase. La donnée *lastHS* prend la valeur du temps actuel de la simulation en seconde. Elle permet de garder en mémoire la temporalité de cet événement, ce qui sera nécessaire à la détermination du temps en pourcentage du cycle de marche. Une fonction est alors appelée pour transmettre les valeurs actuelles des forces de réaction verticales à l'objet *comControl*.

Le fait que le contrôle soit différent en fonction de la phase du cycle provoque une discontinuité des couples articulaires lors des changements de phases. Pour éviter cela, à chaque changement de phase, les valeurs de contrôle actuellement stockées, et correspondant donc aux valeurs du pas de temps précédent, sont stockées dans la donnée *phaseTransitionTorque*. Lors d'un changement de phase devant provoquer une discontinuité des couples articulaires, le nouveau contrôle n'est alors pas immédiatement suivi à 100%, mais une période est définie pendant laquelle les valeurs de couples articulaires passeraient progressivement de leur valeur lors du changement de phase, à la valeur déterminée par le nouveau contrôle.

La phase 2, correspondant à la phase oscillante du membre inférieur gauche et débutant au lever du pied gauche, est détectée lorsque *IGrfY* devient nulle. La phase de marche prend la valeur 2 et la donnée de transition de phase prend la valeur « 1->2 ». Comme pour le changement de phase précédente, la donnée *phaseTransitionTorque* est mise à jour.

La marche étant symétrique, les détections des phases suivantes sont identiques en inversant simplement le côté des forces de réaction prises en compte. Lors de la détection de la phase 3, la donnée de phase prend la valeur 3, la donnée de transition de phase prend la valeur « 2->3 », *lastHS* prend la valeur actuelle du temps de la simulation, les forces de contact sont enregistrées et *phaseTransitionTorque* est mis à jour. Lors de la détection de la phase 4, la donnée de phase prend la valeur 4, la donnée de transition de phase prend la valeur « 3->4 » et *phaseTransitionTorque* est mis à jour.

### 3.3. Détermination du temps de cycle

Les trajectoires de référence étant exprimées en pourcentage du cycle de marche, le temps de la simulation en seconde doit être converti en pourcentage pour être utilisé par le contrôleur. La marche

étant symétrique, la pose du pied contra latéral sur le sol est considérée comme étant toujours située à 50% du cycle de marche. La détermination du temps de cycle en pourcent se base alors sur le côté ayant débuté son cycle le plus récemment et le temps de cycle du côté opposé est considéré comme étant cette valeur plus 50%. Par exemple, si le côté droit se situe à 30% du cycle de marche, alors le côté gauche est à 80% du cycle.

Lors des phases 1 et 2, c'est le membre inférieur droit qui a commencé son cycle le plus récemment et *strideTimeSecond* est calculé tel que :

$$strideTimeSecond = currentSimulationTime - lastHS$$

Cette donnée correspond donc au temps écoulé depuis le dernier évènement de la pose du pied sur le sol et est utilisée pour calculer le temps en pourcentage du cycle pour les côtés droit et gauche :

$$rTime = \frac{strideTimeSecond * 100}{strideDuration}$$

$$lTime = rTime + 50$$

Dans le cas des phases 3 et 4, les calculs de *rTime* et *lTime* sont inversés.

Dans le cas où le pas de temps actuel correspond au début d'une phase oscillante (transition de phase = « 1->2 » ou « 3->4 »), le temps en seconde théorique de la phase oscillante est calculé ainsi :

$$swingDuration = \frac{(100 - lTime) * strideDuration}{100}$$

si le membre inférieur gauche est en phase oscillante et inversement sinon. De plus, la temporalité de l'évènement en pourcentage du cycle est stockée dans *lastTOPercent*.

Ces deux valeurs permettent, lors de la phase oscillante, de déterminer le temps en pourcentage de la phase oscillante. En effet, si les trajectoires des DoF et du CoM sont exprimées en pourcentage du cycle de marche, les trajectoires du talon sont, elles, exprimées en pourcentage de la phase oscillante. Dans le cas où c'est le membre inférieur gauche qui est en phase oscillante, elle est alors calculée telle que :

$$swingTime = \frac{100 * (lTime - lastTOPercent)}{(100 - lastTOPercent)}$$

Le pas de temps en pourcent de la phase oscillante est alors également déterminé tel que :

$$swingTimeStepPercent = \frac{timeStepSecond * 100}{swingDuration}$$

Si le pied en phase oscillante ne touche pas le sol avant 100% du cycle de marche, *rTime* ou *lTime* sont autorisés à dépasser 100%. Les trajectoires étant cycliques, la fonction renvoyant les trajectoires de référence se met alors à renvoyer les trajectoires de début de cycle pour éviter de rester sur une valeur de position ou de vitesse fixe. Si cela arrive, bien que les DoF suivis par le PDC

commencent à suivre les trajectoires de début de cycle alors que le modèle n'est encore qu'en fin de cycle, la perturbation est jugée acceptable en comparaison des erreurs numériques dues à la simulation même, et ce tant que la différence entre la temporalité de la pose théorique du pied et celle de la pose réelle du pied n'est pas trop importante. En revanche, les trajectoires du talon n'étant pas cycliques, *swingTime* n'est pas autorisé à dépasser 100%. Dans ce cas, les couples appliqués aux DoFs par le SFC restent constants. Cette solution permet d'éviter un changement brusque de couple dû une variation rapide de la position ou vitesse de référence. De même que pour le suivi des trajectoires articulaires, cette erreur est jugée acceptable tant que temporalité de la pose théorique du pied et celle de la pose réelle du pied n'est pas trop importante.

### 3.4. Détection des phases d'appui du pied

La détection des différentes phases d'appui se fait dans la classe de contrôle pour directement transmettre les informations sur les positions, vitesses et orientations des points d'application des forces au modèle de contact sans avoir à redéfinir les matrices de passage dans la classe gérant le modèle de contact. Ces informations sont passées via une fonction de l'objet *groundReactionForces* et prenant comme paramètres les positions et vitesses des points d'application des forces, les phases d'appui droite et gauche, et le temps de la simulation. Lorsque le temps de la simulation est égal à 0, les valeurs du point d'application des forces liées aux orteils gauche sont mises à jour dans l'objet.

En début de première phase d'appui, c'est-à-dire en début de première phase de double appui, puisque seuls les éléments du talon produisent une force de réaction, seules les valeurs de position et orientation du point lié au talon sont utilisées pour initialiser les valeurs de référence des éléments viscoélastiques du talon.

En début de deuxième phase d'appui, c'est-à-dire en début de phase de simple appui, le pied ne roule pas, et donc seules les valeurs de positions des points liés aux orteils sont utilisées pour initialiser les valeurs de référence pour les éléments viscoélastiques des orteils.

Lors de la troisième phase d'appui, c'est-à-dire à partir de 60% de la phase de simple appui, les points de référence ne changent pas mais le pied peut rouler sur le sol donc seules les valeurs d'orientation des points liés aux orteils sont utilisées pour initialiser les valeurs de référence pour les éléments viscoélastiques de torsion des orteils.

Lors de la phase oscillante, les points de références n'ont pas à être mis à jour mais la fonction s'en occupant est tout de même appelé pour mettre à jour la phase d'appui.

### 3.5. Trajectoires de référence

Avant d'acquérir les poses cibles des DoF pour le pas de temps actuel, si la transition de phase est différente de 0, les trajectoires de référence sont adaptées pour éviter les discontinuités. En effet, les

trajectoires sont interpolées de sorte que leur valeur et leur dérivée à 0% et 100% du cycle coïncident. Si le pied ne touche pas le sol à exactement 100% du cycle de marche, la trajectoire de référence à ce moment-là sera alors différente de la valeur qu'elle aura à 0%, créant une discontinuité de la trajectoire de référence. L'adaptation des trajectoires se fait par l'appel d'une fonction de l'objet *referenceTrajectories*. Lors des poses de pieds, toutes les trajectoires, contra latérales et ipsilatérales, sont adaptées. Lors du levé d'un pied seule la trajectoire de la cheville en phase oscillante est modifiée. De plus, si le changement de phase correspond au début d'une phase oscillante, des données concernant la position et la vitesse du talon en phase oscillante sont enregistrées.

### 3.5.1. Adaptation des trajectoires du CoM et des DoF

L'adaptation des trajectoires s'effectue à chaque changement de phase via une fonction de l'objet *referenceTrajectories* (Figure 43). La méthode d'adaptation est différente selon les DoF et le changement de phase.

Dans le cas d'une trajectoire définie par une seule spline au moment de la pose du pied ipsilatéral, la première étape consiste à réinitialiser les *vector* contenant les valeurs adaptées (ex : *hipAddAdaptedTime* et *hipAddAdaptedValue*). La deuxième étape consiste à calculer la pente de la trajectoire de référence actuelle en %/s (ex : *hipAddBegSlope*) à partir de celle en m/s. En effet, la vitesse de référence fournie au PDC est calculée par la méthode des différences finies en prenant en compte le pas de temps en seconde. Cependant, l'interpolation étant faite en pourcentage du cycle de marche, la dérivée fournie à la fonction créant la spline doit être calculée en prenant en compte le pas de temps en pourcentage. Elle est calculée ainsi :

$$hipAddBegSlope = \frac{dofReferenceTrajectories[7][1]*strideDuration}{100}$$

Où *dofReferenceTrajectories[7][1]* est la vitesse du septième DoF qui est l'adduction-abduction de hanche droite. Par la suite, deux ratios sont déterminés. Le premier est le ratio entre la position actuelle de la trajectoire de référence du DoF et la position de référence en début de cycle, tandis que la valeur du second ratio est fixée à 1 :

$$ratio\ 1 = \frac{dofReferenceTrajectories[7][0]}{hipAddValue[0]}$$

$$ratio\ 2 = 1$$

L'objectif est de déterminer une fonction linéaire ayant pour valeur *ratio 1* à 0% du cycle et *ratio 2* à 100% du cycle, et de multiplier les valeurs des poses clefs initiales par ce ratio de sorte que la trajectoire adaptée débute à la même valeur que la pose de référence actuelle, et finisse à la même valeur que la trajectoire de référence initiale. Les deux coefficients de la fonction sont alors déterminés tels que :

$$a = \frac{\text{ratio 2} - \text{ratio 1}}{\text{hipAddTime.back()} - \text{hipAddTime}[0]}$$

$$b = \text{ratio 1}$$

Où  $\text{hipAddTime.back}()$  et  $\text{hipAddTime}[0]$  sont la dernière et la première temporalité des poses clefs (100% et 0% dans le cas d'une spline unique pour décrire la trajectoire). Les valeurs de poses clefs adaptées sont alors déterminées telles que :

$$\text{hipAddAdaptedTime}[i] = \text{hipAddTime}[i]$$

$$\text{hipAddAdaptedValue}[i] = \text{hipAddValue}[i] * (a * \text{hipAddTime}[i] + b)$$

Dans le cas d'une trajectoire définie par deux splines au moment de la pose du pied ipsilatéral, la méthode est la même que précédemment à part que la fonction linéaire décrivant l'évolution du ratio s'étend sur les deux splines. Ses coefficients sont donc calculés tel que :

$$a = \frac{\text{ratio 2} - \text{ratio 1}}{\text{hipFlex 2 Time.back()} - \text{hipFlex 1 Time}[0]}$$

$$b = \text{ratio 1}$$

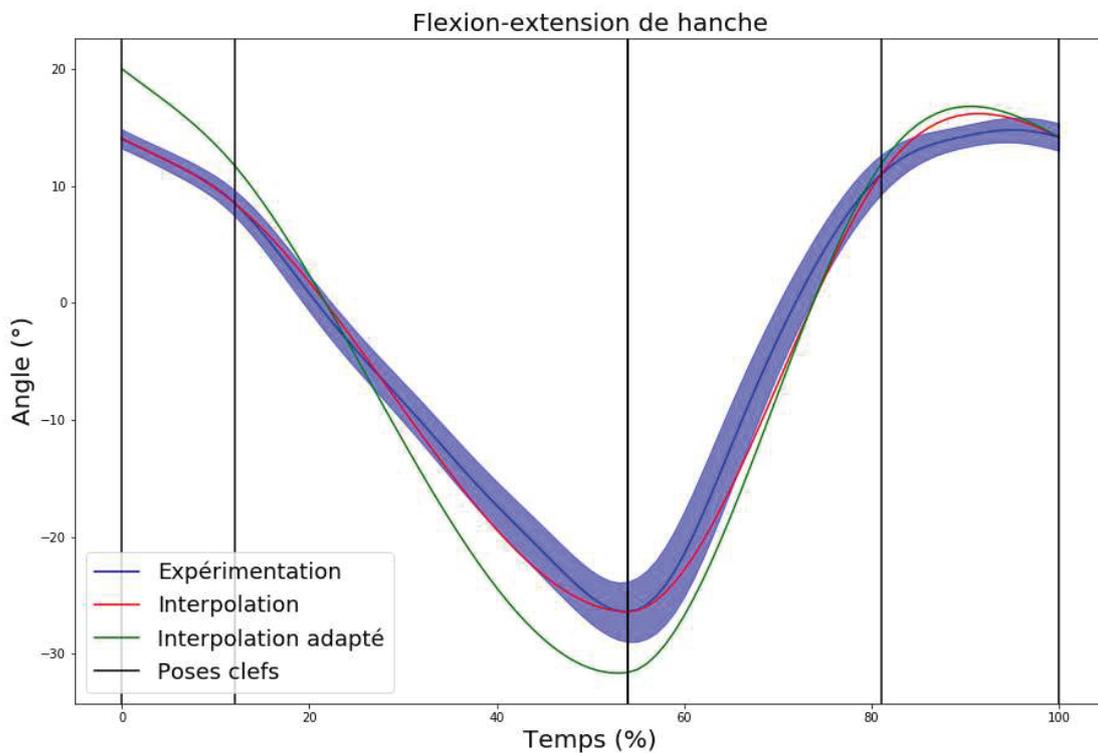


Figure 44 : Exemple d'adaptation de trajectoire de référence.

Dans le cas d'une trajectoire définie par une seule spline au moment de la pose du pied controlatéral, une étape supplémentaire est nécessaire après avoir réinitialisé les *vector* de poses adaptées. En effet, les trajectoires adaptées doivent commencer à 50% du cycle de marche mais tous les DoF ne sont pas assurés d'avoir une pose clef à cet instant précis. Pour y remédier, les poses clefs initiales sont utilisées pour créer une spline représentant la trajectoire de référence initiale. La valeur à 50% de la trajectoire initiale est alors utilisée comme première pose clef du nouveau *vector* de poses adaptées et le *vector* est ensuite complété par les poses se situant à plus de 50% du cycle. Le reste de l'adaptation des poses clefs suit ensuite la même méthode que citée précédemment en adaptant linéairement les poses clefs pour que celle à 50% corresponde à la valeur de référence actuelle et que celle à 100% corresponde à la valeur de la trajectoire initiale.

Dans le cas d'une trajectoire définie par deux splines au moment de la pose du pied contralatéral, la méthode d'adaptation dépend de la position temporelle de la transition entre les deux parties. Si la transition entre les deux parties est avant l'évènement, la trajectoire n'a alors plus besoin que de la seconde partie, et la méthode d'adaptation est exactement la même que celle décrite précédemment pour une trajectoire décrite par une seule spline. Cependant, dans le cas où la transition se fait après l'évènement, la trajectoire est toujours décrite par deux splines et l'adaptation nécessite alors encore une étape supplémentaire. Comme pour le cas précédent, il faut créer une pose clef à 50% du cycle pour la première partie de la trajectoire puis compléter le *vector* de poses avec les poses supérieures à 50% du cycle. Cependant, certaines poses de transition étant proches de 50% du cycle de marche, il n'est pas impossible que le *vector* de poses adaptées de la première partie ne soit composé que de la pose à 50% du cycle et de la pose de transition entre les deux interpolations. La fonction utilisée pour créer les splines nécessitant trois poses différentes, une étape supplémentaire vérifie le nombre de poses disponibles pour interpoler la première partie. Si le nombre de poses est supérieur à 3, aucune opération supplémentaire n'est faite. En revanche, si le *vector* n'est composé que de deux poses, une troisième pose intermédiaire est créée en reprenant la spline de la trajectoire initiale utilisée pour créer la pose à 50%, mais en prenant la valeur de la spline située à mi-distance entre 50% du cycle et le temps de la pose de transition. La suite de l'adaptation est alors la même que pour le cas d'une trajectoire définie par deux splines à la pose du pied ipsilatéral.

L'adaptation de la trajectoire du CoM dans la direction antéro-postérieure dépend du pied qui se pose au sol. Dans le cas de la pose du pied droit, les valeurs temporelles des poses adaptées sont les mêmes que celles des poses initiales, tandis que dans le cas de la pose du pied gauche, les valeurs temporelles des poses adaptées sont les valeurs de poses initiales plus 50 (en % du cycle). Les valeurs de position sont quant à elles définies comme les valeurs initiales plus la valeur de référence du pas de temps précédent. En procédant ainsi, seule la première moitié de la trajectoire est prise en compte. Cependant, la marche étant considérée symétrique, les deux parties de la trajectoire sont supposées identiques. Cette formulation tire profit de cette supposition pour simplifier l'algorithme d'adaptation de la trajectoire par rapport aux DoF.

La trajectoire médio-latérale du CoM étant définie par une fonction plutôt que par des poses clefs, les ratios et coefficients de la fonction linéaire décrivant l'évolution du ratio à appliquer à la

trajectoire de base sont calculés de la même manière que pour les DoF mais sont directement utilisés lors de la détermination des trajectoires.

L'adaptation de la trajectoire de la cheville en phase oscillante est un cas à part. Il est apparu pendant le développement du simulateur que la dorsiflexion de la cheville en phase oscillante était plus faible que dans certains essais expérimentaux, ayant pour conséquence une diminution de la hauteur des orteils provoquant le contact de ceux-ci avec le sol en pleine phase oscillante. Une cause possible de ce problème est la non-correspondance entre la hauteur de la trajectoire du pied et l'angle de la cheville. Plus un sujet monte son pied, moins il a besoin de faire de dorsiflexion et inversement. Les trajectoires des DoF étant déterminées de manière indépendantes les unes des autres, il est possible que les trajectoires calculées provoquent à la fois en une dorsiflexion faible et une trajectoire de pied basse, provoquant le contact prématuré du pied sur le sol. Pour résoudre ce problème, les poses clés de la trajectoire de la cheville en phase oscillante sont déterminées manuellement. La trajectoire choisie est composée de 7 poses clés situées à 0, 7.5, 15, 50, 85, 92.5 et 100% du temps de phase oscillante et ont respectivement pour valeur d'angle la valeur de la trajectoire de référence en début de phase oscillante, la moitié de cette valeur, 0, 0, 0, la moitié de la valeur finale de la trajectoire initiale et enfin, la valeur finale de la trajectoire initiale. L'objectif est de rapidement atteindre une valeur d'angle de 0 et d'en rester le plus proche possible pendant la moitié de la phase oscillante. La valeur de 0 degré est choisie car fidèle aux angles de certains essais expérimentaux et permet d'avoir le pied parallèle au sol lorsque le tibia est perpendiculaire au sol avant de commencer la projection du talon en avant.

### 3.5.2. Calcul des trajectoires du CoM et des DoF

Une fonction de l'objet *referenceTrajectories* est appelée à chaque pas de temps pour déterminer la pose cible actuelle des DoF et du CoM. Pour chaque DoF et pour la trajectoire antéropostérieure du CoM, les poses clés adaptées sont utilisées pour créer la spline représentant la trajectoire de référence adaptée (ex : *rHipFlexSpline*). La valeur de la spline au temps actuel en pourcent est alors stockée en tant que position de référence. Une seconde valeur représentant la vitesse de référence est calculée par la méthode des différences finies en prenant en compte la position au pas de temps actuel, la position au pas de temps suivant, et le pas de temps en seconde pour que la vitesse déterminée soit en m/s :

$$vitesse = \frac{rHipFlexSpline(rTime + timeStepPercent) - rHipFlexSpline(rTime)}{timeStepSecond}$$

Dans le cas où *rTime* ou *lTime* est supérieur à 100 (en % du cycle, la valeur prise en compte est leur valeur actuelle moins 100, ce qui permet de reboucler sur le début de cycle.

La pose de référence du CoM dans la direction médio latérale est déterminée par une fonction sinusoïde :

$$CoM_z = \sin\left(\frac{2*\pi}{100}*rTime\right) - \dot{z} \sin\left(\frac{\frac{2*\pi}{100}*CTO}{2}\right) * comzAmplitude * (aCoMz*rTime + bCoMz) \dot{z}$$

Les valeurs  $\frac{2*\pi}{100}$  dans les deux premiers termes permettent de fixer la période de la sinusoïde à 100. Le deuxième terme permet de décaler la fonction dans le temps pour que la valeur 0 soit la position moyenne du CoM. Le décalage est estimé à  $\frac{CTO}{2}$  qui correspond à la moitié du temps de transfert d'un appui sur l'autre et donc le moment où le modèle est à sa position moyenne. Le troisième terme permet alors à la trajectoire d'avoir l'amplitude déterminée par le modèle statistique. Enfin, le quatrième terme permet d'appliquer le ratio calculé pour adapter la trajectoire.

### 3.5.3. Adaptation et calcul des trajectoires du talon en phase oscillante

Contrairement aux trajectoires des DoF et du CoM qui n'ont besoin d'être adaptées qu'une fois par phase, les trajectoires du talon étant définies par la position d'arrivée, si celle-ci venait à changer au cours de la phase oscillante, les trajectoires devraient à nouveau être adaptées. C'est pourquoi la détermination de la position d'arrivée, l'adaptation des poses clefs et le calcul de la pose de référence actuelle se font dans une unique fonction.

A ce stade du développement du simulateur, la marche simulée est considérée sans perturbation et la position d'arrivée du talon est fixe tout au long de la phase oscillante. La position d'arrivée dans la direction antéropostérieure est définie comme la position du talon du pied en phase d'appui plus la longueur de pas. La position média latérale est quant à elle déterminée par la position théorique du XCoM en fin de cycle, plus une marge de stabilité *MoSz* pour s'assurer que le pied soit extérieur au XCoM.

Ensuite, l'adaptation des poses clefs se fait par la même méthode que les DoF. La seule exception vient de l'adaptation des poses de la hauteur du talon. Comme expliqué lors de la description du choix des poses clefs, un moment critique de la trajectoire du talon dans la direction verticale est le minimum local à environ 70% du cycle de marche. A ce moment-là, le pied passe suffisamment près du sol pour que le manque de précision du contrôle permette au pied de le toucher et de déstabiliser le modèle. Une première solution à ce problème a été apportée en définissant ce point comme jonction des deux parties de la trajectoire de référence pour éviter que celle-ci ne puisse descendre plus bas. Une seconde solution est apportée lors de l'adaptation des poses clefs. Après l'adaptation classique des poses, la valeur du minimum local est modifiée pour correspondre à la valeur du maximum local situé entre 10 et 20% après le minimum (cf. Figure 35). Cette élévation permet alors au reste de la courbe d'être légèrement plus élevée. La valeur est choisie telle quelle car c'est la valeur la plus haute possible qui ne change pas fondamentalement la forme de la trajectoire de référence.

Enfin, les poses de référence pour le pas de temps actuel sont également déterminées via la même méthode que pour les DoF et le CoM.

#### 3.5.4. Acquisition des trajectoires

Une fois les poses de référence du pas de temps actuel calculées et stockées dans l'objet *referenceTrajectories*, elles sont transmises au contrôleur via différentes fonctions de ce même objet.

### 3.6. Appel des briques de contrôle

#### 3.6.1. Calcul des forces de réaction horizontales

La première brique de contrôle appelée est la brique gérant les forces de réaction horizontales du sol. Une première fonction calcule les forces via la méthode décrite dans la partie III.B.2. puis les renvoie. Les forces et moments de réaction sont alors stockés dans des données de *controller*.

#### 3.6.2. Contrôleur proportionnel dérivé

La seconde brique appelée est la brique gérant le PDC en passant comme paramètres les poses de référence des DoF, les valeurs actuelles des DoF et les informations sur la phase et le temps du cycle de marche.

La première étape consiste à appeler une fonction prenant comme paramètres la phase de marche et le temps de la phase oscillante. Cette fonction a pour objectif de déterminer les gains à utiliser au pas de temps actuel (*gain*) en fonction des gains de base (*gainParameters*) et de l'état de la simulation.

Les gains pour la flexion-extension de hanche et l'adduction-abduction de hanche ne varient qu'en fonction de la phase de marche. Leur valeur est directement celle des gains de base lorsque le membre inférieur est en appui au sol et 0 lors de la phase oscillante pour laisser le contrôle des DoF à la brique contrôlant la position du talon.

Les gains pour la rotation interne-externe de hanche ne varient jamais. Initialement, les gains de ce DoF suivaient la même évolution que pour les autres DoF de la hanche, ce qui permettait lors de la phase oscillante de mieux contrôler la position et la vitesse du talon en phase aérienne. Néanmoins, même en phase oscillante, le contrôle de ce DoF devait être repris par le PDC avant la pose du pied pour assurer la bonne orientation du pied lors du contact avec le sol. Une phase de transition entre le contrôle par le SFC et le PDC était alors prévue lors de la phase oscillante pour permettre au SFC de s'aider de ce

DoF pour améliorer le suivi de trajectoire du talon en début de phase, tout en laissant au PDC le contrôle en fin de phase pour assurer l'orientation du pied. Cependant, lors des simulations, le SFC avait tendance à utiliser ce DoF de manière non biofidèle (ex : angle trop important provoquant la collision des membres inférieurs), et le PDC ne parvenait pas à converger suffisamment rapidement vers la trajectoire de référence après avoir repris le contrôle du DoF. Ainsi, il a été décidé de laisser le contrôle de la rotation de hanche au PDC même lors de la phase oscillante. La trajectoire médio-latérale du talon et notamment sa vitesse sont moins bien suivies, mais l'articulation n'est pas utilisée de manière absurde et l'orientation du pied en fin de cycle est assurée.

L'évolution des gains de la flexion-extension du genou est particulière puisque le PDC doit reprendre le contrôle du genou en phase oscillante comme cela était initialement fait pour la rotation interne-externe de hanche. Lorsque le membre inférieur est en appui au sol, le gain est directement celui des gains de base. Lors de la phase oscillante, le gain dépend du temps de la phase oscillante. En effet, au début de la phase, le genou est utilisé par le SFC pour notamment contrôler la hauteur et l'avancée du talon. Cependant, si le contrôle du genou est laissé à cette seule brique de contrôle pour l'entièreté de la phase, le genou est utilisé de telle sorte que le membre inférieur en phase oscillante n'est pas tendu au moment de la pose du pied sur le sol, ce qui n'est pas biofidèle et détériore la marche. Pour s'assurer que le membre inférieur soit tendu en fin de phase, le contrôle du genou est donc rendu au PDC au cours de la phase via une donnée de *pdcontroller* qui permet de gérer la transition. Dans cette donnée est stockée une valeur correspondant à un temps de la phase oscillante. Tant que *swingTime* est inférieur à cette valeur, le gain du genou vaut 0 et le contrôle est donc entièrement laissé au SFC. Lorsque *swingTime* est supérieur à une valeur de transition défini par l'utilisateur et inférieur à 90%, le gain du genou passe linéairement de 0 à la valeur du gain de base. Enfin, au-delà de 90%, le gain du genou est celui du gain de base et le contrôle du genou est donc entièrement laissé au PDC. Le contrôle total du genou est rendu au PDC avant 100% de la phase oscillante dans le cas où le pied se poserait avant 100% du cycle de marche pour que le genou soit déjà dans la position idéale, et pour laisser le temps au PDC de faire converger l'angle du genou à l'angle souhaité avant que le pied ne se pose.

Les gains de la dorsiflexion de cheville, de l'inversion-éversion de l'articulation subtalaire et de la flexion-extension de l'articulation métatarso-phalangienne sont également invariables pendant tout le cycle de marche. Bien que ces DoF du membre inférieur puissent jouer un rôle en phase oscillante, ils ne sont pas utilisés pour contrôler la position du talon et varier leur gain n'est donc pas nécessaire. De même, les gains de l'articulation lombaire et des articulations des bras sont également invariables.

Pour chaque DoF, lorsque le gain en position est déterminé ( $gain[DoF][0]$ ), le gain en vitesse est calculé à partir de celui-ci :

$$gain[DoF][1] = 2 * \sqrt{gain[DoF][0]}$$

Une fois les gains mis à jour, les couples articulaires sont calculés via la formule du PDC Eq. (10) et stockés dans *torque*.

### 3.6.3. Contrôle du talon en phase oscillante

La troisième brique appelée est la brique gérant la pose du talon en phase oscillante en prenant comme paramètre la pose de référence du talon, la pose actuelle du talon, les valeurs des DoF et les informations sur la phase et le temps du cycle de marche.

Dans un premier temps, les tableaux contenant les forces virtuelles à appliquer sur le talon et les couples articulaires sont remis à zéro et les forces à appliquer sont recalculées en fonction de la différence entre la pose de référence et la pose actuelle, à la manière d'un PDC avec comme gain de position et de vitesse 1 et  $2*\sqrt{1}$ .

Ensuite, la donnée représentant la jacobienne du talon à contrôler  $rJt$  ou  $lJt$  est mise à jour. Pour cela, les jacobiennes sont déterminées dans un script python en calcul symbolique et directement écrites dans un fichier à part nommée *HeelJacobian*. Ce fichier dispose de deux fonctions prenant comme argument les valeurs des DoF et les coefficients de mise à l'échelle permettant de calculer et retourner les valeurs des jacobiennes des membres droit et gauche. Les DoF utilisés pour contrôler le talon en phase oscillante sont la flexion-extension et l'adduction-abduction de la hanche ipsilatérale, ainsi que la flexion-extension, l'adduction-abduction et la rotation interne-externe de la hanche contra latérale. En effet, même lorsque la rotation de la hanche ipsilatérale était utilisée pour contrôler le talon, l'utilisation de quatre DoF pour contrôler le talon dans trois directions présentait des limites. Il a donc été décidé d'utiliser en plus les articulations de la hanche contra latérale pour améliorer le contrôle.

Une fois les valeurs de la jacobienne mises à jour, une fonction est appelée pour mettre à jour les gains à utiliser. Habituellement, le rapport entre l'amplitude des couples et la différence entre pose cible et pose actuelle est déterminée par les gains lors du calcul de la force virtuelle à appliquer. Plus les gains sont élevés, plus une même différence entre pose cible et actuelle induira une force virtuelle élevée, et donc des couples élevés :

$$torque[DoF]=rJt[DoF][0]*force[0]+rJt[DoF][1]*force[1]+rJt[DoF][2]*force[2]$$

Où les indices 0, 1 et 2 représentent les directions x, y et z. Dans ce projet, pour plus de précision dans le contrôle du talon, des gains sont directement appliqués à chaque terme du calcul :

$$torque[DoF]=gain[DoF][0]*rJt[DoF][0]*force[0]+gain[DoF][1]*rJt[DoF][1]*force[1]+gain[DoF][2]*$$

Cela permet de donner aux DoF une priorité sur la direction qu'ils permettent de contrôler. En effet, deux forces peuvent provoquer des couples contradictoires sur un même DoF. Dans ce cas, le DoF risque d'osciller autour d'une position et de ne pas permettre au talon de suivre la trajectoire voulue. C'est par exemple le cas de la flexion-extension de hanche au lever du pied. La force antéropostérieure ayant pour but de faire avancer le pied provoquera un couple de flexion, tandis que la force verticale ayant pour but de faire monter le pied provoquera un couple d'extension. Appliquer des gains différents par direction pour chaque DoF permet donc d'éviter ce cas de figure. De plus, en faisant varier les gains au cours de la phase oscillante, il est également possible de faire varier ces priorités. Par exemple, en début de phase oscillante, la flexion de hanche est plus appropriée pour contrôler la position antéropostérieure, tandis

que l'extension de genou est plus appropriée pour contrôler sa hauteur. Cependant, lorsque le tibia est perpendiculaire au sol, le genou n'a plus la possibilité de contrôler la hauteur du talon. Garder sa priorité sur le contrôle de la hauteur à ce moment n'est alors pas utile. Chaque DoF dispose donc d'un gain pour chacune des directions, et la valeur de chaque gain évolue en fonction de trois valeurs de gain de base tout au long de la phase oscillante. Les gains évoluent linéairement entre chaque gain de base selon *swingTime*. La valeur du temps de phase oscillante à laquelle les gains atteignent la deuxième valeur de gain de base est déterminée par l'utilisateur dans une donnée prévue à cet effet. Tant que *swingTime* est inférieur à cette valeur, les gains évoluent de la première valeur à la deuxième valeur. Si *swingTime* est supérieur à cette valeur, les gains évoluent de la deuxième à la troisième valeur. Pour gérer la transition du contrôle du genou du SFC au PDC, un coefficient est appliqué aux gains du genou. La valeur de ce coefficient est inversement proportionnelle à celui de son pendant dans l'objet *pdcontroller* afin que le poids du SFC dans le contrôle du genou soit lui-même inversement proportionnel au poids du PDC.

Une fois les gains mis à jour, si l'un des membres inférieurs est en phase oscillante, les couples pour contrôler le talon sont calculés via l'équation (9) puis retournés au contrôleur.

#### 3.6.4. Contrôle du CoM

La quatrième brique de contrôle appelée est celle appliquant les forces virtuelles sur le CoM pour contrôler sa trajectoire en prenant comme paramètres les poses de référence et actuelles du CoM, les valeurs des DoF, la phase de la marche, le temps de la phase oscillante et les forces de réaction verticales des pieds droit et gauche.

Similairement au contrôle du talon la première étape consiste à réinitialiser les tableaux contenant les couples articulaires et forces à appliquer. Les données *rJt* et *lJt* représentant les jacobiniennes sont mis à jour via deux fonctions situées dans un fichier à part dans lequel sont stockées les jacobiniennes. Comme pour le contrôle du talon, les jacobiniennes ont été déterminées dans un script python en calcul symbolique, puis directement écrites telles quelles dans le code C++. Les DoF utilisés pour l'application des forces virtuelles sont les DoF de l'articulation lombaire et tous les DoF des membres inférieurs en phase d'appui, excepté l'articulation métatarso-phalangienne.

L'étape suivante consiste à déterminer les forces virtuelles à appliquer au CoM. L'objectif de ces forces virtuelles est de contrôler la vitesse du CoM mais aussi d'être un outil supplémentaire pour contrôler l'équilibre du modèle. En s'inspirant des travaux réalisés sur l'équilibre dans le domaine de la biomécanique, il a été décidé de suivre la trajectoire du XCoM. En effet, cette valeur prend à la fois en compte la position et la vitesse du CoM et permet de mettre en pratique la méthode de contrôle utilisant les marges de stabilité. Les forces virtuelles sont donc calculées comme la différence entre le XCoM de référence et XCoM actuel. Un élément supplémentaire est cependant utilisé pour calculer la force virtuelle dans la direction antéropostérieure. Celui-ci est basé sur la vitesse moyenne du modèle depuis le début de la simulation. Bien que le XCoM prenne en compte la vitesse, il est toutefois homogène à une position et cet élément supplémentaire permet de garder un contrôle sur la vitesse. Le fait de prendre en compte la vitesse moyenne depuis le début de la simulation plutôt que la vitesse de référence

instantanée permet alors de corriger les éventuelles accumulations d'erreurs de suivi. Pour calculer les vitesses moyennes de référence et de la simulation, à chaque pas de temps, les vitesses instantanées de référence et de la simulation sont sommées à chaque pas de temps. Les vitesses moyennes sont alors calculées via la division de la somme des vitesses instantanées par le nombre de pas de temps depuis le début de la simulation. Le paramètre supplémentaire est alors calculé comme la différence entre la vitesse moyenne de référence et de la simulation. Enfin, cet élément est multiplié par un paramètre qui détermine donc le poids qu'il doit avoir par rapport à la différence des XCoM dans le calcul de la force virtuelle.

L'étape suivante consiste à mettre à jour les gains qui seront utilisés dans le calcul des couples et à les stocker dans *gain*. En effet, pour plus de précision et similairement au contrôle du talon, l'amplitude des couples par rapport à la différence entre les trajectoires de référence et actuelles n'est pas déterminée lors du calcul des forces virtuelles mais directement lors du calcul des couples via des gains appliqués à chaque élément du calcul. La mise à jour des gains se fait par appel à une fonction prenant comme paramètre la phase de la marche, le temps de la phase oscillante et les forces de réaction du sol verticales. Comme pour le contrôle du talon, les gains peuvent évoluer au cours du cycle. Leur évolution est déterminée par deux gains de base pour chaque direction de chaque DoF. De plus, pour limiter au maximum les pics de forces lors des poses des pieds, les gains évoluent de sorte à commencer à 0 en début de phase de premier double appui, et terminer à 0 en fin de phase de deuxième double appui avant que le membre inférieur n'entre en phase oscillante. Lors de la première phase de double appui, le gain est déterminé comme étant la valeur du premier gain de base multiplié par un coefficient basé sur le ratio entre la force de réaction du pied arrière par rapport à la force de réaction verticale du pied arrière en début de phase de double appui. Par exemple, si c'est le pied droit qui vient de toucher le sol, les gains des DoFs du membre de droite sont calculés tels que :

$$gain[0][0] = gainParameter[0][0] * rCoef$$

$$rCoef = 1 - \left( \frac{lGrfY}{HSGrf[1]} \right)$$

Où  $gain[0][0]$  est le gain de la flexion-extension de hanche droite dans la direction antéropostérieure,  $gainParameter[0][0]$  est le premier gain de base et  $HSGrf[1]$  est la force de réaction verticale du pied gauche au moment de la pose du pied droit. Lors de la phase de simple appui, le gain évolue linéairement de la valeur du premier gain de base au second gain de base entre 0% et 90% du temps de phase oscillante, puis reste à la valeur du second gain de base tant que la phase de simple appui continue. Lors de la seconde phase de simple appui, le gain est calculé de la même manière que lors de la première phase de simple appui, mais en prenant comme référence le second gain de base. Par exemple, si c'est le pied gauche qui vient de toucher le sol, les gains des DoFs du membre de droite sont calculés tels que :

$$gain[0][0] = gainParameter[0][0] * rCoef$$

$$rCoef = \frac{rGrfY}{HSGrf[0]}$$

De cette manière, plus un pied est appuyé au sol, plus il sera utilisé pour contrôler le CoM, ce qui permet d'éviter qu'un membre inférieur venant à peine de se poser ou qu'un membre inférieur sur le point de quitter le sol soit autant utilisé qu'un membre inférieur pleinement appuyé. En revanche, cette méthode fixe la valeur du gain en début de second double appui à la valeur du second gain de base. Si le gain n'a pas atteint cette valeur à ce moment-là, l'évolution du gain, et donc du couple articulaire sera discontinue. C'est pourquoi l'évolution linéaire du gain en phase de simple appui lui fait atteindre cette valeur avant le changement de phase prévu. La raison pour laquelle l'évolution des gains lors du double appui est basée sur l'évolution de la force de réaction verticale du membre arrière à la fois pour les membres inférieurs droit et gauche est due à la mauvaise approximation des forces de réaction du membre avant lors du premier double appui. Celles-ci augmentent puis diminuent fortement pendant le premier double appui avant de se stabiliser. Au contraire, la force de réaction verticale du membre arrière évolue de manière beaucoup plus linéaire, permettant une évolution également plus linéaire des gains.

Au cours du développement du simulateur, deux potentielles limites à l'utilisation des forces virtuelles dans ces conditions de simulations ont été identifiées. Une première limite potentielle vient du fait que lors du double appui le système est en chaîne fermée. Dès lors, une configuration du système pourrait exister de sorte que l'application d'un couple sur l'une des jambes ait un effet opposé à cause de la cinématique du membre contralatéral. Deuxièmement, comme indiqué précédemment, les forces virtuelles sont déterminées comme l'action d'un DoF sur le point d'application de la force virtuelle. En conséquence, le calcul ne prend pas en compte les mobilités du modèle ne se trouvant pas entre le DoF et le point d'application, ce qui peut avoir comme conséquence que l'application d'un couple issu des forces virtuelles ait un effet opposé à celui recherché. Un bon exemple pour illustrer ce phénomène est l'action du couple de cheville avec une mobilité supplémentaire entre le pied et le sol quand celui n'est pas à plat. Lorsque la force virtuelle appliquée au CoM est positive, le couple calculé à la cheville est un couple de dorsiflexion. Or, le couple produit par la cheville en fin de phase d'appui pour propulser le CoM vers l'avant est au contraire un couple de plantiflexion. Cela peut s'expliquer par le fait que le couple de la cheville a également un effet sur le segment du pied, d'autant plus que ce segment a une inertie moindre que le reste du membre inférieur. Si l'articulation de la cheville était fixe dans l'espace, alors un couple de dorsiflexion aurait effectivement toujours pour effet de propulser le CoM vers l'avant. Cependant, la cheville n'étant pas fixe dans l'espace, le couple de cheville peut faire rouler le pied sur le sol, auquel cas l'effet sur le CoM dépend de la configuration du système (Figure 45) :

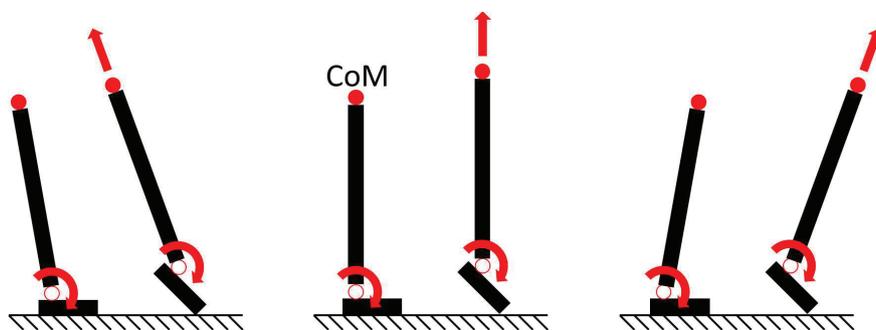


Figure 45 : Effet d'un même couple de plantiflexion sur le pied et sur le CoM selon la configuration du système

La solution retenue pour éviter ces problèmes a été d'autoriser l'utilisation de gains négatifs. Lorsqu'un gain est négatif, le principe utilisé n'est plus exactement celui des forces virtuelles et l'on préférera alors parler de contrôle basé sur la jacobienne.

Une fois les gains mis à jour, les couples sont calculés et stockés dans la donnée *torque* puis retournés au contrôleur.

### 3.6.5. Compensation de la gravité

La cinquième brique de contrôle appelée est celle gérant la compensation de la gravité en prenant comme paramètre la valeur des DoF, la phase de marche et le temps de phase oscillante.

Pour compenser le poids de chaque segment, une force virtuelle opposée au poids est appliquée à leur CoM. Pour chaque segment, les DoF utilisés pour appliquer la force virtuelle sont les DoF situés entre le segment et le pelvis. Pour chaque DoF, un couple par segment est calculé. Si le DoF sert à compenser le poids de plusieurs segments, les couples sont additionnés.

La première étape consiste à mettre à jour les jacobiennes. Puisque les forces ont uniquement une composante verticale, seules les parties nécessaires des jacobiennes sont écrites. Comme pour les jacobiennes servant au contrôle des talons et du CoM, elles sont déterminées par calcul symbolique dans un script python puis directement copiées dans le contrôleur. Puisqu'il y a moins d'éléments, elles sont directement écrites comme données de la classe. De plus, ce contrôle n'agit que sur les segments qui ne sont pas entre un segment en phase d'appui et le pelvis. Les jacobiennes des bras sont donc toujours mises à jour, mais celles des membres inférieurs ne sont mises à jour que lorsqu'elles sont en phase oscillante.

Les couples articulaires sont alors déterminés. La seule exception concerne le genou en phase oscillante. En effet, les gains du PDC étant invariables, ils doivent être suffisamment élevés pour permettre au PDC de suivre les trajectoires des DoF lorsque ceux-ci font partie du membre inférieur d'appui et sont donc par défaut suffisants pour suivre la trajectoire des DoF lorsqu'ils sont en phase oscillante. La compensation de la gravité pour les DoF des membres inférieurs sert donc principalement pour minimiser les gains du contrôle du talon. Or, le contrôle du genou passant du SFC au PDC au cours de la phase oscillante, un coefficient est appliqué au couple du genou pour que celui-ci passe linéairement à 0 en même temps que le changement de contrôle. Ainsi, la compensation de la gravité n'est pas utilisée en même temps que le PDC.

Les couples résultants sont alors retournés au contrôleur.

## 3.7. Détermination des couples totaux

Une fois les couples issus des différentes briques de contrôle acquis, ils sont sommés et stockés dans la donnée *totalTorque*. Dans le cas où un changement de phase vient de se produire, les couples sont adaptés pour éviter les discontinuités et amoindrir les pics de couple liés au contact.

Dans le cas d'un changement de phase dû à la pose d'un pied, tous les couples des membres inférieurs sont adaptés. L'adaptation se fait entre 0% du nouveau cycle et une valeur de temps stockée dans *HSTorqueAdaptationTime*. Pour ce faire, le couple total évolue linéairement de la valeur du couple au moment de la pose du pied, stockée dans *phaseTransitionTorque*, et la valeur calculée par le contrôleur :

$$totalTorque = \frac{totalTorque * rTime}{HSTorqueAdaptationTime} + phaseTransitionTorque * \left( \frac{-rTime}{HSTorqueAdaptationTime} + 1 \right)$$

Cette adaptation était à l'origine nécessaire pour la flexion-extension et l'adduction-abduction de la hanche venant de se poser car le contrôle passait instantanément du SFC + GC au PDC + CC en provoquant une discontinuité. Cependant, elle a été étendue à tous les DoF des membres inférieurs car numériquement, l'impact du pied sur le sol est un moment critique pouvant provoquer des pics de forces, et donc par réaction, des pics de couples articulaires. Cette adaptation permet donc à la fois de transiter d'un contrôle à l'autre sans discontinuité, et d'amoindrir les instabilités dues au contact du pied sur le sol.

Une adaptation est également réalisée au début de la phase oscillante, mais uniquement pour les DoF du membre inférieur débutant la phase oscillante. Cette adaptation est nécessaire car plusieurs DoF de ce membre inférieur changent instantanément du PDC au SFC + GC. L'adaptation se fait exactement de la même manière que la précédente mais en utilisant *swingTime* au lieu de *rTime* ou *lTime*, et sur un laps de temps défini par l'utilisateur. L'envol du pied ne provoquant pas les mêmes instabilités que la pose du pied, seuls les DoF du membre inférieur en phase oscillante sont adaptés.

Finalement, la donnée de transition de phase est remise à 0 si besoin, et les couples sont transmis aux actionneurs pour être utilisés dans l'intégration des équations du mouvement, terminant le rôle du contrôleur pour le pas de temps actuel.

### 3.8. Schématisation des briques de contrôle actives

La Figure 46 résume quels DoFs sont contrôlés par les différentes briques de contrôle en fonction de la phase de la marche.

		Premier double appui	Simple appui	Second double appui	Phase oscillante
Flexion-extension de hanche	PDC				
	SFC				
	CC				
	GC				
Adduction-abduction de hanche	PDC				
	SFC				
	CC				
	GC				
Rotation interne-externe de hanche	PDC				
	SFC				
	CC				
	GC				
Flexion-extension de genou	PDC				
	SFC				
	CC				
	GC				
Dorsi-planti flexion de cheville	PDC				
	CC				
	GC				
Inversion-éversion subtalaire	PDC				
	CC				
	GC				
Flexion-extension métatarsophalangienne	PDC				
Flexion-extension lombaire	PDC				
	CC				
	GC				
Inclinaison latérale droite-gauche lombaire	PDC				
	CC				
	GC				
Rotation axiale lombaire	PDC				
	CC				
	GC				
Membres supérieurs	PDC				
	GC				

Figure 46 : Contrôle des DoFs selon la phase de marche. Les parties sombres représentent les adaptations de couple après changement de phase.

## E. Réglage des paramètres

Les différentes parties du contrôleur dépendent de paramètres pour calculer ou adapter les couples à appliquer. Le réglage de ces paramètres est une étape importante puisque ce sont eux qui déterminent le comportement du contrôleur.

### 1. Réglage manuel

Une méthode pour trouver ces paramètres est celle de l'essai erreur où ils sont manuellement réglés par l'opérateur puis testés. Cela demande une connaissance de l'effet de chaque paramètre sur le contrôle de la part de l'opérateur. Cette connaissance peut tout d'abord venir d'a priori. Par exemple, la flexion-extension de la hanche devrait avoir plus d'impact sur la position antéropostérieure et verticale du talon que sur la position médio-latérale. Il serait donc logique que les gains utilisés pour contrôler ce DoF donnent plus de poids aux forces antéropostérieure et verticale qu'à la force médio-latérale. Cependant, la complexité du système fait qu'un paramètre peut avoir un effet différent de celui escompté comme par exemple avec la plantiflexion de la cheville en début de cycle (cf. partie III.D.3.6.4.). Des connaissances supplémentaires sur le comportement du contrôleur en fonction des paramètres viennent alors de l'expérience de l'opérateur après avoir réalisé différents essais. De plus, modifier un paramètre lié à un DoF modifiera le comportement de ce DoF et donc le comportement de la chaîne cinématique. Les paramètres sont donc interdépendants et plus le système est complexe, plus il sera sensible aux changements de paramètres, et plus l'opération de réglage sera chronophage voire impossible, ce qui est le cas de ce contrôle.

## 2. Réglage par optimisation

Une autre méthode pour déterminer les paramètres est l'optimisation numérique. Dans ce cas, c'est l'optimisation qui fixe la valeur des paramètres à tester et qui évalue le résultat via une fonction de coût. L'avantage de cette méthode est que les différents ensembles de valeurs des paramètres sont testés beaucoup plus rapidement que manuellement, et que l'évolution de la valeur des paramètres se fait selon un algorithme développé spécifiquement pour cette tâche plutôt que par interprétation visuelle des effets des paramètres.

La méthode d'optimisation qui a été utilisée ici est nommée « stratégie d'évolution avec adaptation de la matrice de covariance » (CMA-ES) (Hansen et al., 2010). Les optimisations utilisant une stratégie d'évolution se basent sur le principe d'évolution biologique pour produire les nouveaux ensembles de paramètres à tester. A partir d'une génération parente d'ensembles de paramètres, une génération fille est déterminée par recombinaisons et mutations des meilleurs ensembles parents. La génération fille devient alors la génération parente pour la détermination de nouveaux ensembles à tester. Les dépendances entre les paramètres sont stockées dans une matrice de covariance. Dans les stratégies d'adaptation de la matrice de covariance, cette matrice est mise à jour de sorte à améliorer les chances de déterminer des ensembles avec de bons résultats. Cette méthode est l'une des plus efficaces et est largement utilisée notamment pour les optimisations hors-ligne en infographie.

D'un point de vue technique, l'optimisation est réalisée via un script python utilisant la bibliothèque *pycma* (*pycma*, 2016/2022). Elle permet notamment de réaliser les évaluations des ensembles de paramètres en calcul parallèle pour diminuer le temps de calcul. Pour fonctionner, elle a besoin d'une solution initiale, d'une fonction de coût et d'une valeur nommée *sigma* qui est la déviation standard initiale des paramètres et qui détermine à quel point les valeurs de paramètres testées peuvent être éloignées des valeurs initiales. La valeur de *sigma* étant identique pour tous les paramètres, ceux-ci

doivent être mis à l'échelle pour que l'évolution des paramètres soit la même pour tous. Pour cela, un domaine de valeur sur lequel peut s'étendre les valeurs possibles d'un paramètre a été déterminé pour chaque paramètre en fonction de l'expérience retirée de la phase d'essais erreurs. Tous les domaines ont alors été mis à l'échelle pour s'étendre de 0 à 1000. La valeur de *sigma* doit alors être adaptée pour permettre de chercher des valeurs sur l'ensemble du domaine.

Lors de l'optimisation, à chaque nouvelle évaluation, les paramètres sont écrits dans un fichier texte. Dans le programme principal du simulateur, le fichier texte contenant les paramètres à évaluer est lu pour en extraire les données et les utiliser pour initialiser les paramètres dans le contrôleur. La simulation est alors réalisée avec ces paramètres. La fonction de coût est directement définie dans le contrôleur. Les différents éléments utilisés pour déterminer sa valeur sont déclarés comme des données de la classe *GlobalController* et leurs valeurs sont mises à jour à chaque pas de temps. A la fin de la simulation, la valeur de la fonction de coût est calculée puis stockée dans un second fichier texte. Une fois l'exécution de la simulation terminée, le script python ouvre le fichier texte contenant la valeur de la fonction de coût et l'extrait avant de supprimer les fichiers textes pour libérer de l'espace. Les évaluations étant lancées en parallèle, les paramètres doivent être stockés dans des fichiers spécifiques pour éviter les problèmes d'accès, ce qui peut arriver si l'optimisation tente d'écraser un fichier au moment où il est lu par le simulateur. Pour cela, le nom des fichiers textes créés contiennent le numéro du *thread* gérant cette évaluation particulière. Le numéro de *thread* est alors également passé comme argument à l'exécutable du simulateur, ce qui lui permet d'ouvrir le fichier texte correspondant. La même méthode est ensuite utilisée pour le fichier contenant la valeur de la fonction de coût.

En raison du grand nombre de paramètres, les optimiser simultanément avec un domaine de recherche suffisamment étendu ne permet pas à l'optimisation de converger vers une solution optimale. Afin de tout de même optimiser tous les paramètres, la solution adoptée a été de réaliser l'optimisation en plusieurs étapes. Les premières étapes consistent à optimiser les paramètres liés à une brique de contrôle particulière à chaque étape. L'objectif est d'optimiser les paramètres de chaque brique de contrôle une première fois indépendamment des autres briques de contrôle. Les résultats de ces optimisations partielles pourront alors être utilisés comme solution initiale pour une optimisation globale de tous les paramètres sur un domaine de recherche restreint. A chaque étape, l'optimisation pouvait être réalisée plusieurs fois si le résultat n'était pas jugé convaincant pour passer à l'étape suivante. Une première modification possible entre chaque optimisation d'une même étape était une modification de la fonction de coût utilisée. Cela a notamment permis de déterminer les critères importants pour l'optimisation de chaque brique de contrôle. Une seconde modification possible était une modification du contrôleur lui-même. En effet, outre le réglage des paramètres, l'optimisation permet également de déterminer si la forme actuelle du contrôleur est capable de produire le résultat souhaité. Si ce n'est pas le cas, alors des modifications sont à apporter au contrôleur lui-même, et le résultat des optimisations peut permettre d'orienter le choix des modifications à apporter. Enfin, puisque chaque résultat d'étape sert de solution initiale à l'étape suivante, et que les premières étapes se concentrent sur une brique de contrôle, réaliser l'optimisation en plusieurs étapes permet de mettre en avant l'utilité des différentes briques de contrôle.

Etant donné que les premières étapes de l'optimisation se concentrant sur les briques de contrôle gérant l'équilibre de manière individuelle, elles ne sont réalisées que sur un seul pas. En effet, les deux aspects de l'équilibre, à savoir le contrôle du CoM et le contrôle du pied en phase oscillante, doivent être correctement paramétrés pour pouvoir enchaîner un deuxième pas. De fait, tant que ces deux briques de contrôle gérant les aspects de l'équilibre ne sont pas correctement paramétrées, il a été jugé préférable de ne réaliser les optimisations que sur un seul pas pour limiter le temps de calcul.

## IV. Optimisations et résultats

Pour rappel, le cycle de marche pour l'un des membres inférieurs est divisé en quatre phases délimitées par des événements. La première phase d'une foulée est le premier double appui. Elle commence au contact du pied ipsilatéral sur le sol (HS1) et continue jusqu'au lever du pied controlatéral (CTO). La deuxième phase est le simple appui. Elle commence au CTO et continue jusqu'à la pose du pied controlatéral sur le sol (CHS). La troisième phase est le second double appui. Elle commence au CHS et continue jusqu'au lever du pied ipsilatéral (TO). La quatrième phase est la phase oscillante. Elle commence au TO et continue jusqu'à la pose du pied ipsilatéral (HS2). Pour ce projet, ces phases sont numérotées de 1 à 4 correspondant respectivement au premier double appui, simple appui, second

double appui et phase oscillante du pied droit. Les résultats présentés dans cette partie débutent tous lors du contact du pied droit sur le sol, c'est-à-dire au début de la phase 1. Les évènements y sont représentés par les traits verticaux, en commençant par le HS1, le CTO et le CHS lorsque la simulation s'étend sur un seul pas (phase 1 et 2), puis le TO et HS2 lorsque la simulation s'étend sur une foulée entière.

## A. Solution Initiale

Comme évoqué précédemment, la solution initiale a été déterminée manuellement par un processus d'essai erreur. Celle-ci ne se déroulant que sur un pas, seuls les gains indispensables pour générer un mouvement ont été initialisés. En l'occurrence, les gains du contrôle du CoM (CC) et la majorité des gains du contrôle du talon en phase oscillante (SFC) sont nuls. Les résultats de la simulation utilisant la solution initiale sont présentés dans les figures 47, 48 et 49.

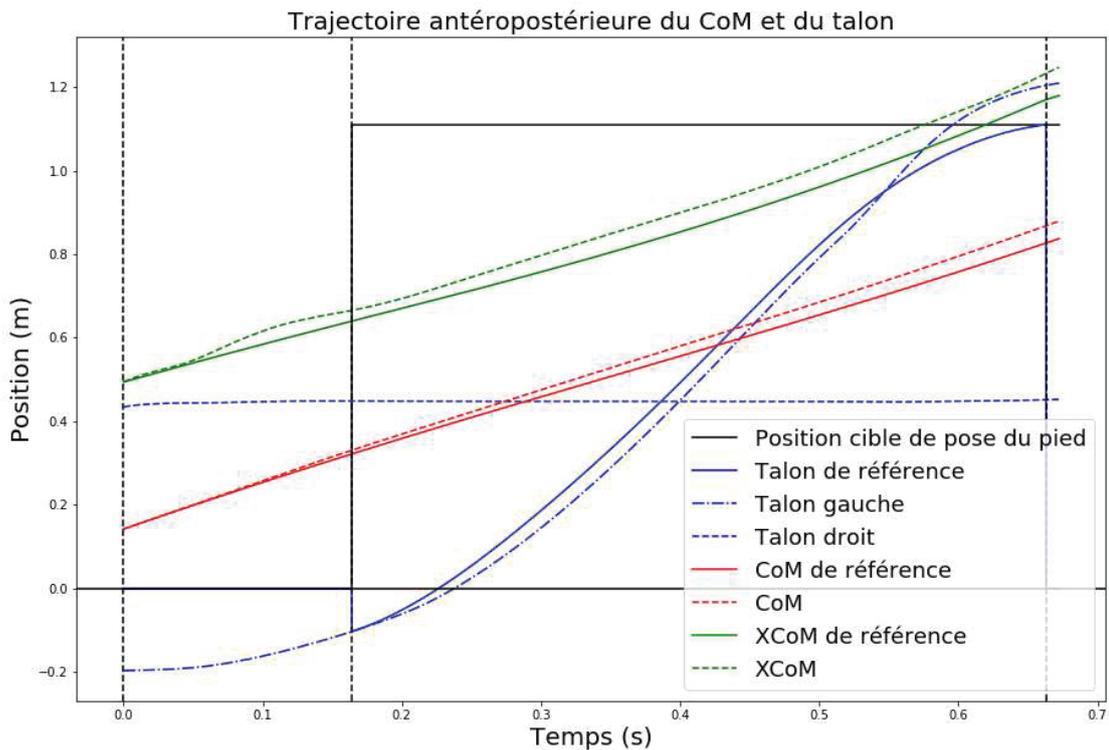


Figure 47 : Trajectoire antéropostérieure du CoM et des talons lors du premier pas de la solution initiale

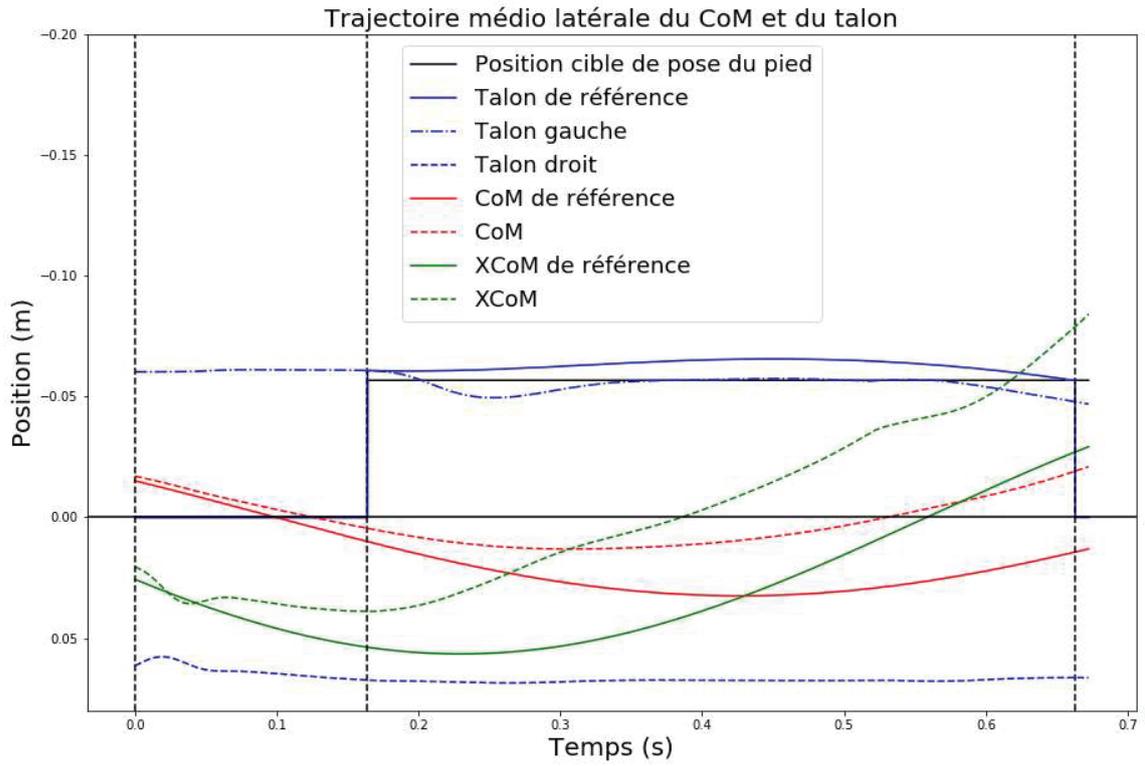


Figure 48 : Trajectoire médio-latérale du CoM et des talons lors du premier pas de la solution initiale

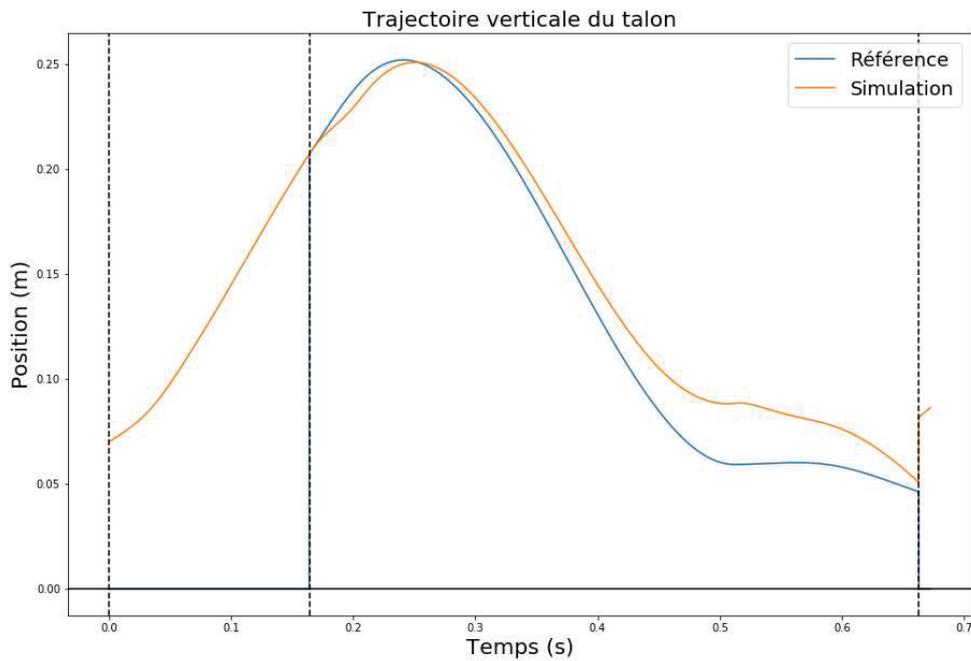


Figure 49 : Trajectoire verticale du talon gauche lors du premier pas de la solution initiale

La figure 47 montre que sans contrôle du CoM, celui-ci avance plus rapidement que celui de référence. De plus le contrôle de la position du talon n'est pas suffisamment précis. Il en résulte un dépassement non négligeable de la trajectoire de référence en fin de phase oscillante du membre gauche.

La figure 48 montre que le balancement médio-latéral du CoM et son renvoi vers le côté en phase oscillante est trop rapide, et ce, dès la phase de double appui. Cela a pour conséquence que le XCoM en fin de pas est trop latéral, au point d'être plus latéral que la position de pose du pied. De plus, le contrôle du talon dans cette direction n'est pas suffisamment précis non plus. La différence avec la trajectoire de référence est certes moindre que dans la direction antéropostérieure, mais l'évolution de cette dernière est également moins importante que dans cette direction.

La figure 49 montre également que le contrôle de la trajectoire du talon n'est pas suffisamment précis puisque la trajectoire réelle est fortement surélevée par rapport à la trajectoire de référence.

## B. Tests d'optimisation

Avant de débiter les différentes étapes de l'optimisation, des tests ont été menés pour s'assurer que les outils d'optimisation étaient correctement implémentés et que les optimisations étaient effectivement capables de converger vers une solution optimale. Un premier test visait à optimiser les paramètres du SFC et du modèle de contact en utilisant la moyenne des écarts quadratiques moyen (RMSE) entre les forces de réactions prédites par le modèle statistique et celles de la simulation comme fonction de coût. Un second test visait à optimiser simultanément les gains du contrôleur proportionnel dérivé (PDC) et du CC en utilisant dans l'ordre d'importance la moyenne des RMSE des trajectoires antéropostérieures et médio-latéral du XCoM, la différence du temps de pas, la RMSE de la vitesse moyenne et la moyenne des RMSE des forces de réaction comme éléments de la fonction de coût. En testant les paramètres issus de ces tests dans les mêmes conditions que la solution initiale, c'est-à-dire en initialisant les gains du CC à zéro mais en gardant tous les gains du SFC, le résultat est plus proche des trajectoires de référence comme le montrent les figures 50, 51 et 52.

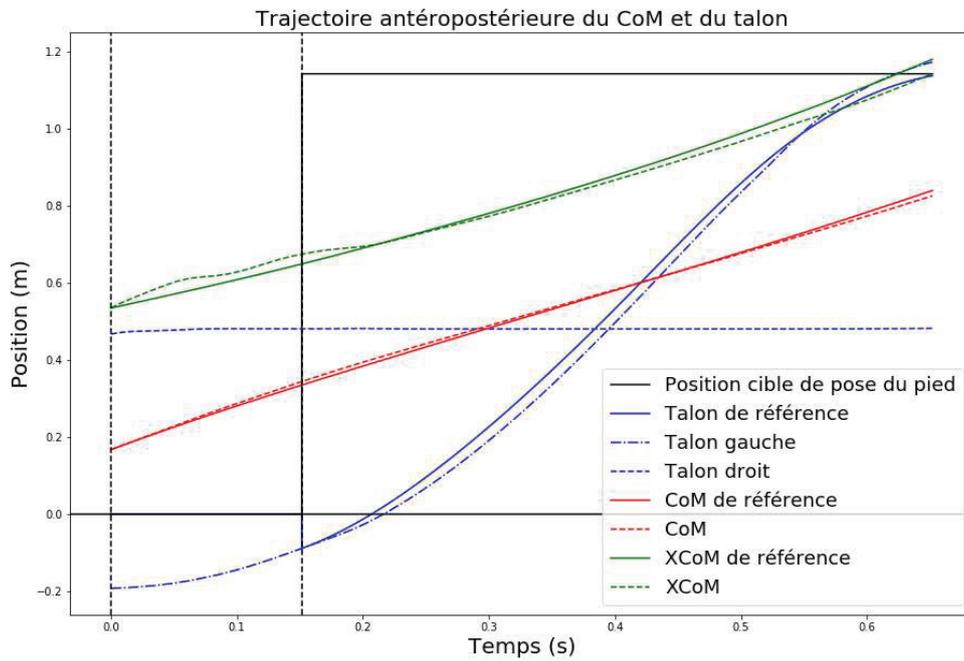


Figure 50 : Trajectoire antéropostérieure du CoM et des talons lors du premier pas après tests d'optimisation

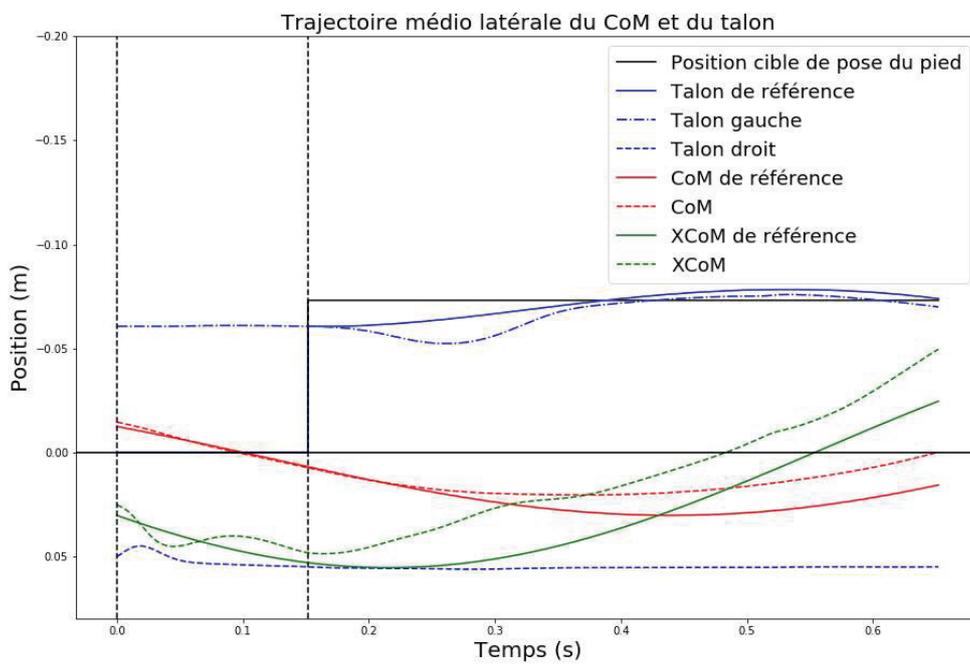


Figure 51 : Trajectoire médio-latérale du CoM et des talons lors du premier pas après tests d'optimisation

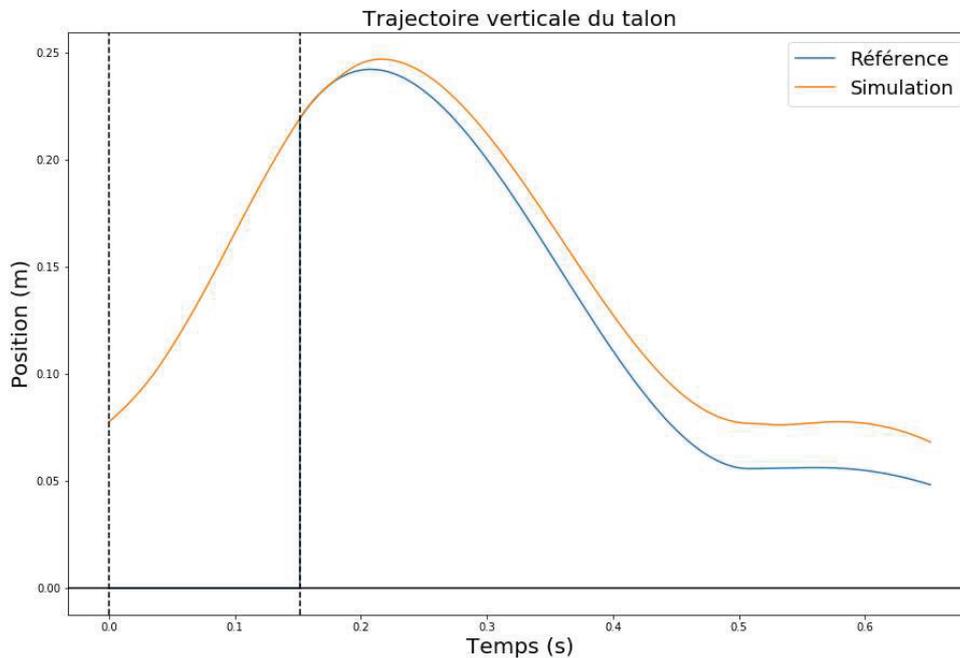


Figure 52 : Trajectoire verticale du talon gauche lors du premier pas après tests d'optimisation

L'évolution de l'écart quadratique moyen entre les trajectoires de référence et de simulation avant et après les tests d'optimisation est présentée dans la table 4.

RMSE (m)	XCoMx	XCoMz	Talon x	Talon y	Talon z
<b>Solution Initiale</b>	0.042	0.034	0.038	0.014	0.007
<b>Tests d'optimisation</b>	0.021	0.017	0.024	0.014	0.005

Table 4 : Evolution des RMSE avant et après les tests d'optimisations.

Un premier résultat intéressant concerne le contrôle du talon. Un mauvais paramétrage de l'optimisation dans cette étape de test ne permettait de chercher des solutions que sur un champ de possibilités de l'ordre de  $10^0$  autour de la solution initiale alors que les paramètres étaient de l'ordre de  $10^2$ . Pourtant, le processus d'optimisation a malgré tout réussi à améliorer le contrôle. Cela montre d'une part que l'optimisation est effectivement capable d'améliorer le contrôle, mais également que ce dernier est sensible au paramétrage. Enfin, le fait d'avoir optimisé simultanément les gains du PDC et du CC a permis de déterminer des gains de PDC plus à même de le faire travailler avec le CC. En effet, les gains du PDC après optimisation étaient généralement plus faibles, permettant au CC d'avoir un effet plus important sur le contrôle. Cette solution permettant en plus un meilleur suivi des trajectoires du

CoM même sans contrôle direct de celui-ci, les paramètres du PDC et du SFC issus de ces tests sont utilisés comme solution initiale pour l'optimisation séquentielle.

## C. Optimisation du contrôle du CoM

Le premier aspect de l'équilibre à être optimisé est le contrôle du CoM. Le choix d'optimiser le suivi du CoM en premier vient du résultat attendu des optimisations. A la fin du premier pas de la solution initiale, le CoM est décalé par rapport à sa position de référence. Après optimisation, il est attendu que le XCoM respecte mieux la trajectoire de référence, ce qui aura pour effet de décaler latéralement le modèle. Si les paramètres du SFC avaient déjà été optimisés pour permettre au talon de suivre sa trajectoire de référence alors que le suivi du XCoM était incorrect, ils risqueraient de ne plus être optimaux lorsque le modèle suit mieux le XCoM. Au contraire, une fois le suivi du XCoM optimisé, l'optimisation des paramètres du SFC devraient avoir moins d'influence sur le suivi du XCoM qu'inversement.

### 1. Fonction de coût

La fonction de coût utilisée pour cette étape est la suivante :

$$f = \text{Pénalité} + 10^3 * XCoM_{RMSE} + 10^2 * pelvis_{RMSE} + 10^1 * CoMv_{RMSE} + 10^0 * Grf_{RMSE} + \text{Gains}$$

Le terme *Pénalité* est constitué de deux parties nommées *CoMh* et *contact*. Ces deux parties ont normalement une valeur nulle. En cas de comportement marquant une perte de stabilité définitive lors de la simulation, elles prennent une valeur d'un million, ce qui représente une valeur ayant plusieurs ordres de grandeur de plus que les autres termes de la fonction afin de pénaliser la solution testée. La partie *CoMh* prend sa valeur de pénalité si la hauteur du CoM passe en dessous de 70% de la longueur des jambes, indiquant que le modèle est en train de tomber. La partie *contact* prend sa valeur de pénalité si le contact avec le sol est entièrement perdu. Cette situation peut par exemple se produire si le contact indiquant le pas suivant arrive trop tôt, provoquant une propulsion du modèle vers le haut sans pour autant suffisamment altérer les autres éléments de la fonction pour considérer la solution comme mauvaise.

Le terme suivant est défini par :

$$XCoM_{RMSE} = \frac{XCoM_{x_{RMSE}} + XCoM_{z_{RMSE}}}{2}$$

Dans notre simulateur, le contrôle de l'équilibre est basé sur le XCoM. Son contrôle est donc indispensable à la stabilité du modèle. Puisque la brique du CC a pour rôle principal de contrôler la position du XCoM, c'est ce terme qui a le poids le plus important dans la fonction.

Le terme suivant est défini par :

$$pelvis_{RMSE} = \frac{pelvisTilt_{RMSE} + pelvisList_{RMSE} + pelvisRot_{RMSE}}{3}$$

Les trajectoires du pelvis n'étant pas directement contrôlées, ce terme est présent pour empêcher d'améliorer le suivi du XCoM au détriment trop important de la qualité des trajectoires du pelvis. En effet, une trop mauvaise trajectoire du pelvis ne permettra pas d'enchaîner plusieurs pas. Cependant, cette étape d'optimisation ne se réalisant que sur un pas, sans ce terme, l'optimisation tend justement à améliorer le suivi du XCoM au détriment des trajectoires du pelvis.

Le terme  $CoMv_{RMSE}$  représente l'erreur quadratique moyenne entre la vitesse moyenne de référence et de simulation sur l'ensemble du temps de simulation. Respecter la vitesse moyenne permet de s'assurer que si la vitesse instantanée n'est pas toujours respectée, le retard ou l'avance prise est rattrapé par la suite. Théoriquement, plus le suivi du XCoM antéropostérieur est correct, moins l'erreur sur la vitesse moyenne est importante. Ainsi, un poids inférieur au suivi des trajectoires du pelvis est attribué à ce terme.

Le terme  $Grf_{RMSE}$  est défini par :

$$Grf_{RMSE} = \frac{rGrf_{x_{RMSE}} + rGrf_{y_{RMSE}} + rGrf_{z_{RMSE}} + lGrf_{x_{RMSE}} + lGrf_{y_{RMSE}} + lGrf_{z_{RMSE}}}{6}$$

Initialement, les paramètres du modèle de contact horizontal faisaient également partie des paramètres optimisés. L'objectif était de continuellement améliorer le modèle de contact en prenant en compte les effets des différentes briques de contrôle. Cependant, en permettant de modifier ces paramètres uniquement en fonction de la différence avec les forces de réaction de référence, les forces permettaient aux pieds de glisser sur le sol. De fait, ce terme est gardé pour préférer les paramètres améliorant le plus possible les forces de réaction, et a donc le poids le moins élevé de la fonction.

Le terme  $Gains$  représente la moyenne des gains du CC. Il est présent pour minimiser les gains et de fait, les couples issus de cette brique. En effet, certains DoFs tels que la flexion-extension ou l'adduction-abduction de hanche ont un effet limité sur le suivi du XCoM. Par conséquent, l'optimisation tend à leur attribuer une valeur de gain élevé pour améliorer le suivi. Cela a notamment pour conséquence de produire des couples importants, résultant en un éloignement important de la trajectoire de référence, et donc de la production d'un couple opposé important de la part des autres

briques de contrôle. En utilisant cet élément, le résultat sur le suivi est légèrement altéré, mais les gains des DoFs les moins utiles sont alors presque nuls.

## 2. Résultats

Les résultats de cette étape d'optimisation sont présentés dans les figures 53, 54 et 55. L'évolution des critères utilisés pour quantifier le résultat du contrôleur est présentée dans la table 5.

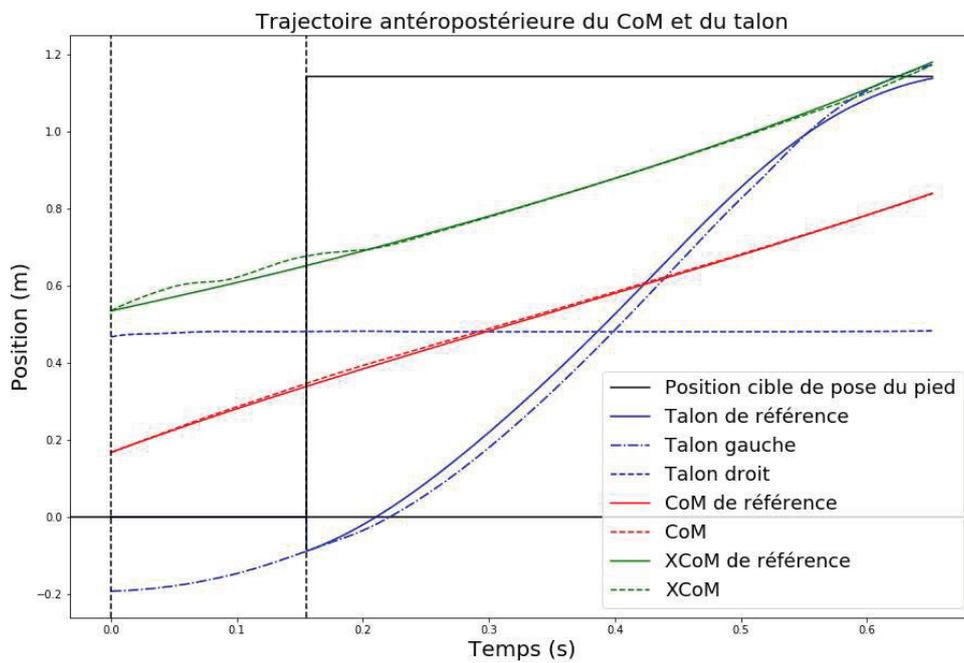


Figure 53 : Trajectoire antéro-postérieure du CoM et des talons lors du premier pas après optimisation du CC

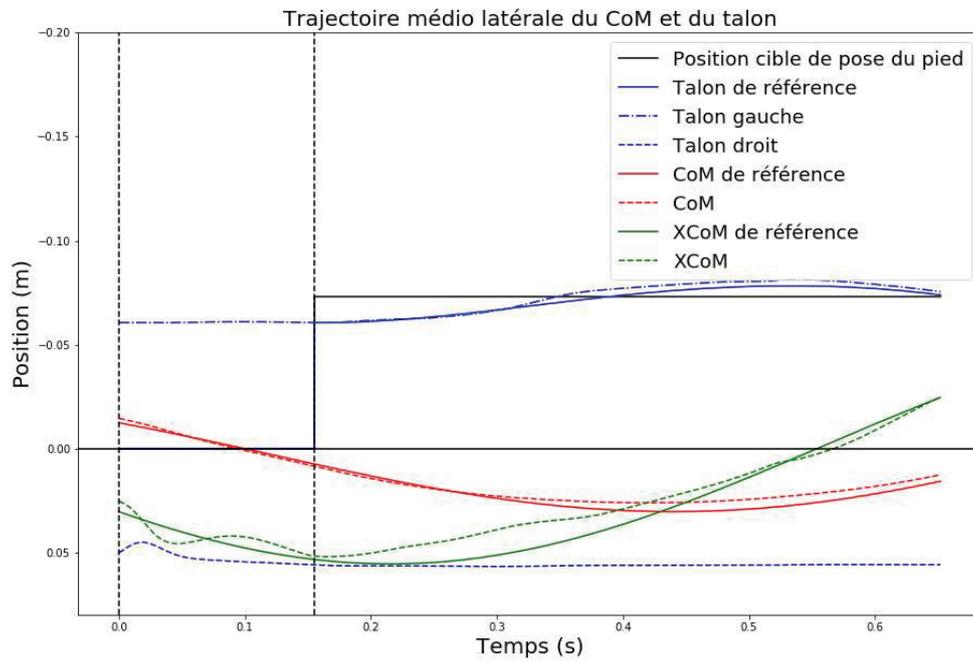


Figure 54 : Trajectoire médio-latérale du CoM et des talons lors du premier pas après optimisation du CC

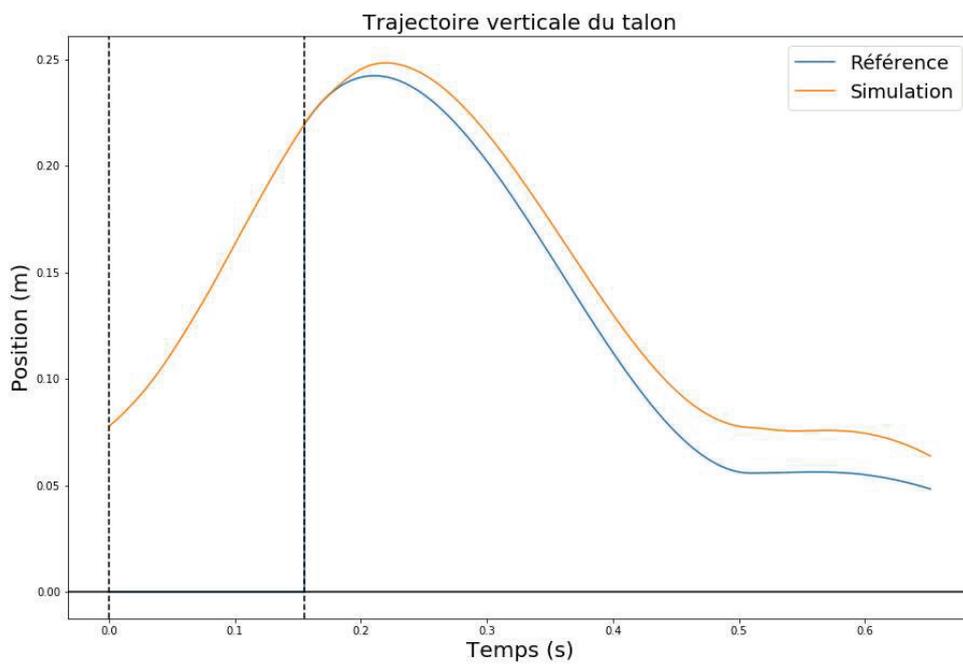


Figure 55 : Trajectoire verticale du talon gauche lors du premier pas après optimisation du CC

RMSE (m)	XCoMx	XCoMz	Talon x	Talon y	Talon z
Tests d'optimisation	0.021	0.017	0.024	0.014	0.005
Optimisation CC	0.011	0.006	0.027	0.014	0.002

RMSE (°)	Inclinaison sagittale du pelvis	Inclinaison frontale du pelvis	pelvis Rot
Tests d'optimisation	1.49	1.031	6.302
Optimisation CC	2.636	0.917	3.209

RMSE (m/s)	vitesse CoMx
Tests d'optimisation	0.056
Optimisation CC	0.034

Table 5 : Evolution des RMSE avant et après optimisation du CC.

La table 5 montre qu'après optimisation, l'écart moyen entre les trajectoires de référence et de simulation diminue d'un centimètre dans chaque direction. Cela représente un suivi de respectivement près de deux fois et trois fois plus précis dans les directions antéropostérieure et médio-latérale. De plus, les figures 53 et 54 montrent que la majorité de l'amélioration se situe durant la phase de simple appui. Cela permet de confirmer que le contrôleur est effectivement capable de contrôler l'équilibre du modèle autrement que par la position de pose du pied, et pendant une période du cycle où il a le moins de DoFs disponibles.

Le suivi des trajectoires du talon dans les directions antéropostérieure et verticale n'est que peu modifié par cette optimisation, ce qui est un résultat attendu. L'amélioration du suivi dans la direction médio-latérale est un résultat indirect du meilleur suivi du XCoM. En effet, dans la solution initiale après test d'optimisation, le talon décrit une courbe dans la direction médiale en début de phase oscillante avant de revenir vers sa trajectoire de référence. Ce phénomène n'est plus présent après optimisation du CC, ce qui facilite le suivi de la trajectoire par la suite et diminue l'écart moyen.

L'amélioration de l'angle de rotation du pelvis par rapport à sa référence est un point positif puisque cet angle contrôle l'orientation globale du modèle. En revanche, on observe que l'amélioration du suivi du XCoM s'est faite au détriment de la trajectoire de l'inclinaison sagittale du pelvis qui a été détériorée de 78% par rapport à la solution initiale après test d'optimisation.

## D. Optimisation du contrôle du talon

### 1. Fonction de coût

La fonction de coût utilisée pour cette étape est la suivante :

$$f = \text{Pénalité} + 10^3 * (\text{heelPos} + \text{heel } y_{RMSE} + \text{stepTimeDiff}) + 5 * 10^2 * (\text{heel } l_{RMSE} + \text{pelvis}_{RMSE} + \text{contactEnvol}_{RMSE}) + 10^2 *$$

Le terme *heelPos* est défini par :

$$\text{heelPos} = \frac{l_{\text{HeelHSxPos}} + l_{\text{HeelHSzPos}}}{2}$$

Les deux parties représentent la distance entre la position du talon gauche au moment du contact et la position cible déterminée par le contrôleur dans les deux directions horizontales. Cette position étant le point principal de l'un des aspects de l'équilibre, cet élément fait partie de ceux ayant le plus de poids dans la fonction.

Le terme *stepTimeDiff* représente la différence entre le temps de pas prescrit par le contrôleur et celui réalisé durant la simulation. Bien que le contrôleur soit pensé pour être robuste aux imprécisions de temps de pas, il est tout de même utile de minimiser cette différence pour améliorer le résultat du contrôle. De plus, la hauteur du talon lors du contact avec le sol étant toujours similaire, l'utilisation du point d'arrivée comme critère d'optimisation n'est pas utilisable. Cependant, un temps de pas similaire à celui de référence indique que le talon arrive au niveau du sol en même temps que la trajectoire de référence. Cet élément peut donc également être vu comme le pendant du terme précédent dans le sens où ce critère permet de s'assurer que la position du talon en fin de trajectoire est bonne. Le besoin de précision sur le temps de pas est jugé aussi important que la position du talon en fin de pas et a donc le même poids que le terme précédent.

Le terme *heel y<sub>RMSE</sub>* est l'écart quadratique moyen entre la trajectoire verticale de référence et de simulation du talon pendant la phase oscillante. Cet élément est ajouté car en n'utilisant que le critère sur le temps de pas, le suivi de la trajectoire n'est pas suffisamment précis et tend à être surélevé. Cependant, un problème pouvant survenir en suivant davantage la trajectoire verticale par rapport à une trajectoire surélevée est le risque de contact entre le pied et le sol lors du minimum local à environ 0.5 s sur notre mouvement simulé. Pour éviter ce problème, le critère doit donc avoir un poids suffisant pour s'assurer que le résultat final ne permette pas au talon de descendre plus bas que la trajectoire de référence.

Le terme *heel l<sub>RMSE</sub>* est défini par :

$$\text{heel } l_{RMSE} = \frac{\text{heel } x_{RMSE} + \text{heel } z_{RMSE}}{2}$$

Les deux parties sont identiques à l'élément précédent mais pour les directions horizontales. Lors d'une optimisation sans critère sur le suivi des trajectoires et avec un critère de minimisation du couple d'adduction-abduction de hanche, l'optimisation a convergé vers un résultat dans lequel les gains liés à l'adduction-abduction de hanche étaient presque nuls et avec une position d'arrivée médio-latérale du pied améliorée. Cependant, le tracé de la trajectoire du talon montrait que celle-ci ne suivait pas la

trajectoire de référence et arrivait à une position correcte par hasard plutôt que par un contrôle correct. Pour éviter ce phénomène, il a donc été décidé d'ajouter un critère sur le suivi des trajectoires de référence. Contrairement à la trajectoire verticale, le suivi des trajectoires horizontales est directement redondant avec le critère de position de pose du pied et n'a pas besoin d'être aussi précis pour éviter une situation à risque. De fait, le poids attribué à ce terme est moindre que les précédents mais moyen par rapport à l'ensemble des termes présents. Pour éviter cette redondance, il aurait également pu être possible de retirer le critère sur la pose du pied. Cependant, pour éviter une situation où le suivi global est correct mais pas en fin de la trajectoire, les deux termes sont gardés, ce qui équivaut à suivre les trajectoires globales avec un poids supérieur en fin de trajectoire.

Le terme  $contactEnvoI_{RMSE}$  représente la moyenne des forces de contact du pied en phase oscillante. Dans une situation idéale, cette valeur reste nulle. Cependant, après une optimisation dans laquelle les valeurs de gains étaient minimisées, un contact était présent sur une partie du pas même avec une hauteur de talon respectant la trajectoire de référence. Cet élément a donc été introduit pour pénaliser les solutions dans lesquelles ce phénomène se produit.

Le terme *Gains* est le pendant de celui présent lors de l'optimisation du CC mais pour le SFC. Il est également calculé comme la moyenne des gains du SFC et est introduit pour les mêmes raisons.

Les termes restants sont repris de la fonction de coût de l'étape précédente. Le terme  $pelvis_{RMSE}$  est présent pour les mêmes raisons que lors de l'étape précédente, à savoir pour éviter que l'amélioration du contrôle du talon ne se fasse au détriment des trajectoires du pelvis et a donc un poids moyen dans la fonction. Quant aux éléments concernant le suivi du CoM, ils sont présents pour éviter que la solution n'altère trop le résultat de l'optimisation précédente. Leur priorité est donc moindre et le poids qui leur est appliqué dans la fonction est donc également plus faible.

## 2. Résultats

Les résultats de cette étape d'optimisation sont présentés dans les figures 56, 57 et 58. L'évolution des critères utilisés pour quantifier le résultat du contrôleur est présentée dans la table 6.

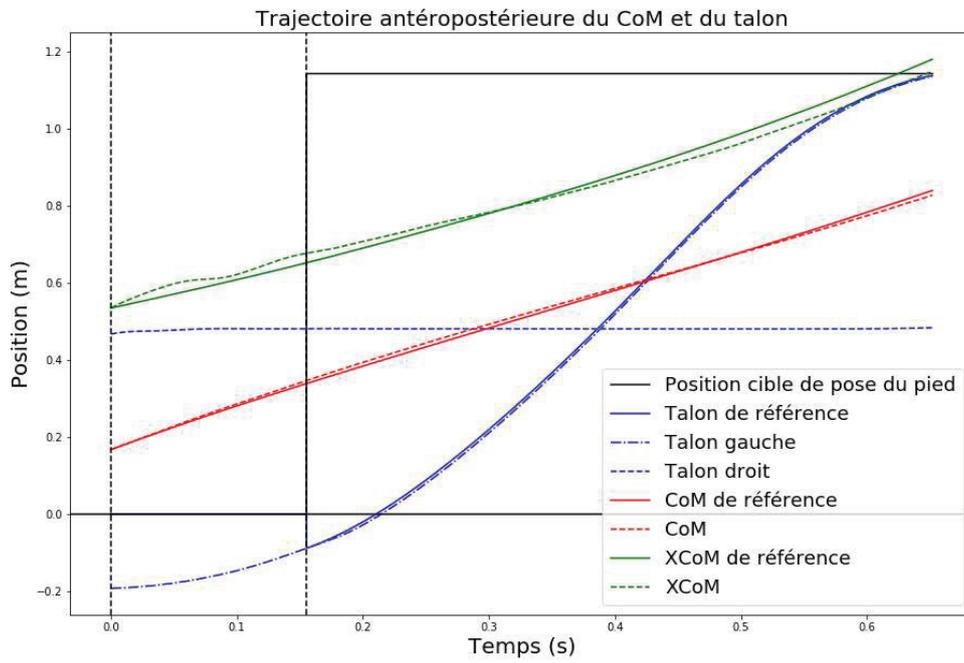


Figure 56 : Trajectoire antéropostérieure du CoM et des talons lors du premier pas après optimisation du SFC

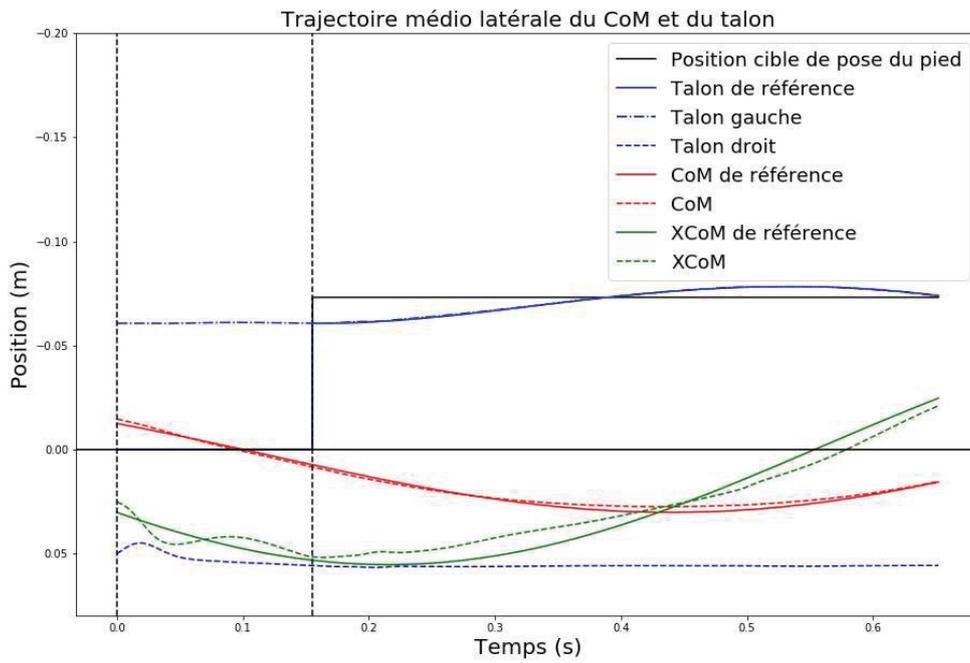


Figure 57 : Trajectoire médio-latérale du CoM et des talons lors du premier pas après optimisation du SFC

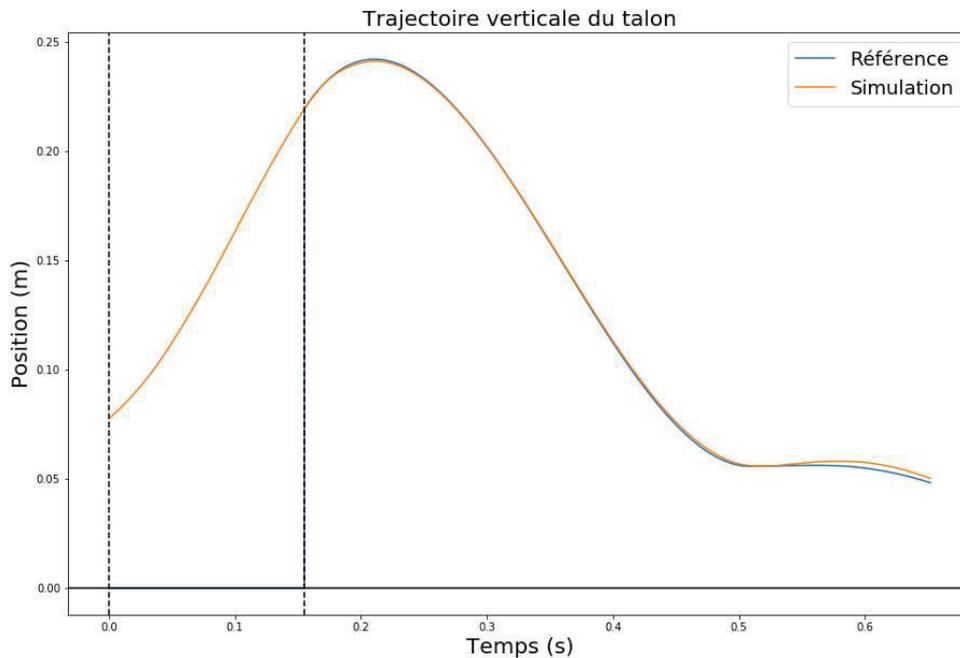


Figure 58 : Trajectoire verticale du talon gauche lors du premier pas après optimisation du SFC

Différence (m) / (m) / (s)	Position d'arrivée x	Position d'arrivée z	Temps de pas
Optimisation CC	0.048	0.001	2.174
Optimisation SFC	0.002	0.0002	0.206

RMSE (m)	Talon x	Talon y	Talon z	XCoMx	XCoMz
Optimisation CC	0.027	0.013	0.002	0.013	0.006
Optimisation SFC	0.006	0.001	0.0002	0.021	0.005

RMSE (°)	Inclinaison sagittale du pelvis	Inclinaison frontale du pelvis	Pelvis Rot
Optimisation CC	2.51	1.05	3.05
Optimisation SFC	1.135	0.9	1.96

Table 6 : Evolution des RMSE avant et après optimisation du CC. (Les différences de valeur pour l'optimisation CC viennent du fait que la simulation est faite sur un temps plus long pour permettre à la solution initiale de réaliser un pas complet)

La table 6 montre que les différences entre position de pose et trajectoire du talon ont diminué d'un ordre de grandeur dans les trois directions et sont désormais de l'ordre du millimètre ou moins. La

différence de position de pose du pied dans la direction médio-latérale étant déjà de l'ordre du millimètre avant l'optimisation, il n'est pas attendu que l'amélioration de cette valeur impacte fortement l'équilibre du modèle. En revanche, l'amélioration du suivi de la trajectoire dans cette direction permet d'affirmer que ce résultat est dû au contrôle. La différence de temps de pas a également diminué d'un ordre de grandeur en passant de plus de deux secondes à deux dixièmes de seconde, limitant le besoin de recourir aux adaptations en cas de mauvais temps de pas. La figure 58 montre que la trajectoire verticale du talon ne passe pas sous la trajectoire de référence au moment critique évoqué précédemment.

Les conséquences de l'optimisation du SFC sur le contrôle du CoM sont minimales dans la direction médio-latérale mais plus visible dans la direction antéropostérieure. La raison principale de cet effet est que la hanche en phase de simple appui est simultanément utilisée par le CC et le SFC. Optimiser ces briques indépendamment induit donc une diminution de l'efficacité de la brique non optimisée et démontre la nécessité de terminer par une étape optimisant toutes les briques de contrôle ensemble.

Enfin, on observe que les trajectoires du pelvis ont été améliorées après cette étape d'optimisation, notamment les inclinaisons et la rotation du pelvis.

### 3. Test du résultat sur deux pas

Avant de passer à l'étape suivante, il est intéressant de tenter une simulation sur deux pas avec les paramètres optimisés sur un pas, afin de déterminer la robustesse du résultat des premières étapes et les éventuels problèmes présents lors du second pas pour orienter la définition de la fonction de coût. Les résultats sont présentés dans les figures 59, 60, 61, 62 et 63.

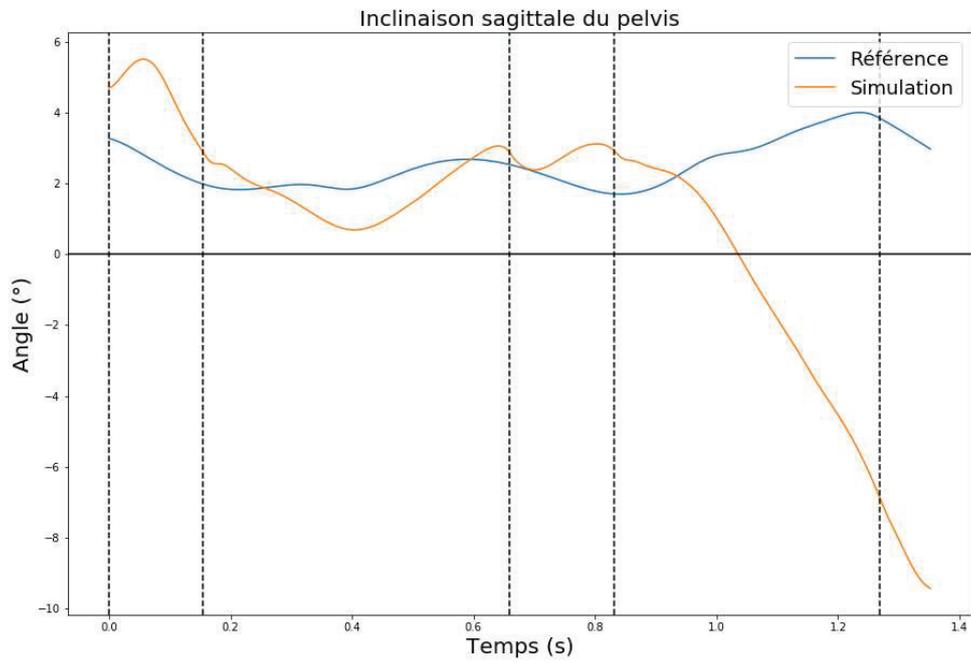


Figure 59 : Angle de l'inclinaison sagittale lors du premier pas après optimisation du SFC

La figure 59 montre que la trajectoire du pelvis est fortement altérée à partir de la phase de simple appui du second pas. Etant donné la gravité de l'altération, le suivi des trajectoires du pelvis doit être prioritaire dans la fonction de coût suivante.

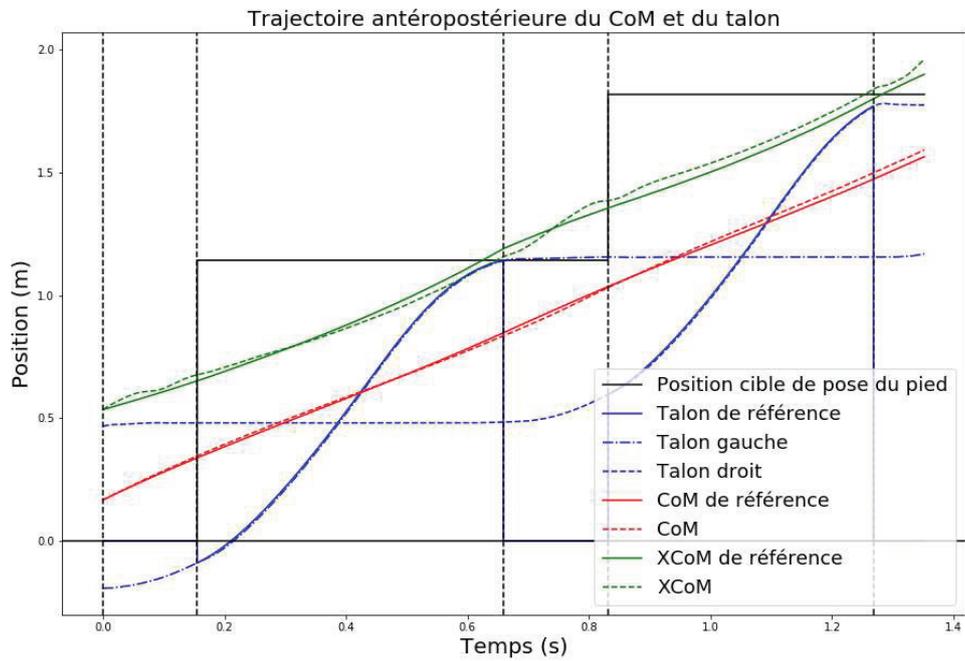


Figure 60 : Trajectoire antéropostérieure du CoM et des talons lors d'une foulée après optimisation du SFC

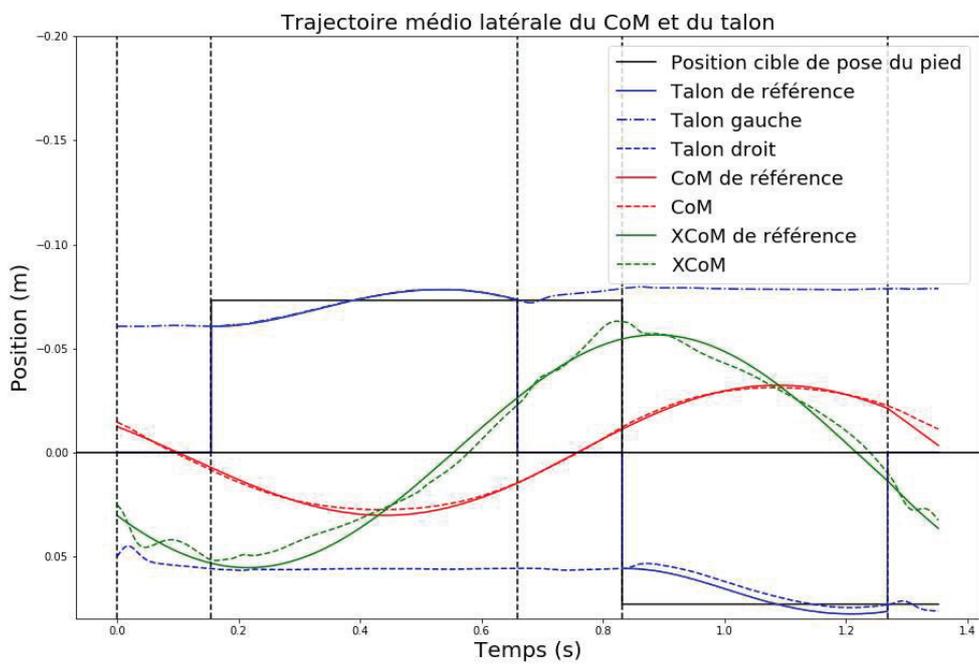


Figure 61 : Trajectoire médio-latérale du CoM et des talons lors d'une foulée après optimisation du SFC

La figure 60 montre que le suivi du XCoM antéropostérieur est légèrement altéré lors du second pas. En revanche, le suivi de la trajectoire du talon semble rester aussi précis lors du deuxième pas que lors du premier. Le critère de suivi du XCoM devra donc avoir une priorité plus importante que le suivi de la trajectoire antéropostérieure du talon. La figure montre également que la trajectoire de référence n'a pas atteint la position cible à la fin du second pas, indiquant que le pied s'est posé trop tôt.

La figure 61 montre un résultat opposé dans la direction médio latérale par rapport à la direction antéropostérieure puisque le suivi du XCoM semble aussi précis lors du second pas mais pas le suivi de la trajectoire du talon.

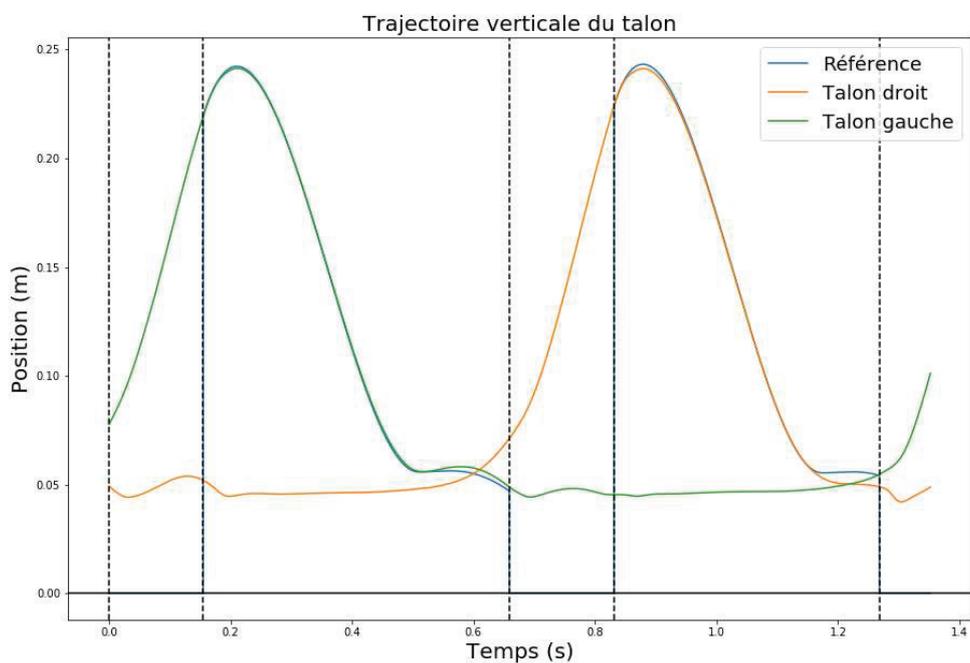


Figure 62 : Trajectoire verticale des talons lors d'une foulée après optimisation du SFC

La figure 62 montre que la trajectoire verticale du talon ne remonte pas après le minimum local lors du second pas. Cette erreur de suivi provoque la pose prématurée du pied. Le suivi de cette trajectoire devra donc être un critère prioritaire lors de l'étape suivante.

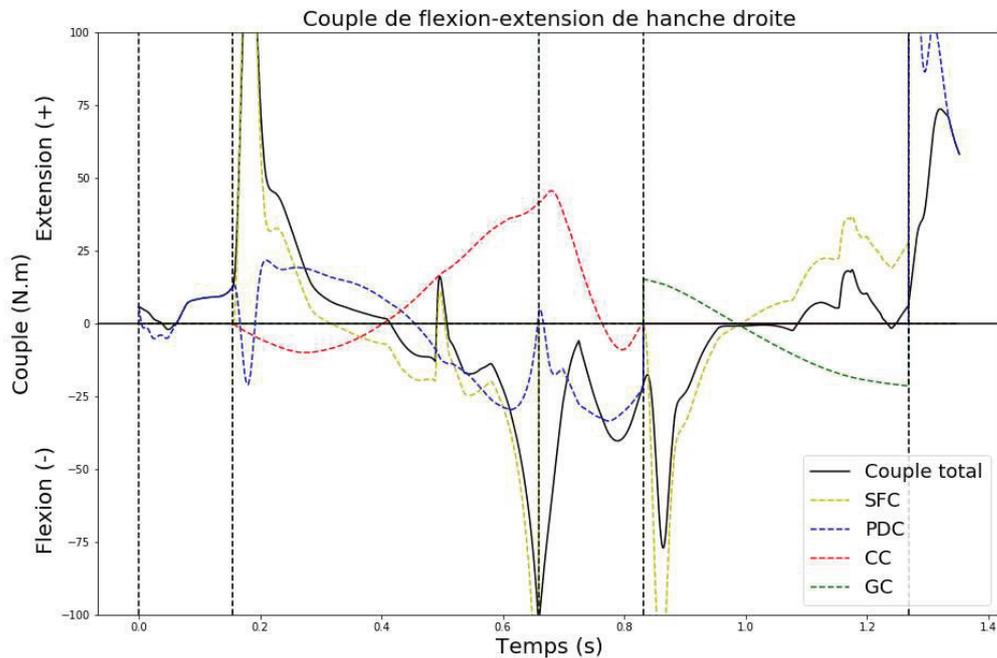


Figure 63 : Couple de flexion-extension de hanche droite lors d'une foulée après optimisation du SFC

La figure 63 montre un point améliorable via le contrôle des gains. Les couples de flexion-extension de la hanche controlatérale issus du SFC provoquent des pics de couple en début de phase de simple appui et oscillante. Ces pics sont explicables par le fait que la priorité est mise sur le suivi de trajectoires de référence. Même en minimisant les gains, ils restent suffisamment élevés pour tenter de rattraper fortement les erreurs de suivi s'accumulant en début de phase lorsque le contrôle est encore en phase d'adaptation entre les phases. Il pourra donc être intéressant de minimiser en particulier les gains du SFC provoquant ces pics pour déterminer s'ils ont un impact sur le suivi de l'inclinaison sagittale du pelvis.

## E. Optimisation globale

Au vu des résultats précédents, la brique du CC est considérée comme ayant le plus besoin d'optimisation. En effet, le suivi des trajectoires du talon semble aussi correct lors du deuxième pas que lors du premier, à l'exception de la fin de phase oscillante du membre inférieur droit dans la direction verticale. Les DoFs contrôlant la hauteur du talon contrôlant également son avancée, l'un des rôles de l'optimisation était justement de déterminer des valeurs de gains capables de suivre simultanément les trajectoires dans ces deux directions. Le suivi de la trajectoire antéropostérieure étant le même lors du second pas que lors du premier, il est possible que le problème de suivi dans la direction verticale soit dû aux problèmes de suivi du CC qui impactent le SFC.

Cependant, un gain du SFC sera tout de même optimisé, à savoir le premier gain utilisé pour déterminer l'action de la hanche controlatérale pour le suivi de la position antéropostérieure du talon. Ce gain est responsable du pic de couple présenté dans la figure 63. Or, l'inclinaison sagittale du pelvis étant à corriger en priorité, il peut être intéressant de minimiser ce pic en début de phase oscillante puisque la flexion-extension de hanche influe sur l'inclinaison sagittale du pelvis.

Initialement, les deux briques de contrôle devaient être optimisées simultanément en profitant du résultat des optimisations précédentes pour définir un domaine de recherche réduit. Cependant, au vu des résultats, et notamment de ceux du pelvis, il a été décidé de profiter de la précision de la brique du SFC pour garder un domaine de recherche étendu pour l'optimisation du CC.

### 1. Fonction de coût

La fonction de coût utilisée pour cette étape est la suivante :

$$f = \text{Pénalité} + 10^3 * (XCoM_{RMSE} + Pelvis_{RMSE}) + 5 * 10^2 * (heelPos + heel_{RMSE} + Tdiff) + gain_{CC} + gain_{SFC}$$

Les deux éléments prioritaires lors de cette étape sont le suivi du XCoM et des trajectoires du pelvis pour les raisons évoquées précédemment. Les critères concernant le suivi des trajectoires du talon sont également gardés puisque l'hypothèse faite est que le problème de suivi de trajectoire du talon est en partie dû au CC. L'élément concernant les gains du CC est le même que celui utilisé lors de l'optimisation du CC et l'élément concernant les gains du SFC est adapté pour ne prendre en compte que le gain optimisé afin de le minimiser.

## 2. Résultats

### 2.1. Equilibre

L'évolution des résultats et des critères pour quantifier les résultats de cette étape d'optimisation sont présentés dans les figures 64, 65, 66, 67, 68 et 69 et la table 7. Un snapshot du résultat est également présenté (Figure 102)

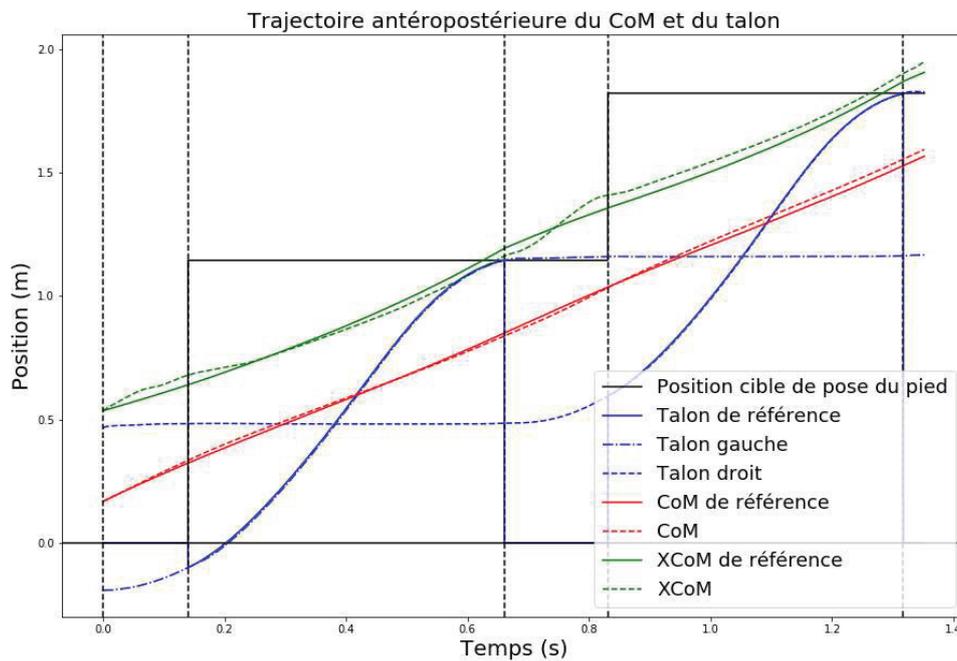


Figure 64 : Trajectoire antéropostérieure du CoM et des talons lors d'une foulée après optimisation globale

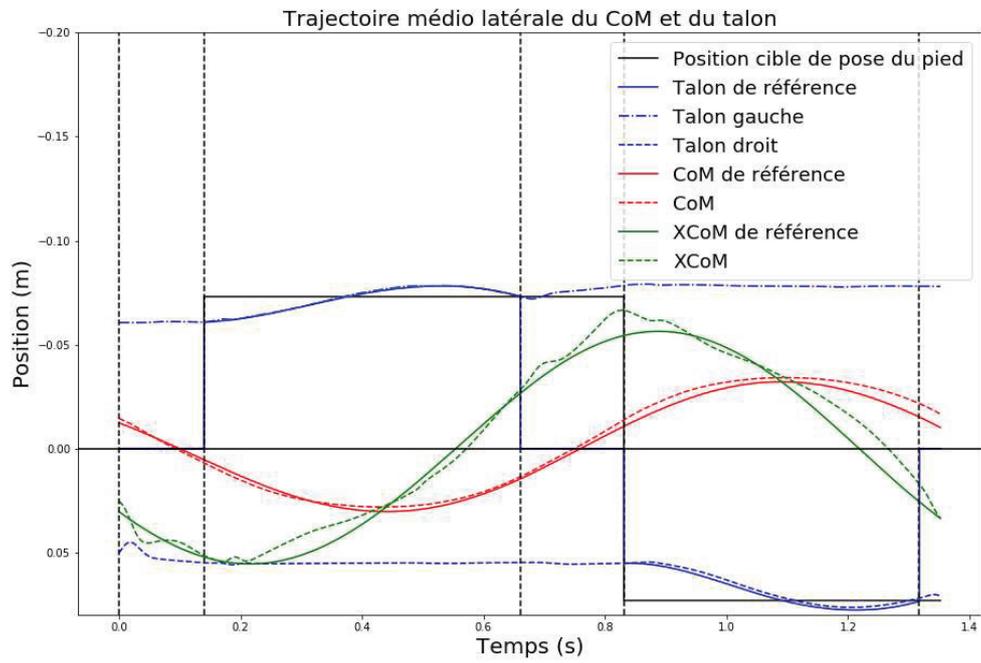


Figure 65 : Trajectoire médio-latérale du CoM et des talons lors d'une foulée après optimisation globale

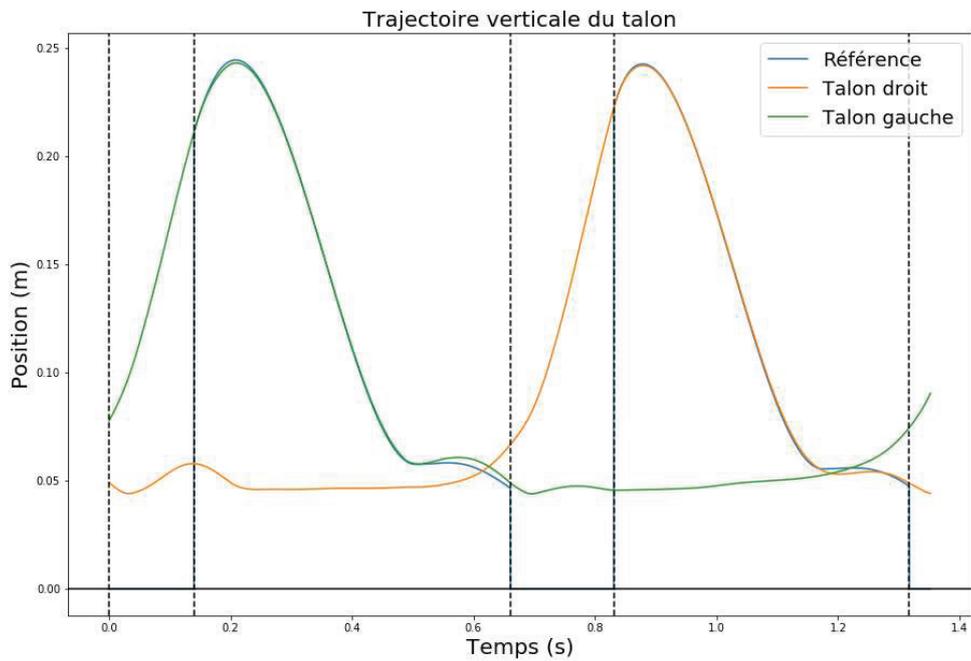


Figure 66 : Trajectoire verticale des talons lors d'une foulée après optimisation globale

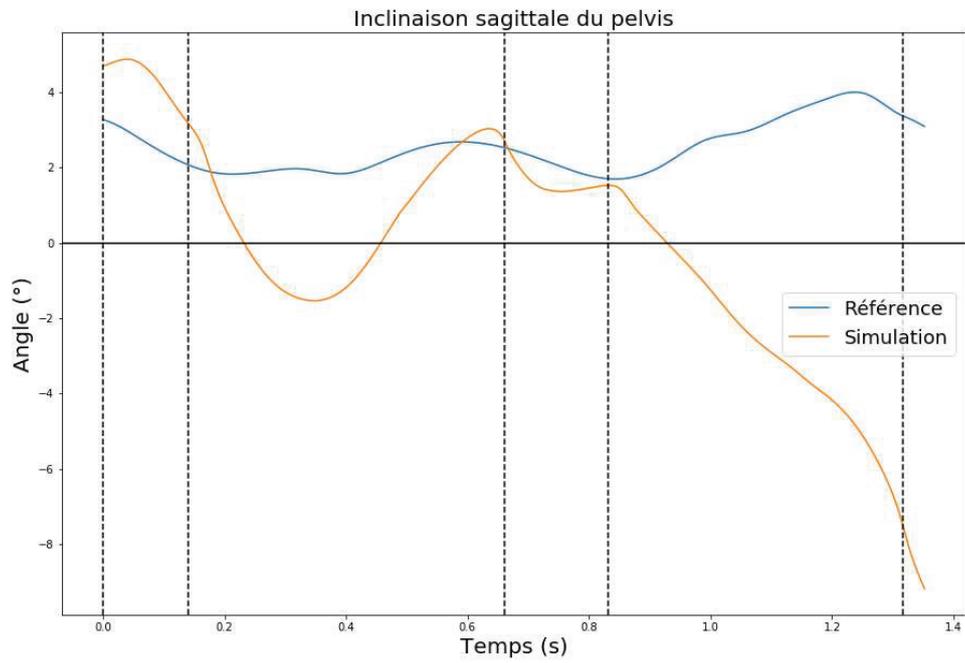


Figure 67 : Angle de l'inclinaison sagittale du pelvis lors d'une foulée après optimisation globale

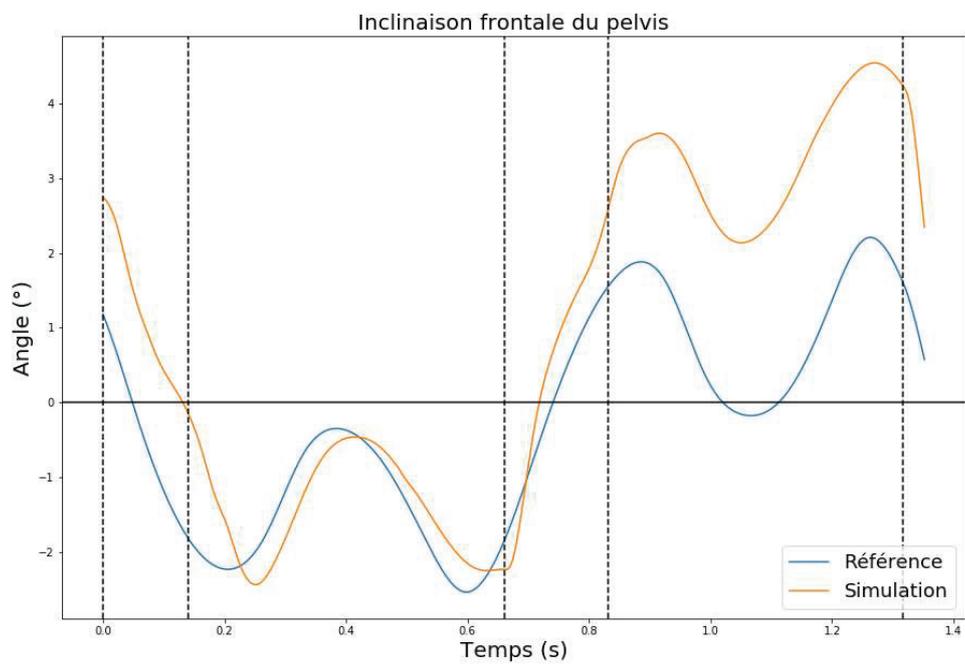


Figure 68 : Angle de l'inclinaison frontale du pelvis lors d'une foulée après optimisation globale

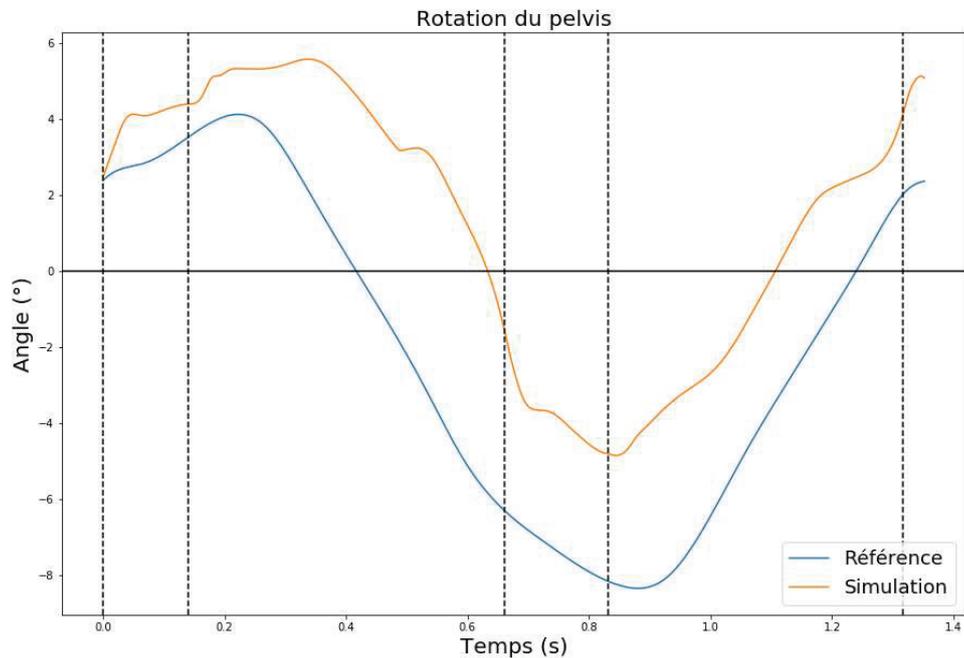


Figure 69 : Angle de la rotation du pelvis lors d'une foulée après optimisation globale

RMSE (m)	XCoMx	XCoMz	Talon x	Talon y	Talon z
Optimisation CC	0.028	0.005	0.006	0.002	0.002
Optimisation SFC	0.031	0.006	0.005	0.001	0.001

RMSE (°)	Inclinaison sagittale du pelvis	Inclinaison frontale du pelvis	Pelvis Rot
Optimisation CC	4.289	4.66	3.364
Optimisation SFC	4.408	1.574	3.673

Différence (m) / (m) / (s)	Position d'arrivée x		Position d'arrivée z		Temps de pas	
	Gauche	Droite	Gauche	Droite	Gauche	Droite
Optimisation CC	0.002	0.05	0.0002	0.0003	0.206	3.977
Optimisation SFC	1.00E-05	0.0001	2*10E-6	0.001	0.085	0.373

Table 7 : Evolution des RMSE avant et après optimisation

Les résultats montrent qu'après optimisation, la pose du pied droit est plus proche de la référence. L'amélioration la plus nette vient de la trajectoire de l'inclinaison frontale du pelvis qui est

près de trois fois mieux suivie qu'avant optimisation. Il est possible que l'amélioration dans le suivi de la trajectoire verticale du talon soit un résultat indirect de cette amélioration. En revanche, les suivis du XCoM et de l'inclinaison sagittale du pelvis n'ont pas été améliorés et ont même été légèrement dégradés. La seule optimisation du CC n'est donc pas suffisante pour corriger les problèmes d'angle d'inclinaison sagittale du pelvis et de suivi du XCoM simultanément. Dans sa forme actuelle, elle semble avoir privilégié une solution permettant d'améliorer le suivi du pas plutôt que les autres problèmes.

## 2.2. Flexion-extension de hanche

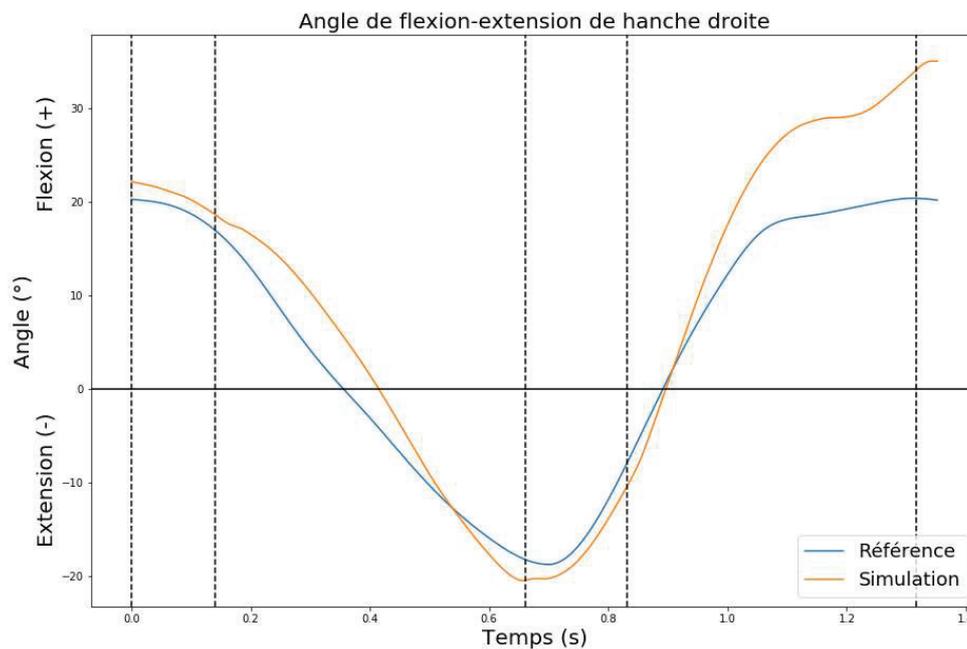


Figure 70 : Angle de flexion-extension de hanche droite lors d'une foulée après optimisation globale

La figure 70 montre que la trajectoire de la flexion de hanche droite s'éloigne peu de sa trajectoire de référence lors des trois premières phases (premier double appui, simple appui, deuxième double appui) puisqu'en partie contrôlée par le PDC. En revanche, elle s'éloigne fortement lors de la quatrième phase (oscillante). Cela est dû à l'inclinaison sagittale du pelvis qui impose une flexion de hanche plus importante, relativement au pelvis, pour positionner le talon au bon endroit.

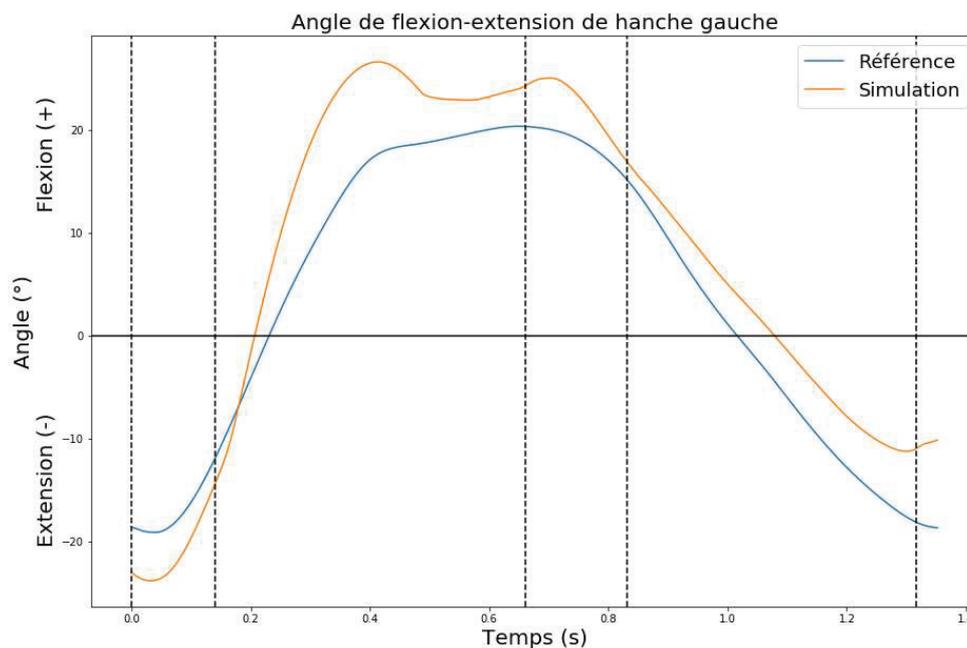


Figure 71 : Angle de flexion-extension de hanche gauche lors d'une foulée après optimisation globale

La figure 71 montre que la trajectoire de la flexion de hanche gauche s'éloigne de sa trajectoire de référence pendant la deuxième phase, (phase oscillante) lorsque le DoF est contrôlé par le SFC. Par la suite, le PDC tend à le ramener vers la trajectoire de référence mais n'y parvient pas et diverge en fin de phase de simple appui, provoquant une inclinaison sagittale du pelvis trop importante.

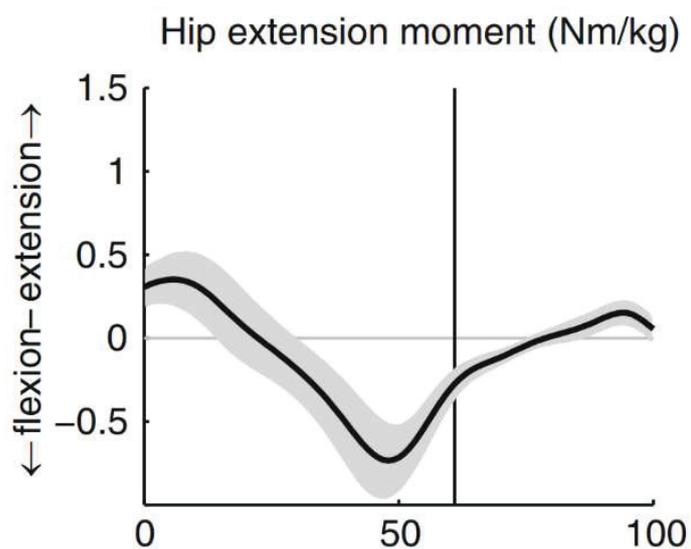


Figure 72 : Couple de flexion-extension dans la littérature (Sloot & van der Krogt, 2018)

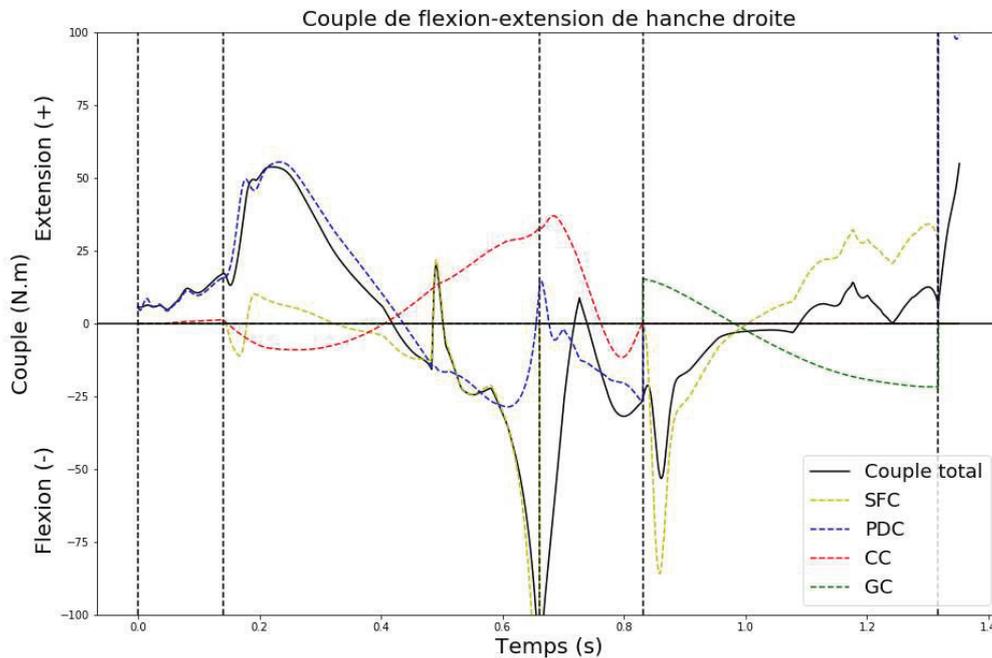


Figure 73 : Couple de flexion-extension de hanche droite lors d'une foulée après optimisation globale

La figure 73 montre que la minimisation du gain du SFC lié à ce DoF a effectivement permis de supprimer le pic en début de deuxième phase. Bien que le passage d'un couple d'extension à un couple de flexion pendant la phase de simple appui soit attendu, la valeur de flexion résultant du SFC en fin de phase est trop élevée par rapport à ce qui est attendu. En fin de troisième phase, (second double appui) le couple de flexion retrouve une valeur attendue mais après être passé par une valeur nulle. Ce passage est à la fois dû au CC et au PDC qui tendent à compenser la flexion trop importante provoquée par le pic de SFC en fin de deuxième phase (simple appui). Lors de la quatrième phase, (oscillante) outre le pic de couple du SFC, la trajectoire du couple a la forme et les valeurs attendues. Le couple lors de la première phase (premier simple appui) ne peut être utilisé comme critère d'évaluation car le modèle sort alors de sa phase d'initialisation dans lequel le contrôle de la phase oscillante de la hanche est différent de celui du reste de la simulation. La transition de phase n'est donc pas représentative du contrôle. Enfin, le pic de couple de SFC à environ 0.5 secondes est dû à un changement soudain de la vitesse de référence du talon de la jambe gauche. Ce phénomène se retrouvant dans différentes figures, il sera abordé de manière subséquente, de même que les pics et oscillations en début de phase de simple appui et oscillante.

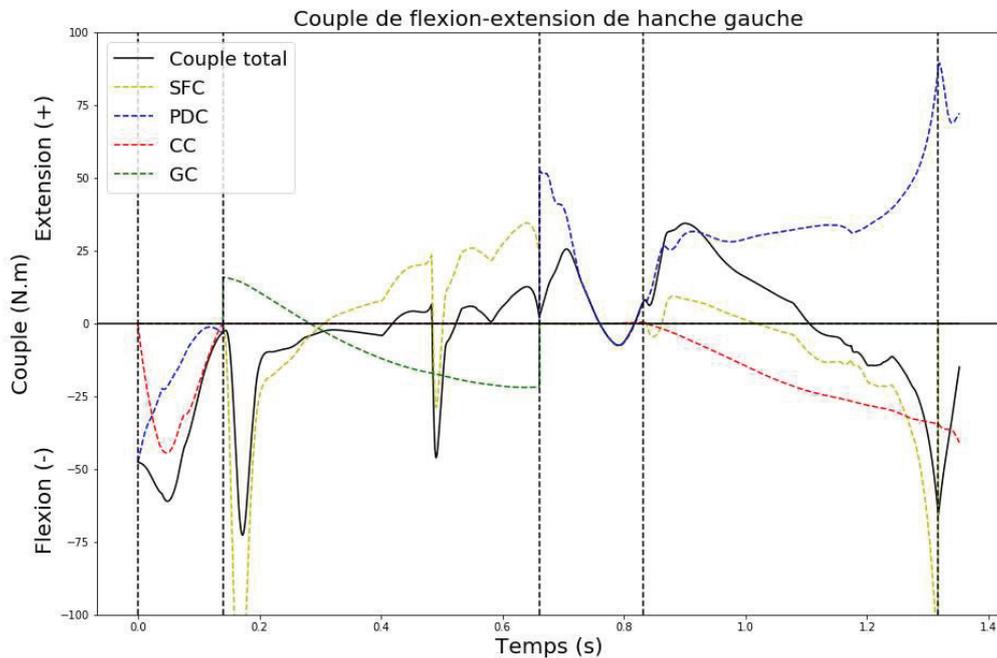


Figure 74 : Couple de flexion-extension de hanche gauche lors d'une foulée après optimisation globale

La figure 74 montre que le couple lors des phases oscillante et de simple appui ont globalement les valeurs attendues (Figure 72), mais pas en phase de double appui. Le premier pic en début de deuxième phase (oscillante) est dû au fait que la flexion en fin de première phase (second double appui) est trop faible. L'avancée du pied à ce moment-là est donc insuffisante et le SFC compense alors en prescrivant un couple plus important. Après ce premier pic, la trajectoire du couple est celle attendue à l'exception d'un pic à environ 0.5 seconde pour la même raison que citée précédemment. Lors de la troisième phase (premier double appui), le couple passe par une valeur nulle au lieu de rester en extension mais reprend une trajectoire attendue en quatrième phase (simple appui).

### 2.3. Adduction-adduction de hanche

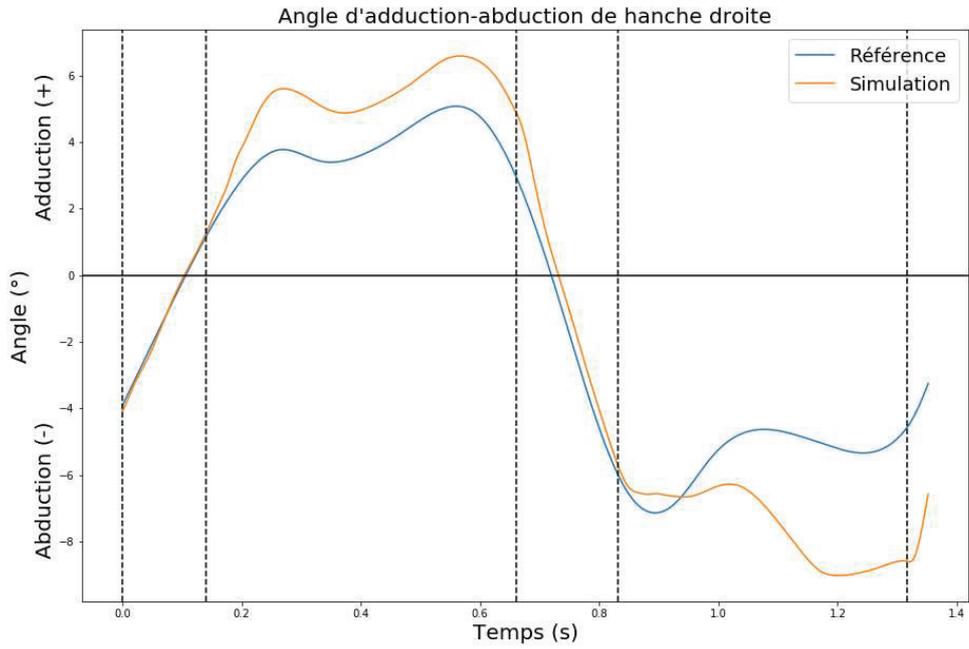


Figure 75 : Angle d'adduction-abduction de hanche droite lors d'une foulée après optimisation globale

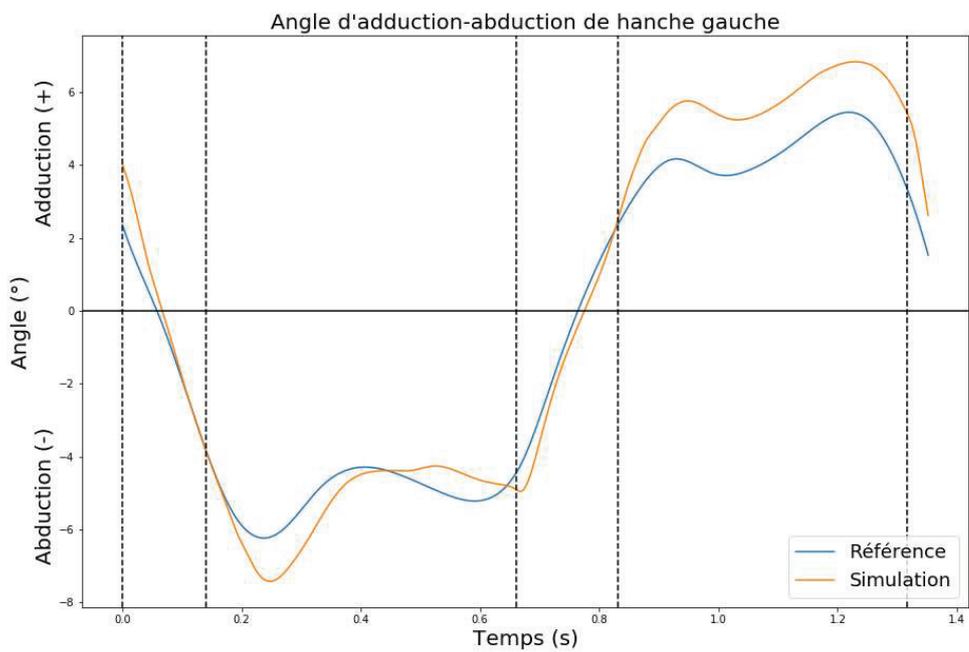


Figure 76 : Angle d'adduction-abduction de hanche gauche lors d'une foulée après optimisation globale

Les figures 75 et 76 montrent qu'en phase d'appui, l'adduction-abduction de hanche respecte la trajectoire de référence. Lors de la phase de simple appui, l'adduction-abduction est légèrement plus élevée que la référence. Cela est dû au poids des membres supérieurs et inférieurs contralatéraux exerçant un couple sur l'adduction de hanche. Avec un gain plus élevé, l'écart serait moindre mais cela compliquerait les adaptations des autres briques de contrôle. Lors de la phase oscillante de la jambe gauche (phase 2), la trajectoire est fidèle à la trajectoire de référence alors que le contrôle n'est pas directement fait pour, indiquant que le contrôle avec ce paramétrage est capable de produire indirectement un mouvement biofidèle. En revanche, la trajectoire de la hanche droite en phase oscillante (phase 4) n'est, elle, pas fidèle à la trajectoire de référence. Le paramétrage étant correct pour le premier pas, il est probable que cette différence soit due à l'inclinaison frontale du pelvis qui ne suit plus sa trajectoire de référence lors du deuxième pas, forçant la hanche à s'éloigner de la trajectoire de référence pour contrôler le talon.

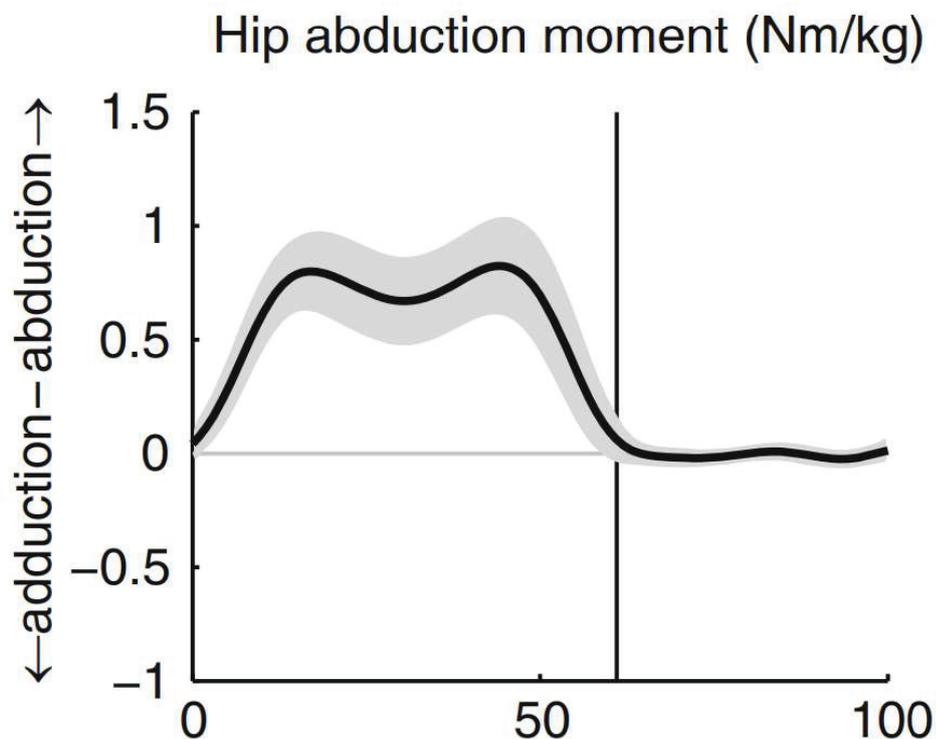


Figure 77 : Couple d'adduction-abduction dans la littérature (Sloot & van der Krogt, 2018)

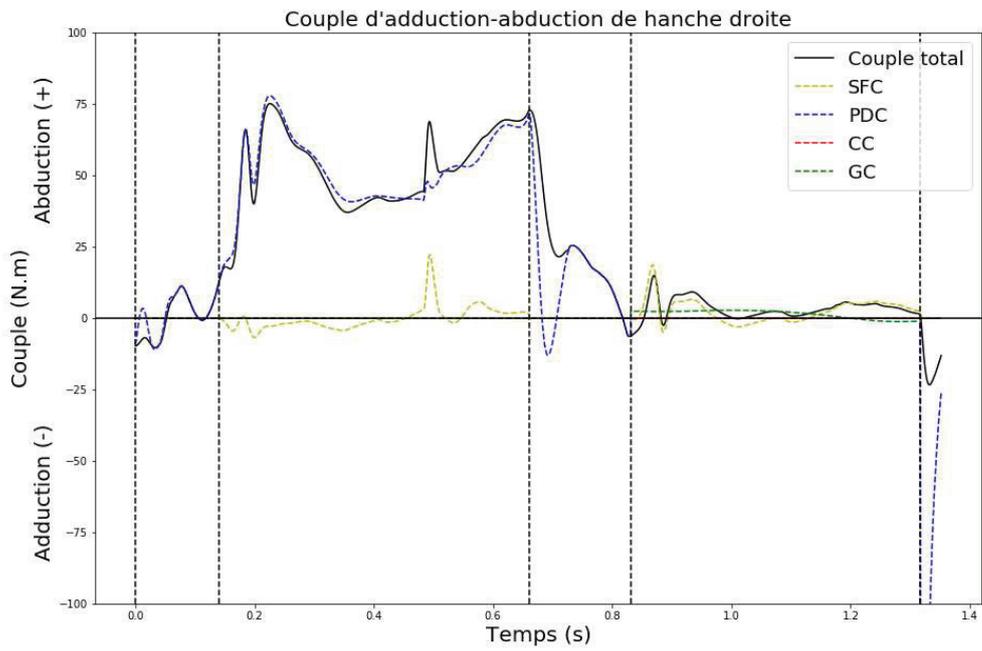


Figure 78 : Couple d'adduction-abduction de hanche droite lors d'une foulée après optimisation globale

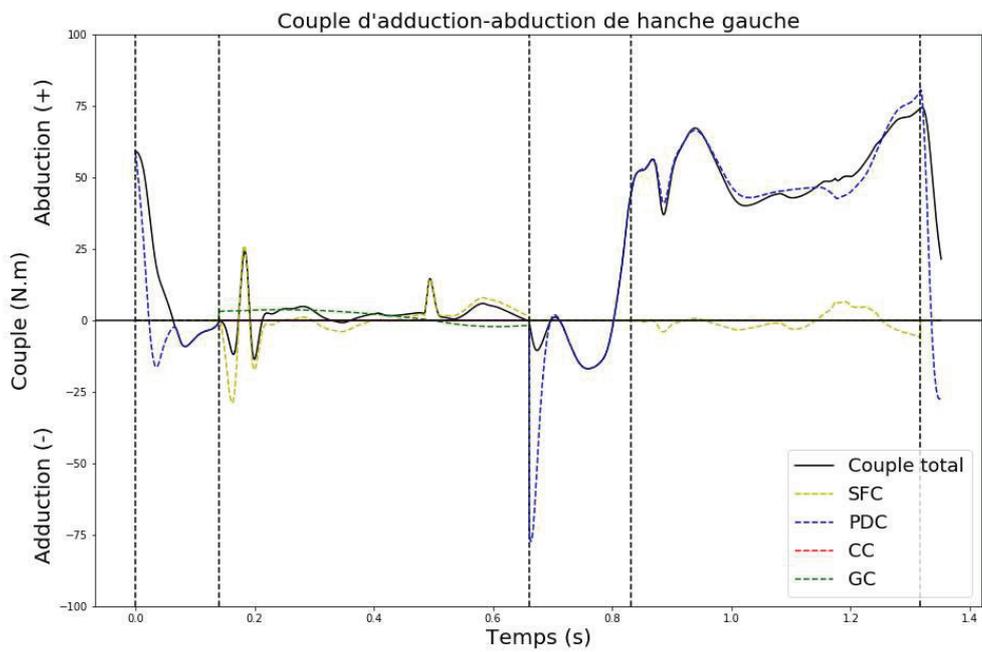


Figure 79 : Couple d'adduction-abduction de hanche gauche lors d'une foulée après optimisation globale

Les figures 78 et 79 montrent un comportement globalement attendu (Figure 77) pour les deux pas avec un plateau de couple d'abduction en phase de simple appui et un couple proche de zéro en phase oscillante. Une limite est toutefois visible pour le couple gauche lors de la troisième phase (premier double appui dans le cycle des membres gauche). En effet, le couple d'abduction est censé augmenter dès le début de la phase de double appui, ce qui n'est pas le cas ici. Cela peut être la preuve d'une imperfection dans le transfert de poids d'une jambe à l'autre. De plus, des pics de couples peuvent être observés au milieu du simple appui (à environ 0.5 seconde) et en début de phase oscillante comme pour la flexion de hanche.

#### 2.4. Rotation interne-externe de hanche

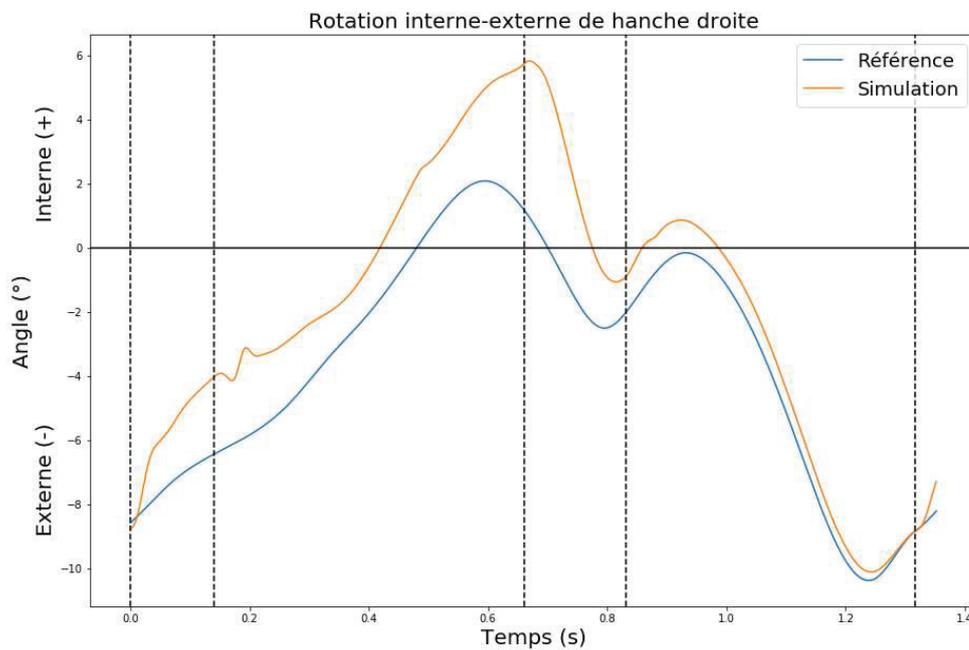


Figure 80 : Angle de rotation interne-externe de hanche droite lors d'une foulée après optimisation globale

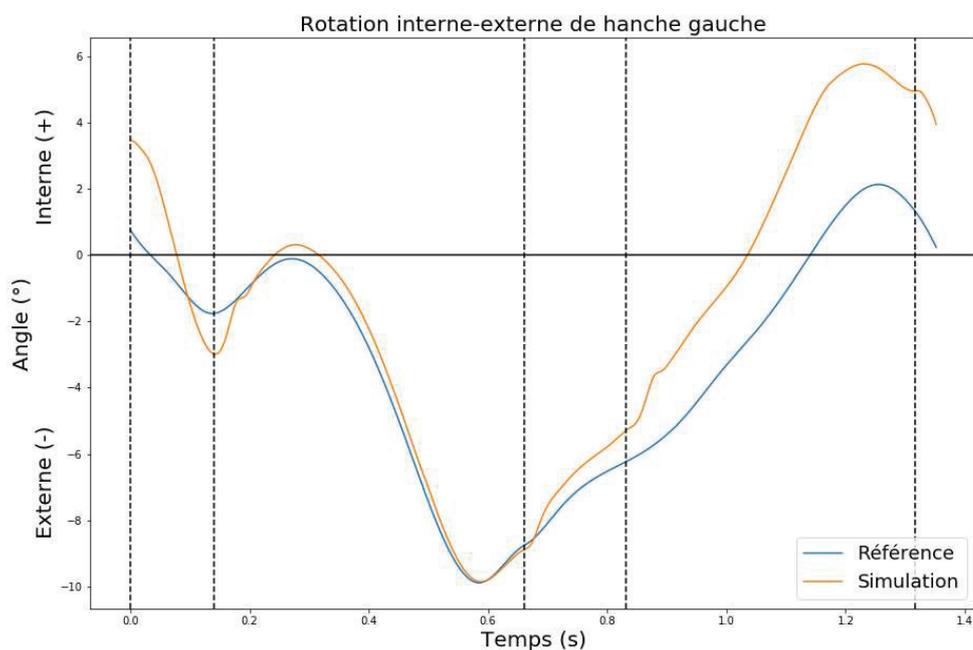


Figure 81 : Angle de rotation interne-externe de hanche gauche lors d'une foulée après optimisation globale

Les figures 80 et 81 montrent un comportement similaire pour la rotation interne-externe de hanche des côtés droit et gauche. En phase d'appui, le DoF tend à s'éloigner de sa trajectoire de référence tout en gardant une forme similaire. Au contraire, en phase oscillante de chaque jambe (phase 2 pour la jambe droite et phase 4 pour la jambe gauche), le DoF parvient à rejoindre la trajectoire de référence avant la pose du pied.

Les figures 82 et 83 montrent des pics de couples importants dus au SFC au début des phases d'appui de chaque jambe. Bien que d'amplitudes supérieures, ces pics sont issus du même phénomène que ceux présents aux flexion-extension et adduction-abduction de hanche. Outre ces pics, l'ordre de grandeur des valeurs des couples est celui attendu.

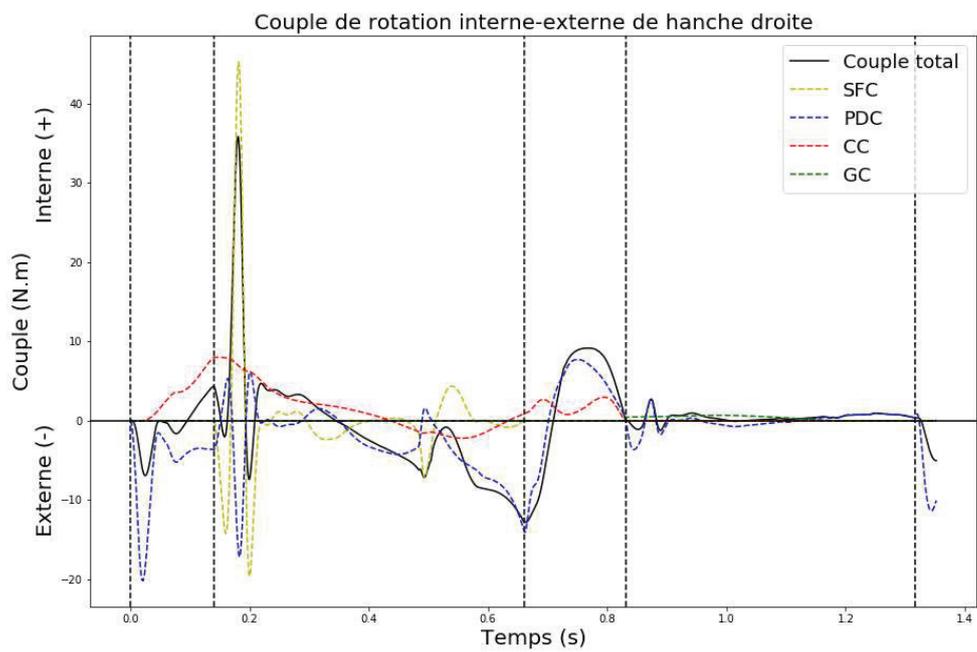


Figure 82 : Couple de rotation interne-externe de hanche droite lors d'une foulée après optimisation globale

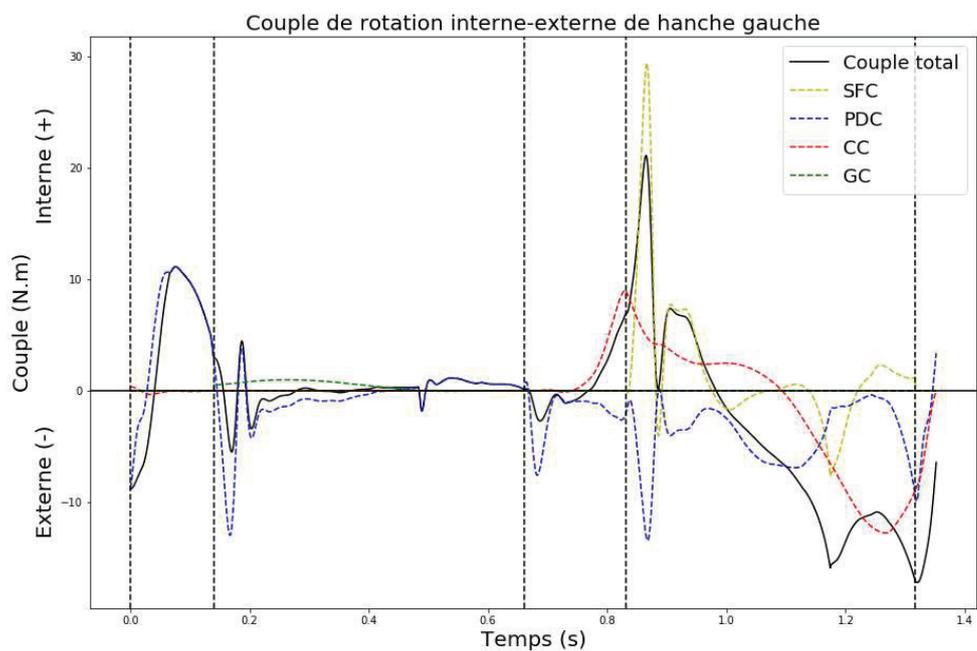


Figure 83 : Couple de rotation interne-externe de hanche gauche lors d'une foulée après optimisation globale

## 2.5. Flexion-extension de genou

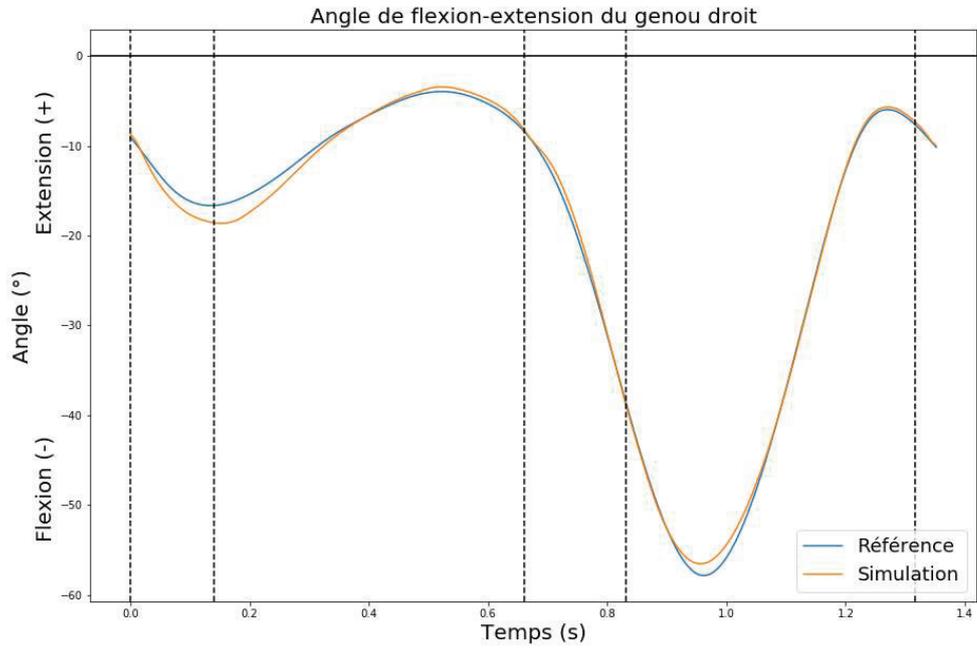


Figure 84 : Angle de flexion-extension de genou droit lors d'une foulée après optimisation globale

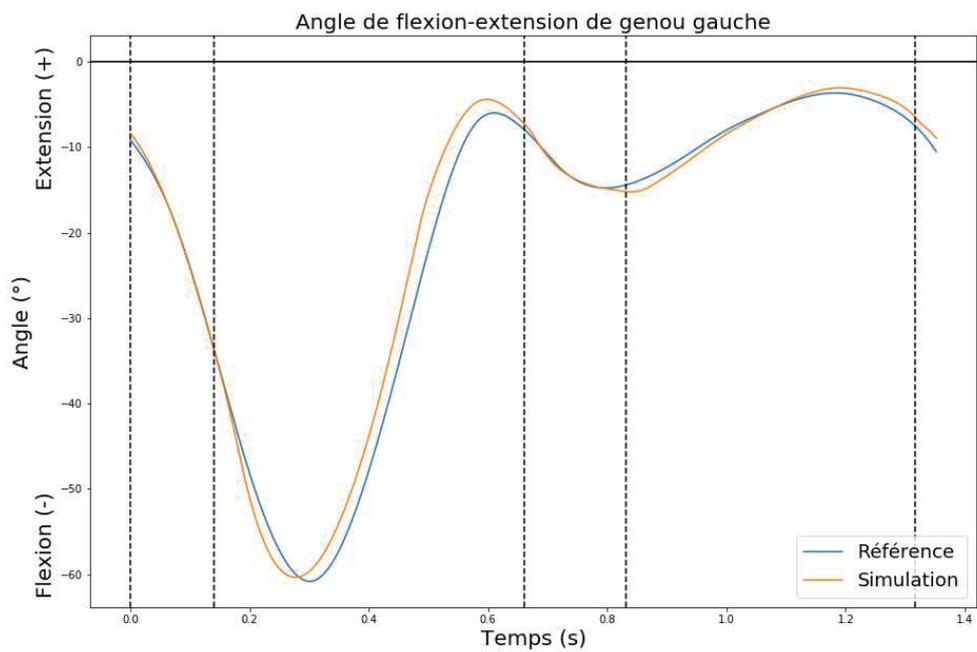


Figure 85 : Angle de flexion-extension de genou gauche lors d'une foulée après optimisation globale

Le seul moment du cycle de marche où le genou n'est pas contrôlé par le PDC est le début de phase oscillante. Les figures 84 et 85 montrent que même sans contrôle direct, les trajectoires de simulation restent fidèles aux trajectoires de référence à ce moment-là.

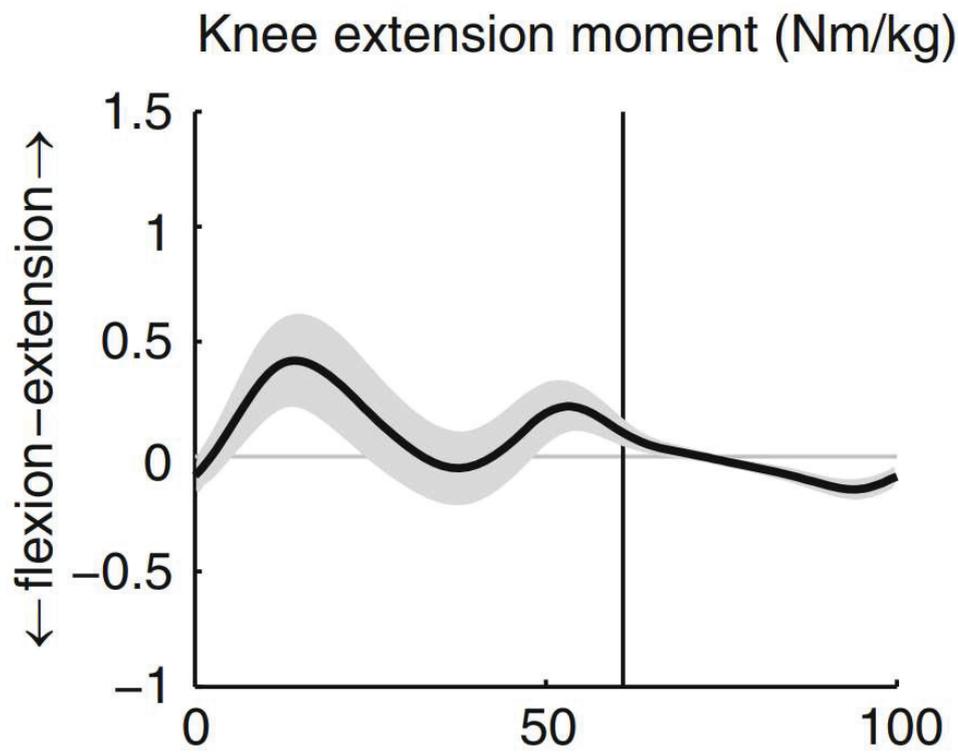


Figure 86 : Couple de flexion-extension de genou dans la littérature (Sloot & van der Krogt, 2018)

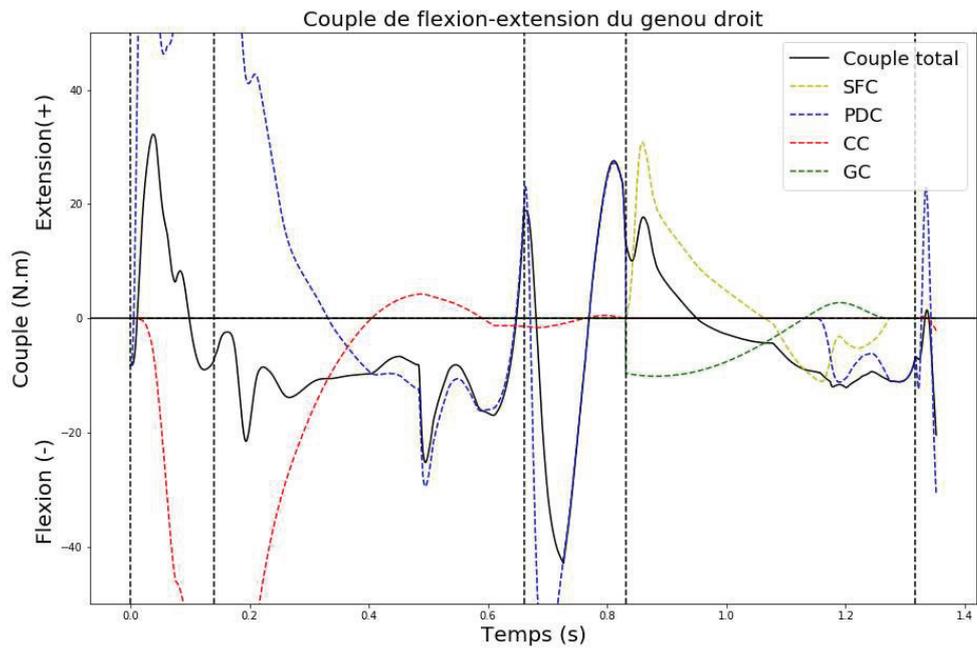


Figure 87 : Couple de flexion-extension de genou droit lors d'une foulée après optimisation globale

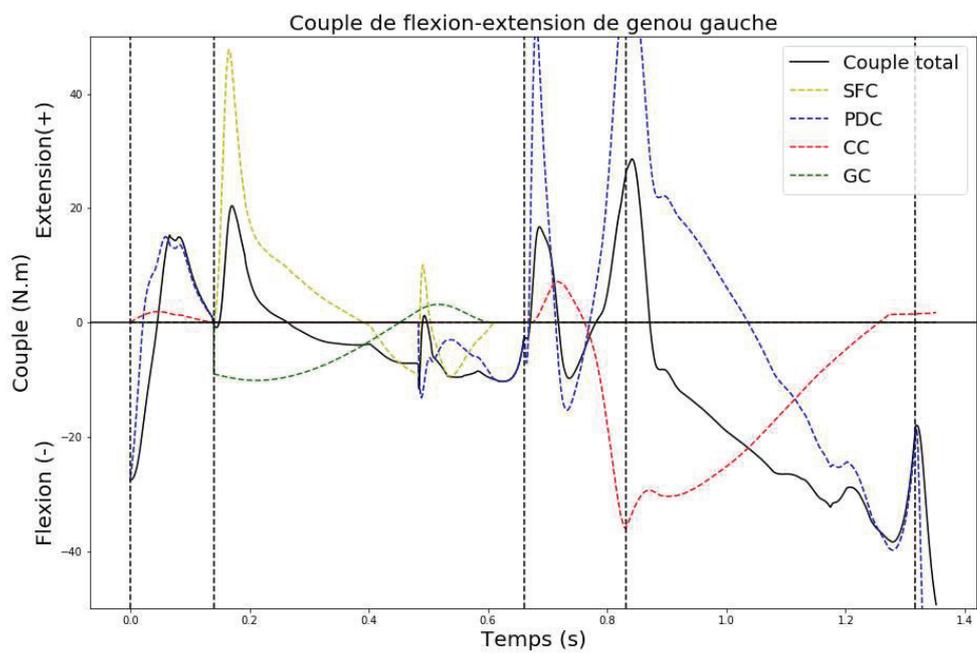


Figure 88 : Couple de flexion-extension de genou gauche lors d'une foulée après optimisation globale

La figure 87 montre que la forme générale et les valeurs du couple de flexion-extension du genou respectent ce qui est attendu (Figure 86), à l'exception du moment de la pose du pied gauche où le couple passe temporairement en extension au lieu de rester en flexion. Cela semble correspondre au pic de flexion de la hanche droite en fin de deuxième phase (simple appui) et il est probable que cette extension en soit le résultat. En effet, un couple de flexion de hanche à ce moment aura tendance à envoyer le genou vers l'avant. Le pied étant posé au sol, cela se traduira par une flexion du genou qui sera compensée par le PDC via un couple d'extension. Les couples du genou semblent également produire plus d'oscillations que les couples des hanches. Concernant le couple de flexion-extension du genou gauche, la figure 88 montre que sa forme et ses valeurs lors de la deuxième phase (oscillante) sont celles attendues, mis à part les pics en début de phase et à 0.5 seconde pour les mêmes raisons que la flexion de hanche. Cependant, pendant la troisième phase (premier double appui), au lieu de produire un couple d'extension de plus en plus élevé, le genou oscille et repasse par une phase de flexion avant de produire le bon couple au passage de cette phase à la phase d'envol de la jambe droite. Enfin, la flexion lors de cette quatrième phase est trop importante par rapport à ce qui est attendu. Ce résultat peut toutefois être également lié au problème d'orientation du pelvis lors de cette phase qui peut perturber le reste de la cinématique.

## 2.6. Dorsi-planti flexion de cheville

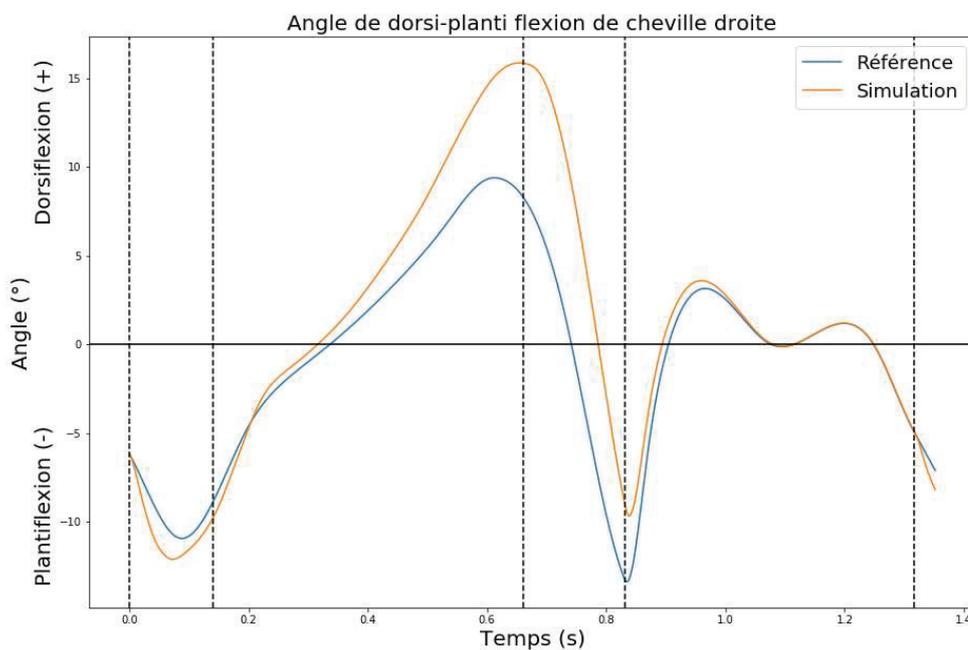


Figure 89 : Angle de dorsi-planti flexion de cheville droite lors d'une foulée après optimisation globale

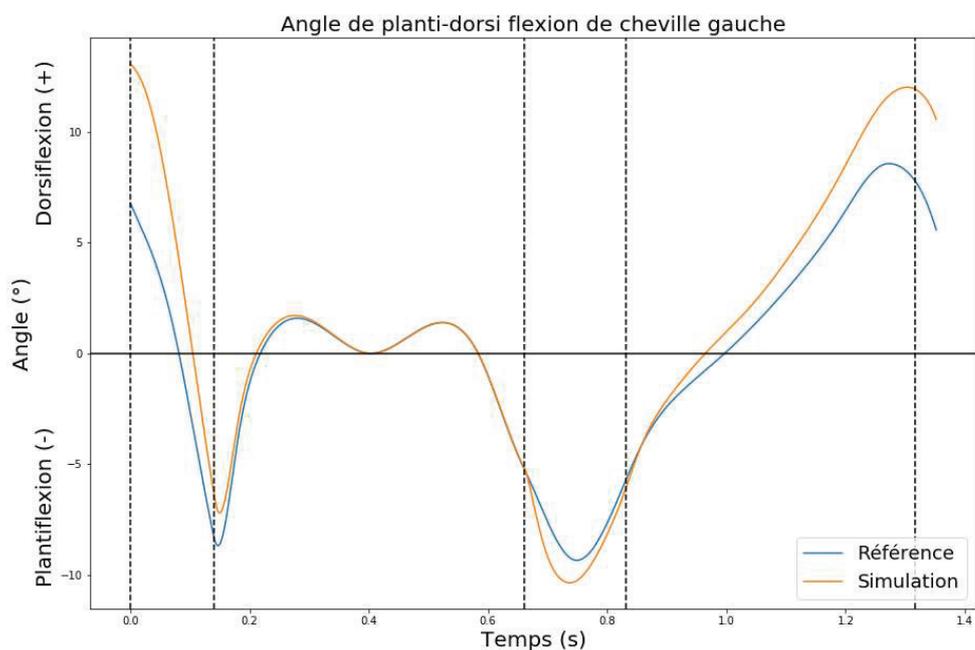


Figure 90 : Angle de dorsi-planti flexion de cheville gauche lors d'une foulée après optimisation globale

Les figures 89 et 90 montrent que les angles de dorsi-planti flexion de cheville suivent globalement leurs trajectoires de référence mais ont tendance à produire plus de dorsiflexion en fin de phase de simple appui de chaque jambe (phase 2 pour la jambe droite et phase 4 pour la jambe gauche), lorsque la cheville a pour rôle de propulser le corps vers l'avant. Cela peut signifier que le pied ne roule pas assez sur le sol et que le talon ne se lève pas assez en fin de ces phases de simple appui. Une autre explication possible vient de la simplification du segment du pied en un segment rigide. Puisque le segment du modèle est indéformable alors que le pied l'est, pour un même angle entre le pied et le sol, la cheville ne se situe pas à la même hauteur dans la réalité ou la simulation. La différence d'angle de cheville entre celui de simulation et celui de référence lors de la propulsion est alors potentiellement une adaptation due au contrôleur pour compenser la hauteur différente de cheville.

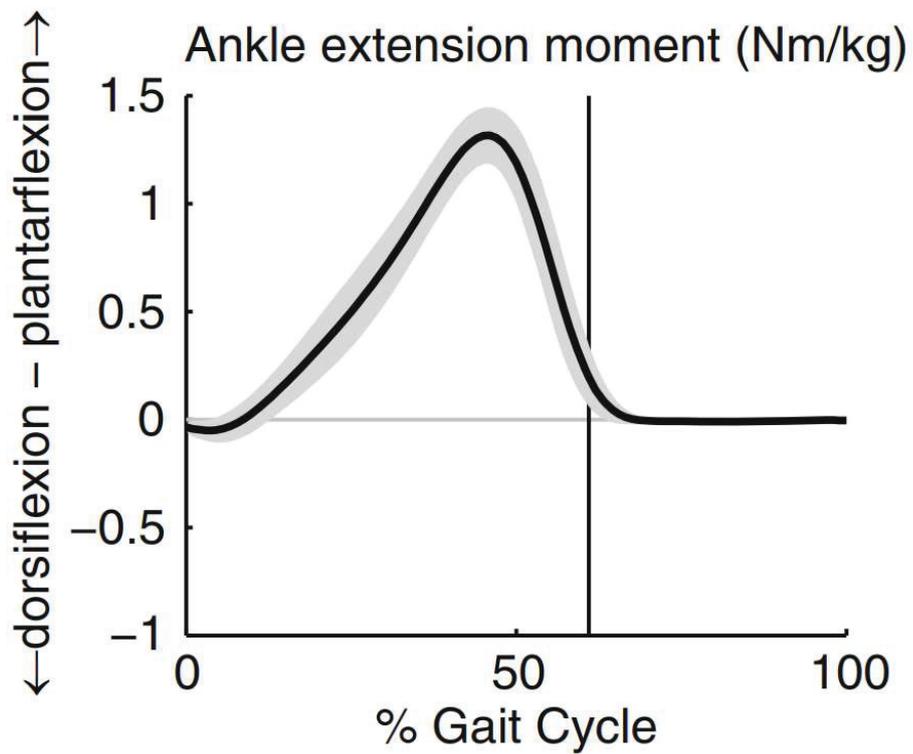


Figure 91 : Couple de dorsi-planti flexion de cheville dans la littérature (Sloot & van der Krogt, 2018)

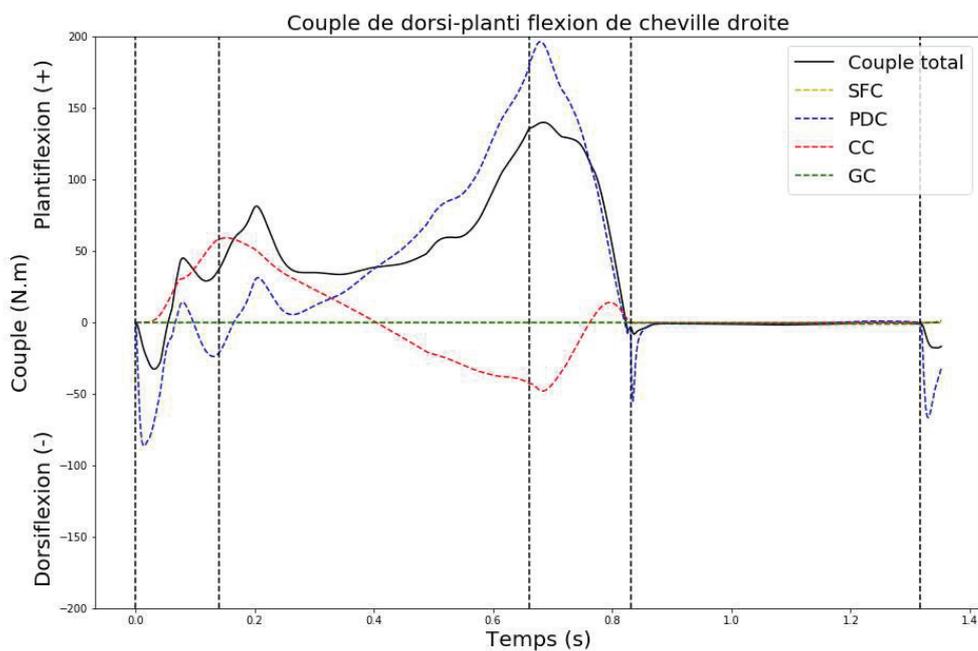


Figure 92 : Couple de dorsi-planti flexion de cheville droite lors d'une foulée après optimisation globale

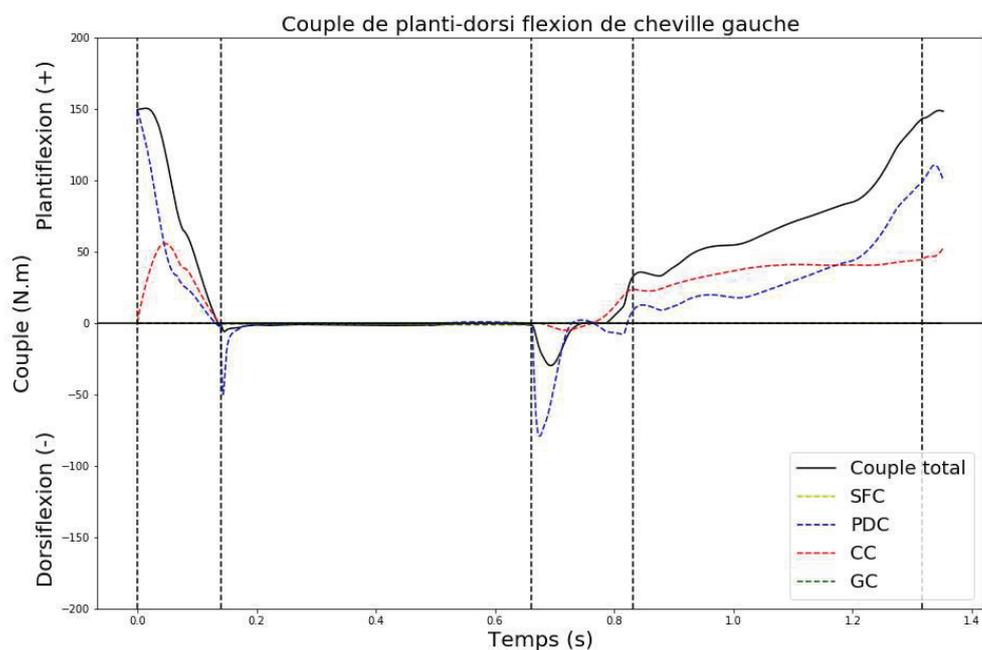


Figure 93 : Couple de dorsi-planti flexion de cheville gauche lors d'une foulée après optimisation globale

La figure 92 montre que le couple de dorsi-planti flexion de cheville droite ne suit pas complètement un comportement attendu (Figure 91). En effet, au lieu de produire progressivement un couple de plantiflexion tout au long de la phase de simple appui, elle produit un fort couple de plantiflexion en début de phase. Ce surplus est dû au CC qui tend à produire un couple de plantiflexion en début de pas pour freiner le CoM. Comme le montre la figure 93, ce phénomène n'est pas présent au pied gauche où le couple produit suit une forme et des valeurs attendues. Ce surplus de plantiflexion est alors potentiellement dû aux conditions de début de premier pas puisque celui-ci suit directement la phase d'initialisation.

## 2.7. Inversion-éversion de subtalaire

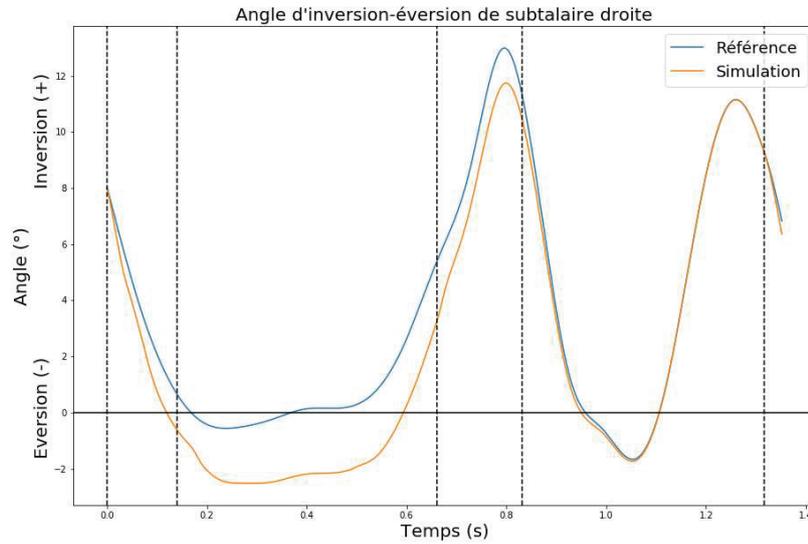


Figure 94 : Angle d'inversion-éversion de subtalaire droite lors d'une foulée après optimisation globale

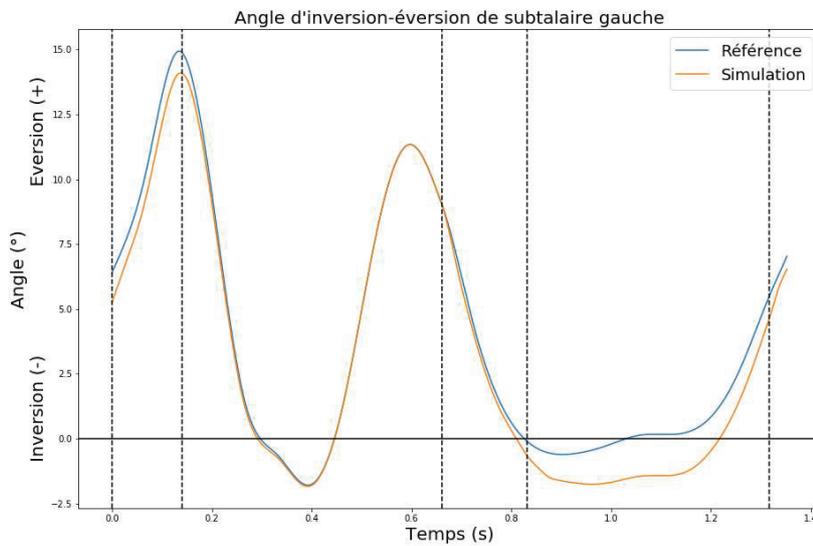


Figure 95 : Angle d'inversion-éversion de subtalaire gauche lors d'une foulée après optimisation globale

Les figures 94 et 95 montrent que, de manière similaire à l'adduction-adduction de hanche, les DoF suivent correctement leurs trajectoires de référence pendant les phases oscillantes (phase 4 pour la jambe droite et phase 2 pour la jambe gauche) et tendent à en dévier en phase d'appui tout en gardant la même forme.

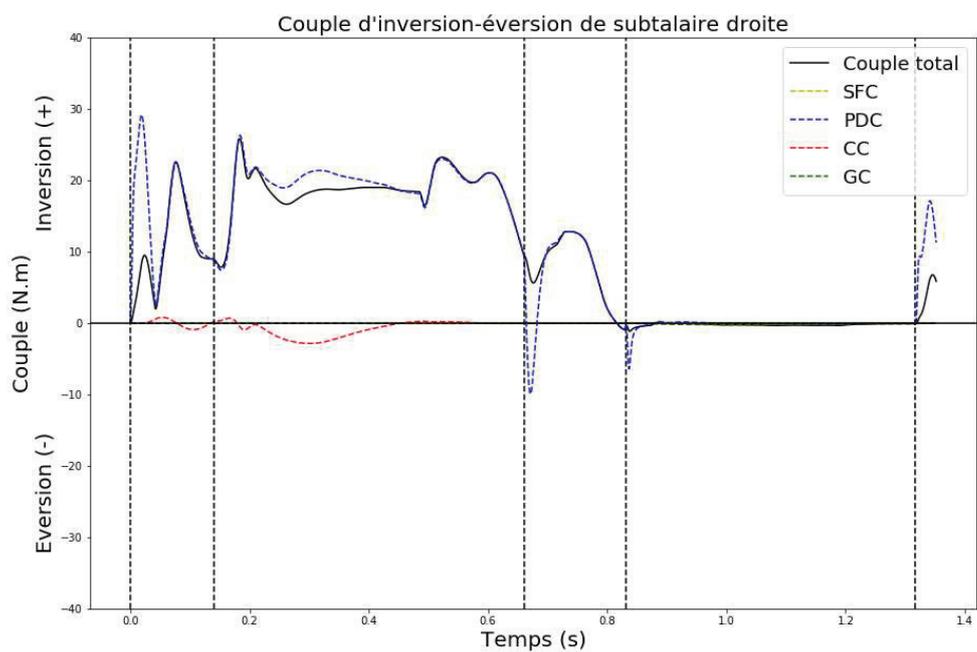


Figure 96 : Couple d'inversion-éversion de subtalaire droite lors d'une foulée après optimisation globale

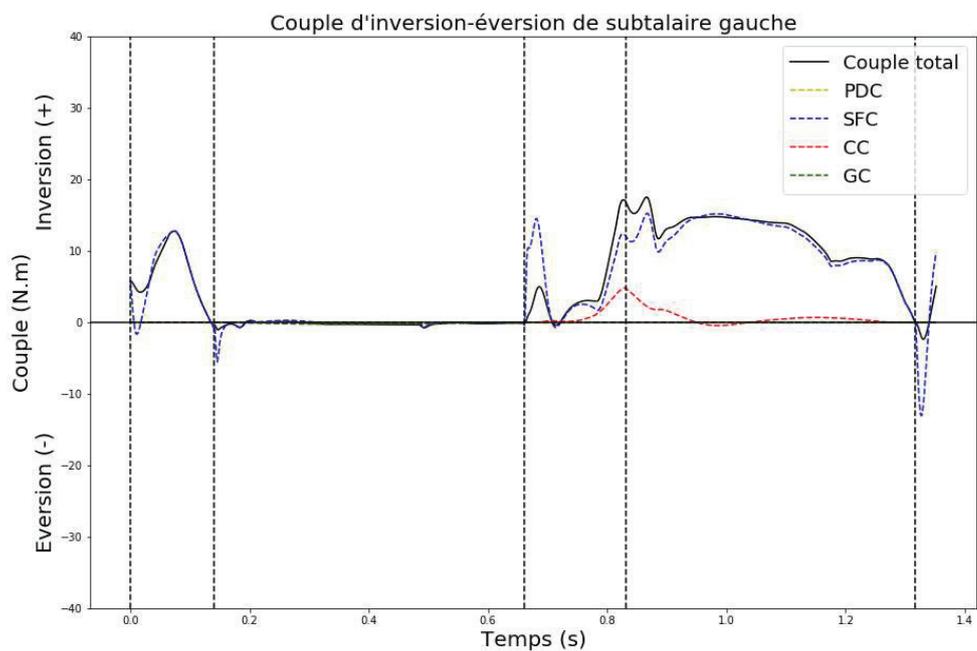


Figure 97 : Couple d'inversion-éversion de subtalaire gauche lors d'une foulée après optimisation globale

Les couples de l'articulation subtalaire telle que décrite dans les modèles OpenSim ne sont pas couramment étudiés et il est donc difficile de comparer les résultats à la littérature. L'on peut cependant noter que les figures 96 et 97 montrent que les courbes semblent similaires entre les côtés droit et gauche.

## 2.8. Forces de réaction antéropostérieures

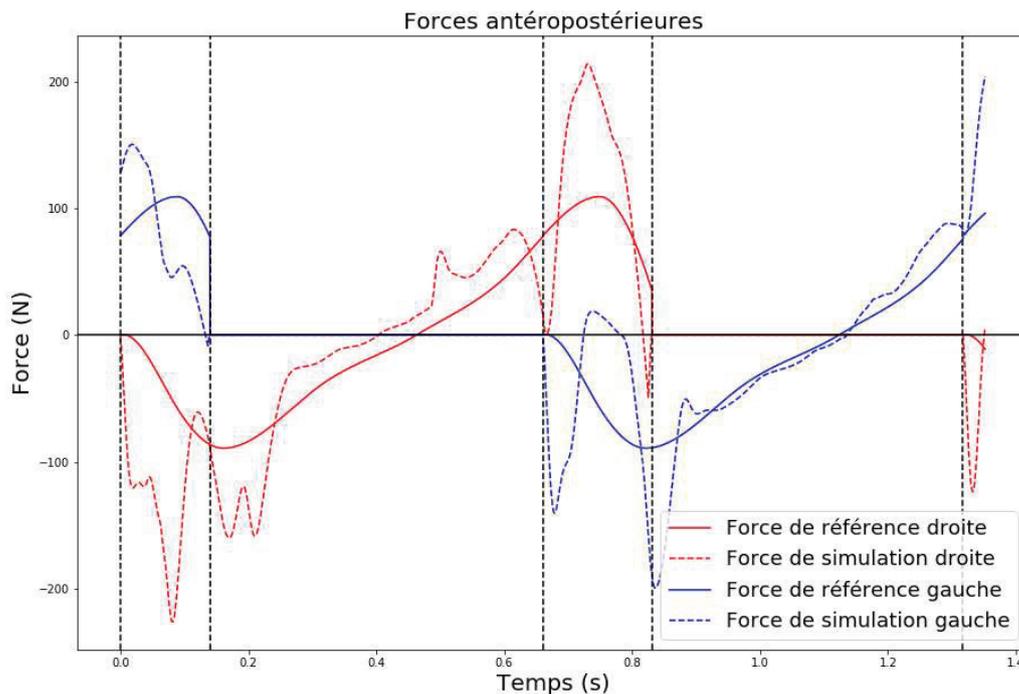


Figure 98 : Forces de réaction antéropostérieure lors d'une foulée après optimisation globale

La figure 98 montre les forces de réaction du sol antéropostérieures calculées par le modèle de contact. Le modèle semble adapté pour les phases de simple appui. Les phases les plus critiques semblent être les phases de premier et double appui. Au contact, la force de réaction augmente trop rapidement, ce qui a pour résultat une oscillation du pied autour de sa position de référence comme le montre les oscillations des forces. On remarque également que la force droite chute fortement en fin de deuxième phase (simple appui). Cela semble correspondre au pic de flexion de la hanche droite. Il est probable que la flexion de la hanche soit suffisamment importante pour modifier la position du pied au sol au point que les éléments viscoélastiques ne calculent plus de force de contact.

## 2.9. Forces de réaction verticales

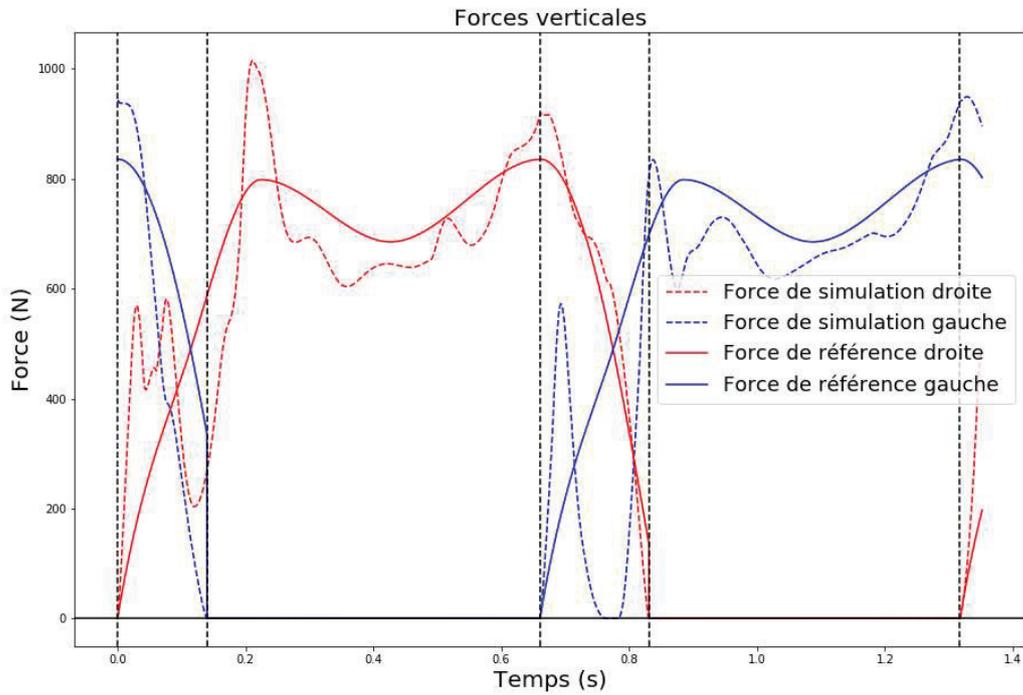


Figure 99 : Forces de réaction verticales lors d'une foulée après optimisation globale

La figure 99 montre que, de manière similaire aux forces de réaction antéropostérieures, la première phase de simple appui est une partie critique de la détermination des forces de contact. Comme précédemment, la force calculée augmente fortement dès le contact du pied avec le sol puis diminue fortement au point de perdre le contact lors du contact du pied gauche. Initialement, ce phénomène était pris pour un rebond du pied dû à la forte réaction du sol et au fait que le talon se relevait après le contact initial. Cependant, le talon gauche ne semble pas rebondir après son contact initial mais perd tout de même le contact, signifiant que le problème est de nature différente. Cela pourrait être attribué à une plantiflexion de la cheville trop précoce. En revanche pour le reste de la phase d'appui, les forces calculées semblent suivre une trajectoire attendue.

## 2.10. Forces de réaction médio-latérales

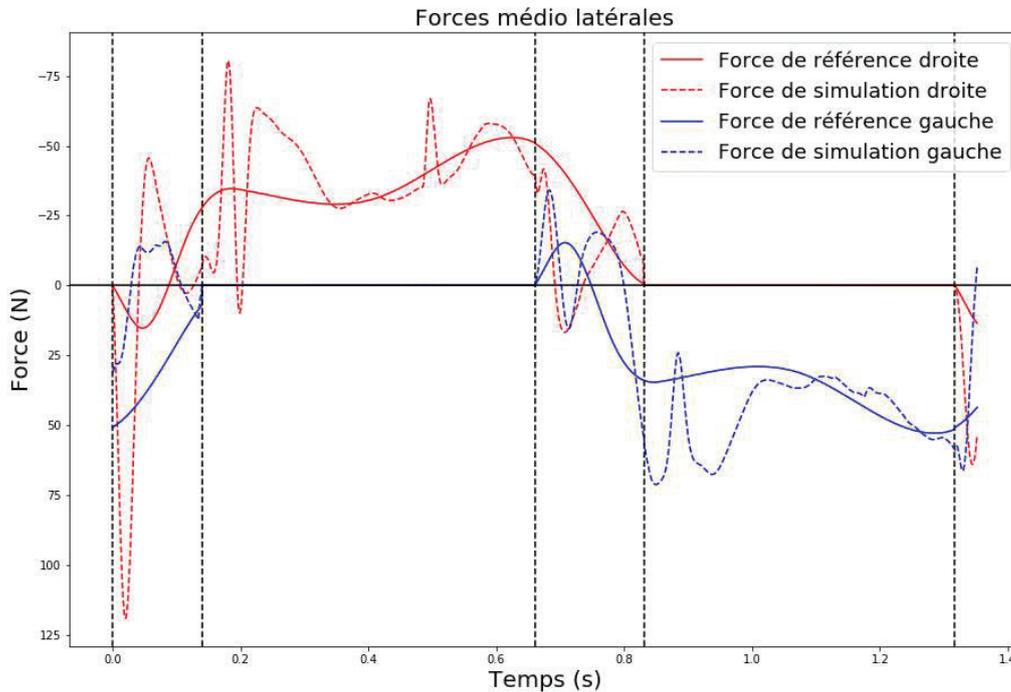


Figure 100 : Forces de réaction médio-latérales lors d'une foulée après optimisation globale

La figure 100 montre que les forces de réaction médio-latérales sont les moins bien calculées, mais cela vient aussi du fait qu'elles sont plus faibles que dans les autres directions. Les phases de double appui restent des phases critiques mais les oscillations continuent sur une partie des phases de simple appui de chaque jambe (pendant le simple appui pour la jambe droite et pendant l'envol pour la jambe gauche). De plus, les forces calculées varient également fortement lors des secondes phases de double appui.

## 2.11. Pics de couples communs à différentes articulations

Certains pics de couples se retrouvent de manière commune sur plusieurs DoF comme les oscillations en début de phase oscillante et le pic à environ 0.5 seconde (pendant le simple appui).

Les oscillations en début de phase oscillante sont dues à l'imprécision du suivi des trajectoires des talons. En effet, avec l'adaptation des couples aux changements de phases, les couples calculés pour permettre de contrôler le talon ne sont pas appliqués tels quels. Cela a pour conséquence que le suivi des trajectoires du talon est moins précis en début de phase. Cette imprécision se traduit par une augmentation de la différence entre la position du talon et sa position de référence, qui à son tour

entraîne le calcul d'un couple plus élevé. Ce phénomène provoque le pic de flexion-extension de hanche d'appui avant l'optimisation globale de tous les paramètres du contrôleur, et a été diminué par la minimisation du gain responsable du calcul du couple en début de phase. Ce phénomène reste particulièrement gênant pour le contrôle de la trajectoire médio-latérale du talon. En effet, ce contrôle est extrêmement sensible au paramétrage. La fenêtre entre des gains trop faibles pour correctement traquer la trajectoire et trop élevés pour éviter les instabilités numériques est étroite. Les DoF les plus utilisés pour contrôler cette trajectoire sont l'adduction-abduction de hanche ipsilatérale et la rotation interne-externe de hanche controlatérale. Cela explique donc également l'amplitude du pic de couple de rotation interne-externe de hanche.

Les pics de couple à environ 0.5 seconde, pendant le simple appui sont, quant à eux, provoqués par la vitesse verticale de référence du talon. Comme évoqué précédemment, la trajectoire verticale du talon est décrite par deux splines différentes afin d'imposer une dérivée nulle à un endroit précis de la trajectoire. Cependant, cela provoque une discontinuité de la seconde dérivée de la trajectoire qui se traduit par une variation brutale de la vitesse de référence (Figure 101). Les différents DoF ayant une influence sur la trajectoire verticale du talon, à savoir les flexions-extensions de hanche et de genou, sont alors impactés, résultant en ce pic de couple. La figure 101 montre que lors du deuxième pas, la vitesse étant moins bien traquée avant le changement de spline, le changement de contrôle est moindre, ce qui explique que le pic n'est pas ou peu présent à ce moment-là.

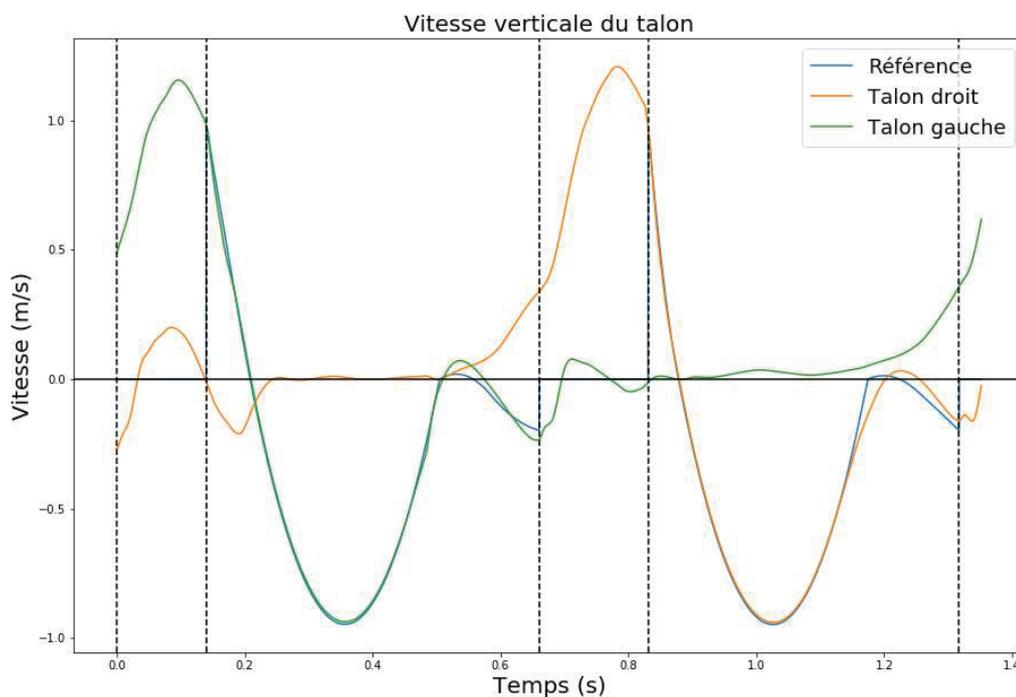


Figure 101 : Vitesse verticale des talons lors d'une foulée après optimisation globale

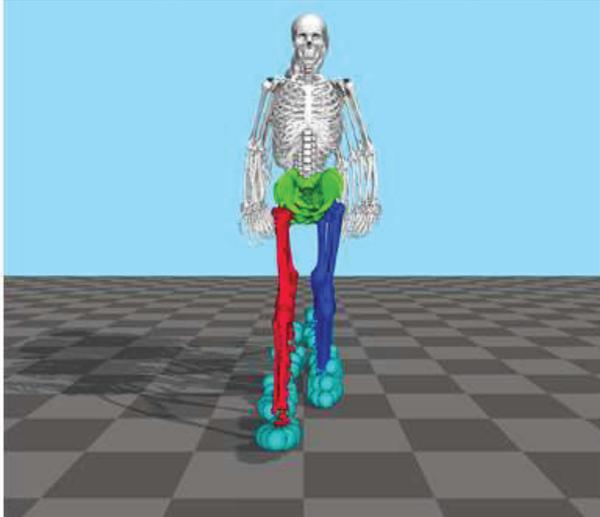
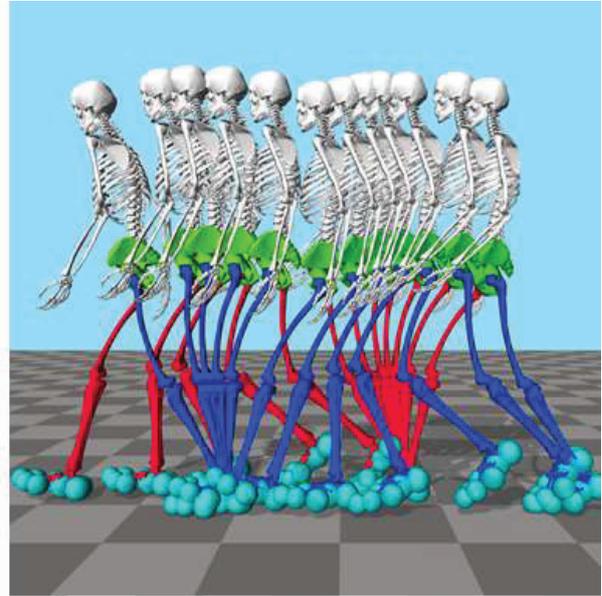
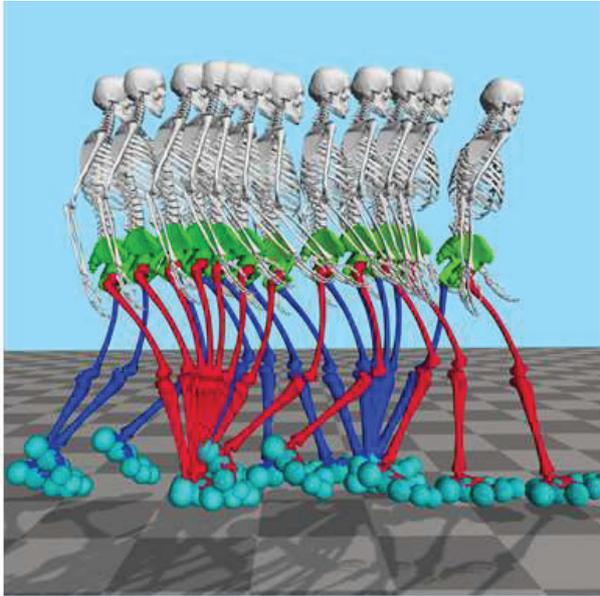


Figure 102 : Snapshot du résultat final. Le membre inférieur droit est coloré en rouge et le membre inférieur gauche est coloré en bleu



# V. Acquisition et exploitation de nouvelles données

## A. Motif d'expérimentation

L'un des objectifs du simulateur est de pouvoir produire un mouvement de marche lors du passage d'obstacle. Pour ce faire, de nouvelles données expérimentales sont nécessaires pour deux raisons : modifier ou compléter le modèle statistique et valider le mouvement simulé dans des tâches plus complexes que la marche.

Le premier motif d'expérimentation est le besoin de compléter le modèle statistique. Le passage d'un obstacle de la vie courante, qu'il s'agisse d'une marche, d'une lacune, ou du passage sur un tapis roulant, requiert une adaptation du mouvement. Deux cas de figures sont alors envisageables.

Dans un cas idéal, le modèle statistique provenant uniquement de données concernant la marche serait suffisant. Il serait possible de garder le modèle statistique tel quel et de n'adapter que ce qui est à l'origine adapté lors de la simulation. Il serait par exemple envisageable de ne modifier que la position antéropostérieure de la pose du pied pour simuler le passage d'une lacune, modifier la hauteur de la pose du pied pour simuler le passage d'une marche, ou adapter les forces virtuelles s'appliquant au CoM pour s'adapter au changement de vitesse lors du passage sur un tapis roulant et si besoin au passage d'une marche ou d'une lacune.

Cependant, il est probable que le modèle statistique issu de la marche normale soit insuffisant pour modéliser un passage d'obstacle de manière biofidèle, voire de pouvoir le modéliser de manière stable. Dans ce cas, il sera nécessaire de recourir à des modèles statistiques complémentaires se basant chacun sur des mouvements de passage d'obstacle différents. Les différentes trajectoires seront alors utilisées lorsque nécessaire. La transition s'effectuerait lors d'un changement de phase, qui est déjà conçu pour assurer la continuité des trajectoires.

Quoi qu'il en soit, le travail pour adapter le mouvement de la marche au mouvement de passage d'obstacle, et notamment la détermination de la nécessité ou non d'un nouveau modèle statistique pour un mouvement particulier procurerait un résultat intéressant en soit. En effet, s'il est possible de produire un mouvement biofidèle de passage d'obstacle à partir du modèle statistique de la marche, cela pourra être interprété comme indice que l'adaptation à cet obstacle ne relève que d'une légère modification de la stratégie de marche existante. Au contraire, s'il est impossible de produire un mouvement biofidèle à partir du modèle statistique de base, cela indiquera que le passage d'un obstacle nécessite une telle modification du mouvement que l'on pourra considérer la stratégie mise en œuvre

comme étant différente. De même, si le modèle statistique de base permet le passage d'obstacle mais de manière non biofidèle, cela indiquera que la stratégie de marche initiale, même si suffisamment robuste pour passer un obstacle, est jugée trop peu optimale, et qu'un changement de stratégie est alors préféré.

Le second motif du besoin de données supplémentaires concerne la validation du simulateur. Que l'adaptation du mouvement se fasse avec ou sans modèle statistique complémentaire, des données expérimentales seront nécessaires pour valider le résultat de la même manière que pour la marche simple.

## B. Expérimentation

Les obstacles auxquels le simulateur devra s'adapter sont le passage de lacune, le passage d'une marche et le passage sur un sol en mouvement. L'objectif initial de l'expérimentation était donc d'obtenir des données de mouvements sur ces différents obstacles et dans différentes conditions (longueur de lacune, hauteur de marche, vitesse du tapis). Cependant, étant donné la complexité de la maquette à réaliser pour simuler tous ces obstacles, et notamment le passage d'une marche à différentes hauteurs, il a été décidé, pour cette expérimentation, de ne simuler que le passage de lacune et le passage sur un sol en mouvement. Cette expérimentation a fait l'objet de la rédaction d'un article concernant la modification des marges de stabilité antéropostérieures lors du passage d'un sol fixe à un sol en mouvement et vice-versa. La méthodologie employée et un exemple de résultats qu'il est possible d'obtenir via la base de données obtenue sont présentés dans cet article.

## C. Article

Perturbation de la marche sur trottoir roulant – Analyse de la marge de stabilité antéropostérieure et de la longueur de pas.

Article soumis au journal *Gait & Posture*

### 1. Introduction

Le contrôle de la stabilité est un aspect fondamental de la marche humaine. La stabilité a été largement étudiée pour la marche sur sol plat et pour des tâches entraînant des perturbations telles que le passage d'obstacle (Hak et al., 2019), le glissement ou trébuchement (König et al., 2019; T.-Y. Wang et al., 2012) et des perturbations latérales (Hak et al., 2012; Roeles et al., 2018). La majorité des perturbations ont été induites par accélération ou décélération de tapis roulant pendant que le sujet

marche sur celui-ci. Cependant, un type de perturbation ayant peu été étudié dans la littérature est la montée et descente d'un trottoir roulant. Les trottoirs roulants sont des systèmes destinés à diminuer le temps de trajet des piétons ou pour leur permettre d'avancer sans avoir à marcher.

Jusqu'à présent, certains auteurs ont analysés l'ergonomie des trottoirs roulants dans les aéroports, signalant des problèmes potentiels de stabilité (léger balancement, secousses) (Hawkins & Atha, 1976) et décrivant deux stratégies principales adoptées par les passagers : s'arrêter après être monté ou continuer à marcher (Young, 1995). Les perturbations de la marche sur les tapis roulants sont un problème connu et des messages de prévention peuvent être diffusés. Cependant, à la connaissance des auteurs, la stabilité de la marche induite par un trottoir roulant n'a jamais été étudiée de manière quantitative.

Une méthode d'analyse de la perturbation de la marche largement utilisée dans la littérature est le calcul de la marge de stabilité (MoS) tel que proposé par Hof et al. (Hof et al., 2005). La MoS a été largement utilisée pour étudier la stabilité de la marche pour différentes situations et pathologies (Watson et al., 2021) et certaines études ont démontré une relation entre la MoS et les paramètres spatio-temporels de la marche tels que la fréquence, la longueur, le temps, la largeur de pas et la vitesse de marche (Alamoudi & Alamoudi, 2020; D. Anderson et al., 2007; Hak et al., 2013; Iwasaki et al., 2022).

L'objectif de cet article est d'étudier la MoS ainsi que ses relations avec les paramètres spatio-temporels dans des situations demandant des adaptations, en l'occurrence monter et descendre d'un trottoir roulant. Plus précisément, nous nous concentrerons sur la MoS antéropostérieure pendant le pas de transition au moment où les sujets ont un pied sur le tapis roulant et un pied sur le sol.

Notre hypothèse pour cette étude est que monter ou descendre d'un trottoir roulant est une perturbation qui va demander aux sujets d'adapter leur marche pour garder l'équilibre, et que ces perturbations devraient être visibles via les valeurs de la MoS.

## 2. Méthode

### 2.1. Sujets

Les sujets pour cette étude étaient 16 adultes en bonne santé, 10 hommes et 6 femmes. Leur âge (moyenne, écart type) était 27,9 (7,2) ans, leur masse était 69,5 (11,9) Kg, et leur taille était 1,73 (0,08) m. Ils n'ont déclaré aucune pathologie ou trouble de la marche. Le protocole expérimental a été approuvé par le comité d'éthique de l'Université Gustave Eiffel et les sujets ont donné leur permission écrite.

## 2.2. Acquisition des données

Il a été demandé aux sujets de réaliser sept tâches dans un couloir de marche. Chaque tâche a été répétée sept fois après une période d'adaptation pendant laquelle les sujets étaient autorisés à réaliser autant d'essais que nécessaire pour que la tâche soit confortable. Le couloir de marche était composé d'un tapis de marche double bande instrumenté (Treadmetrix©, Park City, Utah, Etats Unis), de deux estrades de deux mètres chacune qui étaient positionnées à chaque extrémité du tapis. Le long du couloir de marche, des barres de sécurité étaient installées dans le cas où les sujets perdraient l'équilibre. Ce montage simulait un vrai trottoir roulant. (Figure 103).



Figure 103: Montage expérimentale. Les points rouges représentent l'emplacement des marqueurs réfléchissants et les rectangles verts représentent l'emplacement des plateformes de force.

Les tâches étaient de la marche à vitesse confortable sur sol fixe, et différentes conditions de marche et de position statique avant et après être monté ou descendu du tapis à deux vitesses de bande : 0,5 et 1 m/s. Ces conditions étaient basées sur les stratégies adoptées par les piétons utilisant un trottoir roulant (Young, 1995) et sont résumées dans la table 8.

Nom de la tâche	Montée sur le trottoir roulant		Descente du trottoir roulant		Vitesse de bande (m/s)
	Situation sur le sol	Situation sur le trottoir roulant	Situation sur le trottoir roulant	Situation sur le sol	
Walk	Marche	Marche			0
In-Walk-Walk* (IWW)	Marche	Marche			0.5
In-Walk-Stand (IWS)	Marche	Statique			0.5 et 1
Out-Walk-Walk* (OWW)			Marche	Marche	0.5
Out-Stand-Walk (OSW)			Statique	Marche	0.5 et 1

Table 8: Résumé des différentes stratégies.

\* Ces conditions ne pouvaient pas être réalisées à 1 m/s à cause de la longueur du tapis qui ne permet pas de reproduire une marche réaliste sur le trottoir roulant

Les données cinématiques ont été acquises via un système de caméra optoélectronique (Qualisys, Göteborg Suède) composé de 10 caméras. Les sujets étaient équipés d'un ensemble de marqueurs réfléchissants composé de 42 marqueurs, mais seulement une partie a été utilisée dans cette étude : les 13 marqueurs du modèle simplifié de Tisserand et al. (Tisserand et al., 2016) pour le calcul du CoM, plus les marqueurs du talon et de la première tête des métatarses du modèle de Leardini et al. (Leardini et al., 2007) pour l'estimation des paramètres spatiotemporels. Les données de force ont été acquises via trois plateformes de forces. Deux d'entre elles faisaient partie du tapis de marche instrumenté. La troisième était une plateforme (Bertec®, Columbus, Ohio Etats Unis) placée dans l'une des estrades du couloir de marche. (Figure 103). Les données cinématiques et de force ont été acquises à une fréquence de 300 Hz.

### 2.3. Traitement des données

Le CoM a été calculé en utilisant un modèle segmentaire et les équations de régressions de Tisserand et al. (Tisserand et al., 2016), et le XCoM correspondant a été défini comme :

$$\overrightarrow{XCoM} = \overrightarrow{CoM} + \frac{\overrightarrow{v_{i,appui}}}{\sqrt{\frac{g}{l}}}$$

Avec  $l$  la longueur des jambes,  $g$  l'accélération de la gravité et  $\overrightarrow{v_{i,appui}}$  la vitesse du CoM exprimée dans le repère du pied d'appui : si le pied d'appui est sur le sol (i.e. fixe dans le repère du laboratoire),  $\overrightarrow{v_{i,appui}}$  est

égal à  $\vec{v}_{i_{lab}}$ , la vitesse du CoM calculée par les données cinématiques dans le repère du laboratoire ; et si le pied est sur le tapis roulant  $\vec{v}_{i_{stance}}$  était calculée en soustrayant la vitesse des bandes à  $\vec{v}_{i_{lab}}$ . Ainsi, le XCoM est toujours calculé en fonction de l'environnement sur lequel les sujets marchent. A partir de là, la MoS peut être définie par rapport au centre de pression (COP) tel que (Hof et al., 2007) :

$$\vec{MoS} = \vec{CoP} - \vec{XCoM}$$

Pour le calcul du XCoM,  $l$  a été définie comme la hauteur moyenne du CoM pendant la tâche de marche et la vitesse a été filtrée via un filtre Butterworth avec une fréquence de coupure à 12 Hz. La position du CoP a été directement extraite des données des plateformes de force. Dans la littérature, la MoS a été à la fois définie comme la différence entre le XCoM et la base de support ou le CoP, portant une signification différente selon la définition. Dans cette étude, la MoS est calculée comme un outil pour caractériser la dynamique de la marche, notamment le positionnement du CoP pour contrôler le CoM.

La longueur de pas a été définie comme la distance antéropostérieure entre le marqueur du talon de la jambe de réception et le marqueur de la première tête des métatarses de la jambe de propulsion lors de la pose du pied (Figure 104).

Pour chaque tâche, deux MoS antéropostérieures ont été calculées à différents instants (Figure 104). La première MoS, correspondant à la jambe de propulsion, est calculée à la pose du pied controlatéral et est appelée MoS de propulsion. La MoS de propulsion sera utilisée pour caractériser les potentielles adaptations de la marche. La seconde MoS, correspondant à la jambe de réception, est calculée au lever du pied controlatéral et est appelée MoS de réception. La MoS de réception sera utilisée pour caractériser les éventuelles perturbations.

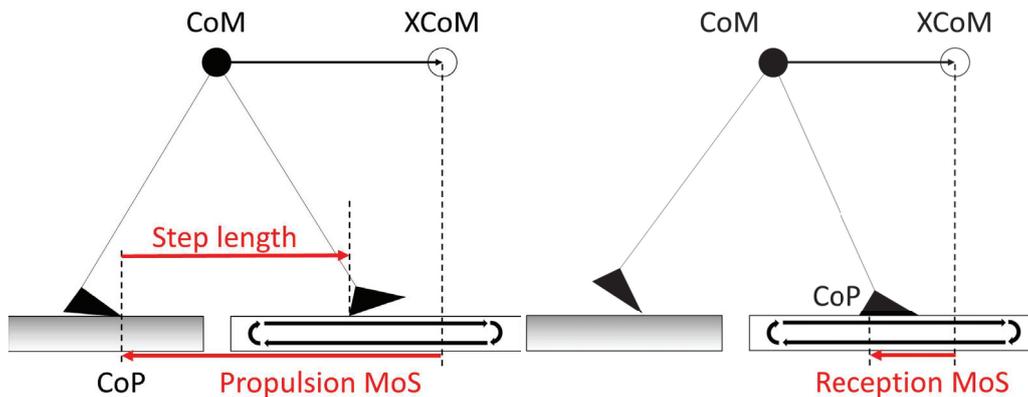


Figure 104: Illustration de la longueur de pas et de la MoS de propulsion à la pose du pied controlatéral, et de la MoS de réception au lever du pied controlatéral.

Avec cette formulation, une MoS positive indique que le CoP est en avant du XCoM, i.e. si le COP n'est pas déplacé, le CoM ne le dépassera pas et inversement. Pour que la marche soit stable, la MoS antéropostérieures doit être négative (le XCoM doit rester devant le CoP pour permettre au CoM de continuer à avancer). Cependant, pour des raisons de compréhension, bien que les MoS soient

présentées avec leur vraie valeur, elles seront discutées en valeurs absolues. Ainsi, pour un CoP identique, une MoS plus large signifiera que le XCoM est plus loin et inversement.

Les différences statistiques entre le MoS et la longueur de pas pour les tâches de marche et pour les tâches avec tapis roulant (excepté les OSW pour les MoS de propulsion et les IWS pour les MoS de réception) ont été réalisées via le test de Student. La tâche consistant à être statique sur le tapis puis en descendre en continuant à marcher (OSW) a été considérée comme représentative d'un début de marche lors de la pose de la jambe de réception. La tâche consistant à monter sur le tapis et s'y arrêter en position statique (IWS) a été considérée comme représentative d'une fin de marche lors du lever de la jambe de propulsion. Les corrélations entre les MoS et la longueur de pas ont été calculées pour toutes les tâches. Les différences statistiques et corrélations étaient considérées significatives si la valeur p était inférieure à 0,05.

### 3. Résultats

La vitesse de marche des sujets pour la tâche de marche était de 1,22 (0,05) m/s, calculée sur le temps de trois pas.

#### 3.1. Marge de stabilité

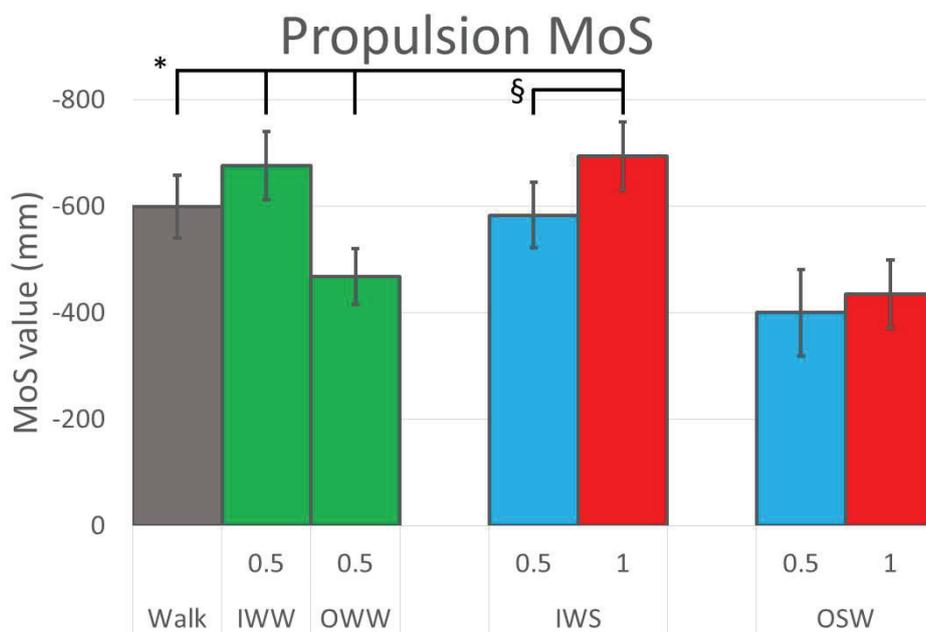


Figure 105 : MoS de propulsion, correspondant à la jambe de propulsion à la pose du pied controlatéral.

\*: Différence significative avec la tâche de marche. Puisque la MoS de propulsion pour les tâches OSW correspond à un début de marche, la différence avec la marche n'est pas indiquée pour ces situations.  
 §: Différence significative entre les vitesses des bandes à 0,5 et 1 m/s

La figure 105 montre que les MoS de propulsion des tâches IWW et OWW sont différentes de la marche. En l'occurrence, la MoS du IWW est plus grande et celle du OWW est plus petite que celle de la marche. Il y a également une différence entre les tâches de marche et de IWS quand la vitesse des bandes est à 1 m/s. La MoS du IWS est plus large que celle de la marche mais pas quand le tapis est à 0,5 m/s. De plus, une différence entre les deux MoS de propulsion des tâches de IWS est montrée avec une MoS plus grande quand la vitesse des bandes est à 1 m/s. Une telle différence n'est pas présente pour les tâches OSW.

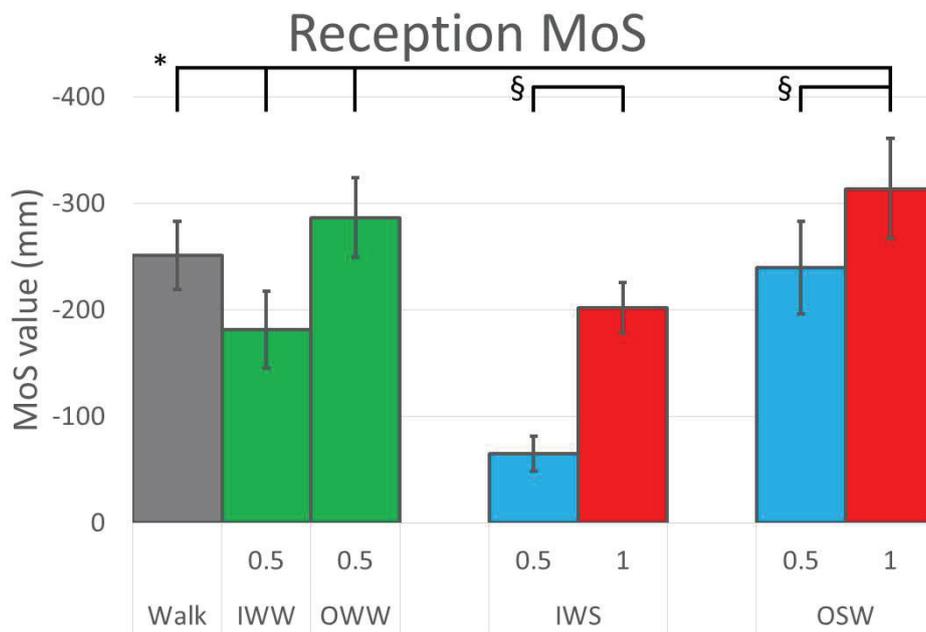


Figure 106 : MoS de réception correspondant à la jambe de réception au levé de pied controlatéral.  
 \*: Différence significative avec la tâche de marche. Puisque la MoS de propulsion pour les tâches IWS correspond à une fin de marche, la différence avec la marche n'est pas indiquée pour ces situations.  
 §: Différence significative entre les vitesses de bandes à 0,5 et 1 m/s

La figure 106 montre que les MoS de réception des tâches IWW et OWW sont différentes de la tâche de marche avec une MoS du IWW plus petite et une MoS du OWW plus grande. Il y a également une différence entre les MoS de marche et de OSW lorsque la vitesse des bandes est à 1 m/s avec une MoS du OSW plus large, mais pas lorsque le tapis est à 0,5 m/s. De plus, les tâches IWS et OSW montrent une différence de valeur de MoS de réception en fonction de la vitesse des bandes, les MoS étant plus grandes lorsque le tapis est à 1 m/s.

### 3.2. Relation entre marge de stabilité et longueur de pas

R <sup>2</sup>	Propulsion MoS	Reception MoS
<b>Walk</b>	<b>0.84</b>	<b>0.36</b>
<b>In-Walk-Walk (0.5 m/s)</b>	<b>0.69</b>	0.15
<b>In-Walk-Stand (0.5 m/s)</b>	<b>0.89</b>	0.17
<b>In-Walk-Stand (1 m/s)</b>	<b>0.74</b>	0.01
<b>Out-Walk-Walk (0.5 m/s)</b>	<b>0.27</b>	0.17
<b>Out-Stand-Walk (0.5 m/s)</b>	<b>0.77</b>	<b>0.56</b>
<b>Out-Stand-Walk (1 m/s)</b>	<b>0.55</b>	0.01

Table 9 : Coefficient de détermination entre les MoS et la longueur de pas  
Les valeurs significatives ( $p < 0,05$ ) sont écrites en gras.

La table 9 montre que pour toutes les tâches, la MoS de propulsion est corrélée à la longueur de pas. Cependant, la valeur des coefficients de détermination varie beaucoup en fonction de la tâche, allant de 0,27 pour le OWW, à 0,89 pour le IWS à 0,5 m/s. Seul le coefficient du IWS à 0,5 m/s a une valeur plus élevée que pour la marche. Toutes les autres tâches résultent en un coefficient plus faible, en particulier le OWW. Au contraire, seules deux tâches montrent une corrélation significative entre la MoS de réception et la longueur de pas, ces tâches étant la marche et le OSW à 0,5 m/s.

## 4. Discussion

### 4.1. Marge de stabilité

Une plus grande MoS génère une plus grande accélération antérieure du CoM alors qu'une plus faible MoS génère une moins grande accélération antérieure. Pendant la tâche IWW, la vitesse globale des sujets augmente de leur vitesse de marche jusqu'à atteindre leur vitesse de marche plus la vitesse des bandes. La plus large MoS de propulsion observée pendant cette tâche indique que les sujets semblent se préparer à la transition en augmentant leur vitesse de CoM avant de marcher sur le tapis. Au contraire, pendant la tâche OWW, la vitesse globale des sujets diminue de leur vitesse de marche plus la vitesse des bandes pour revenir à leur vitesse de marche. La plus faible MoS de propulsion indique

alors que les sujets semblent se préparer à la transition en diminuant leur vitesse de CoM avant de marcher hors du tapis. Cependant, la MoS de réception pour ces deux tâches suggère que ces adaptations ne sont pas suffisantes pour éviter les perturbations. En effet, la plus faible MoS de réception pour la tâche IWW indique que la vitesse du CoM dans le repère des bandes est plus faible que la vitesse du CoM pendant la marche dans le repère du sol, signifiant que les sujets ont encore besoin d'accélérer vers l'avant. Au contraire, la plus grande MoS de réception pour la tâche OWW indique que la vitesse du CoM est encore plus grande que pendant la marche classique, signifiant que les sujets ont encore besoin de décélérer. Dans les deux cas, la MoS de réception montre que la marche n'est pas encore complètement adaptée juste après la transition.

La MoS de propulsion pour le IWS montre une différence de préparation selon la vitesse des bandes puisque les sujets semblent davantage accélérer vers l'avant quand les bandes ont une vitesse de 1 m/s. De plus, il est intéressant de noter que lorsque les bandes sont à 0,5 m/s, la MoS de propulsion n'est pas significativement différente de la MoS de propulsion de la tâche de marche. Cela peut indiquer que la stratégie pour se préparer à la fin de marche sur les bandes à 0,5 m/s n'est pas différente de la stratégie pour continuer de marcher normalement, ou au moins que la différence de stratégie n'impacte pas la MoS.

Les MoS de réception montrent également une différence de perturbation en fonction de la vitesse des bandes. Les deux tâches IWS correspondent à une fin de marche. Il est attendu que les MoS de réception pour de telles tâches soient les plus petites possibles pour permettre au CoM d'atteindre la position du CoP avec une vitesse faible. En s'arrêtant sur le tapis, cela semble être effectivement le cas pour une vitesse de bande de 0,5 m/s. Cependant, on observe que la MoS de réception lorsque les bandes sont à 1 m/s est significativement plus grande. Similairement à la tâche du IWW, cela peut indiquer que la préparation à la perturbation n'a pas été suffisante et que la vitesse du CoM dans le repère des bandes n'est pas adaptée à une fin de marche.

La MoS de propulsion pour les tâches de OSW ne sont pas différentes en fonction de la vitesse des bandes, indiquant que dans ces situations, les sujets semblent se préparer à la perturbation indépendamment de la vitesse des bandes. Les conséquences d'une adaptation indépendante à l'intensité des perturbations semblent être une différence significative de la MoS de réception puisqu'elle est plus grande lorsque les bandes sont à 1 m/s. Similairement à la tâche de OWW, cela peut indiquer que la vitesse du CoM est exagérée dans le repère du sol comparée à la marche classique. De plus, il est intéressant de noter que lorsque les bandes sont à 0,5 m/s, la MoS de réception n'est pas significativement différente de celle de la marche classique. Cela peut indiquer que la stratégie de préparation pour cette tâche permet à la marche d'être directement initiée sans besoin d'adaptation majeure ou au moins sans que l'adaptation n'affecte la MoS de réception.

De manière générale, les résultats montrent que les sujets tendent à anticiper le changement de vitesse mais que les adaptations ne sont pas suffisantes et que des adaptations supplémentaires sont nécessaires à la réception. Un indice supplémentaire de ceci est le fait que la corrélation entre les MoS de propulsion et la longueur de pas est conservée pour toutes les tâches mais que celles avec les MoS de réception sont perdues pour la majorité des tâches. En effet, en considérant que le système se comporte comme un pendule inversé linéaire seulement contrôlé par le placement du pied, il y a un lien entre la

longueur de pas et la MoS. L'utilisation de mécanismes de régulations (modification de la cadence, évolution du CoP ou modulation du moment angulaire de l'ensemble du corps) va altérer cette relation. Ainsi, le fait que cette corrélation soit conservée pendant la propulsion indique que le placement du pied est le système de régulation principal pendant la propulsion, incluant les situations où les sujets devaient adapter leur vitesse. Nous interprétons alors le fait que cette relation disparaisse à la réception comme une indication que le placement du pied seul n'était plus suffisant et que d'autres mécanismes doivent être utilisés.

Cependant, les tâches IWS et OSW quand les bandes sont à 0,5 m/s semblent être des exceptions. La MoS de réception pour le IWS et la MoS de propulsion pour le OSW étant similaires à la marche classique, et la MoS de réception pour le IWS étant ce qui est attendu s'il n'y avait pas de perturbation, cela suggère l'existence d'une vitesse de bande permettant une transition sans perturbation du MoS et potentiellement de la marche en général. Des études supplémentaires sur d'autres aspects de la marche sont cependant nécessaires pour confirmer ou infirmer cette interprétation. Un indice supplémentaire de cette supposition est tout de même que la corrélation entre la MoS de réception et la longueur de pas pour la tâche OSW à 0,5 m/s est la seule étant encore significative, signifiant que les autres aspects de la marche ne sont pas aussi altérés que dans les tâches de IWW, OWW et OSW à 1 m/s.

#### 4.2. Limitations et perspectives

Il existe plusieurs limitations à cette étude. La première concerne la longueur des bandes du tapis qui mesurent deux mètres, ce qui peut être trop faible pour que les sujets aient le temps de complètement s'adapter à la situation et ce qui peut donc limiter l'aspect reproduction d'un vrai trottoir roulant de l'expérimentation. Deuxièmement, cette étude se concentre sur les MoS AP et la longueur de pas pour quantifier les adaptations pendant les tâches. Des études supplémentaires se concentrant sur les MoS médio-latérales et les autres paramètres spatiotemporels ainsi que sur les données cinématiques et de forces pourraient compléter nos résultats.

#### 4.3. Conclusion

Notre hypothèse de départ est partiellement confirmée. Des perturbations antéropostérieures ont effectivement été détectées via la MoS, mais pas pour toutes les tâches. En particulier, les perturbations montrées par les MoS de réception indiquent que les adaptations avant de réaliser le pas sont insuffisantes et il en résulte un besoin d'adaptations supplémentaires après le pas. De plus, les résultats suggèrent l'existence d'une vitesse optimale du trottoir roulant permettant un minimum de perturbation lorsque l'on s'y arrête ou que l'on initie la marche depuis celui-ci, ce qui peut être intéressant pour un usage pratique.

## D. Résultats complémentaires à l'article

L'article présenté précédemment se concentre sur les marges de stabilités antéropostérieures. Les données concernant les marges médio-latérales ont également été traitées mais pas présentées dans l'article.

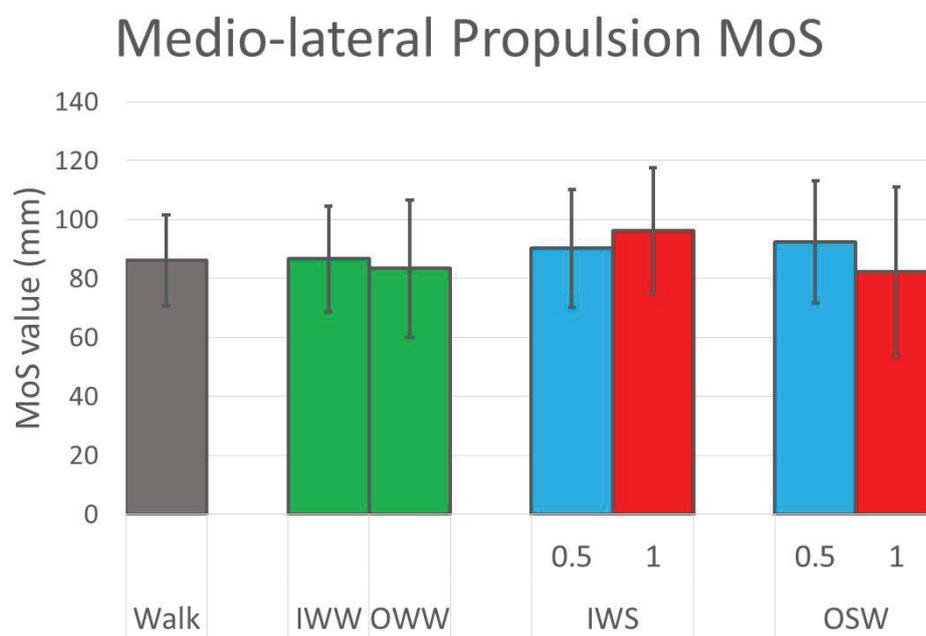


Figure 107 : MoS médio-latérales de propulsion, correspondant à la jambe de propulsion à la pose du pied controlatéral.

La figure 107 montre qu'il n'y a pas de différence significative entre les MoS de réception des différentes tâches. Cela peut indiquer que les perturbations médio-latérales sont jugées trop faibles, voire inexistantes, pour que les sujets adaptent leur marche dans cette direction.

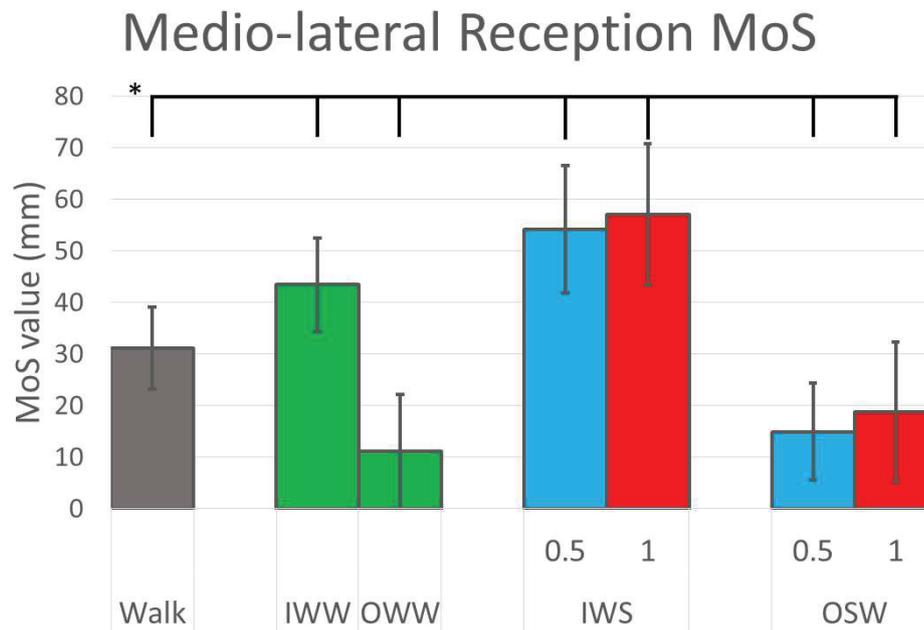


Figure 108 : MoS médio-latérales de réception, correspondant à la jambe de réception au lever du pied controlatéral.

La figure 108 montre une différence de MoS de réception selon que les tâches demandent de passer du sol au tapis et inversement. En effet, les MoS médio-latérales des tâches consistant à monter sur le tapis sont plus larges que pour la marche classique alors que les MoS des tâches consistant à descendre du tapis sont plus faibles. Plusieurs hypothèses peuvent être apportées pour expliquer ce résultat. Une première hypothèse concerne le temps de double et simple appui. Leurs durées peuvent varier lors du passage de l'obstacle, impactant les MoS. Cependant, une autre hypothèse possible est celle du ciblage de la pose des pieds par les sujets. En effet, selon le sens dans lequel les bandes avancent, un gap pouvant aller jusqu'à 5 cm est visible entre les bandes, ce qui peut pousser les sujets à augmenter leur largeur de pas lors du passage sur le tapis. Au contraire, la plateforme de force placée dans l'estrade étant centrée, il est possible que les sujets aient diminué leur largeur de pas lors du passage sur l'estrade, ce qui correspond aux résultats obtenus. Un autre indice allant dans ce sens est la corrélation entre la MoS médio-latérale et la largeur de pas (Table 10). Alors que l'on s'attendait à une corrélation au moins pour la tâche de marche classique, on observe que ce n'est pas le cas. Il est donc probable que le montage expérimental ait influencé les résultats et de ce fait, les informations que peuvent apporter ces données sont limitées dans la direction médio-latérale. On peut tout de même noter que les tâches consistant à passer du sol au tapis montrent toute la présence d'une corrélation. Cela semble indiquer que dans le cadre de cette expérimentation, une perturbation dans la direction antéropostérieure induit des adaptations dans la direction médio-latérale. Il pourrait donc être intéressant de conduire des expérimentations supplémentaires afin de mieux comprendre les phénomènes en jeu.

R <sup>2</sup>	Propulsion	Reception
Walk	0.08568	0.01621
<b>In-Walk-Walk (0.5 m/s)</b>	<b>0.3318</b>	<b>0.28908</b>
<b>In-Walk-Stand (0.5 m/s)</b>	<b>0.36365</b>	<b>0.26535</b>
<b>In-Walk-Stand (1 m/s)</b>	<b>0.52978</b>	<b>0.45036</b>
Out-Walk-Walk (0.5 m/s)	0.18532	0.15375
<b>Out-Stand-Walk (0.5 m/s)</b>	<b>0.3128</b>	<b>0.21232</b>
<b>Out-Stand-Walk (1 m/s)</b>	0.19671	0.1657

Table 10 : Coefficient de détermination entre les MoS médio-latérales et la largeur de pas  
Les valeurs significatives ( $p < 0,05$ ) sont écrites en gras.

## E. Réflexions sur le contrôle lors de passage d'obstacles

Bien qu'aucune simulation concernant le passage d'obstacle n'ait été réalisée, il est possible de réfléchir à la manière dont elles peuvent l'être.

### 1. Approche

La première étape pour passer un obstacle quel qu'il soit, est l'approche. Cette étape consiste à positionner l'appui avant le passage à un endroit optimal pour garantir la stabilité pendant le passage d'obstacle, généralement proche de celui-ci. Cela est réalisé en effectuant un ou des pas plus courts ou plus longs avant d'arriver à l'obstacle. Dans le simulateur, cela est contrôlé par le SFC dans lequel il est possible de modifier la position d'arrivée du pied en phase oscillante. Il peut également être nécessaire de diminuer la vitesse de marche, notamment avant de monter ou de descendre une marche. Dans le simulateur, cela peut être réglé en adaptant la trajectoire du CoM et par conséquent sa vitesse. Les forces virtuelles appliquées au CoM permettront alors au modèle de ralentir si besoin.

### 2. Passage de l'obstacle

Comme énoncé précédemment, deux cas de figures sont envisageables pour adapter le mouvement, l'un dans lequel le modèle statistique de base est suffisant pour adapter le mouvement, et l'autre dans lequel un nouveau modèle est nécessaire.

Se réalisant sur un sol gardant la même hauteur avant et après l'obstacle, le passage de lacune et le passage sur un sol en mouvement sont les deux situations les plus à même de pouvoir être simulées à partir du modèle statistique de base. Au contraire, monter ou descendre une marche exige une action

particulière de la jambe oscillante mais aussi de la jambe d'appui, et donc risque de nécessiter un modèle complémentaire.

## 2.1. Lacune

Dans le cas d'une lacune de longueur trop importante, la stratégie adoptée pour la passer est le saut qui est donc le résultat d'une stratégie fondamentalement différente de la marche puisque le mouvement introduit une phase sans appui. En revanche, pour une lacune pouvant se passer sans sauter, le passage peut s'assimiler en la réalisation d'un pas plus grand que la normale. Dans le simulateur, la position de la pose du pied, et donc la longueur du pas, est directement contrôlée par le SFC. Réaliser un pas plus grand revient donc à modifier la position antéropostérieure de la position cible du pied en phase oscillante afin de correspondre à une longueur suffisante pour passer la lacune. Les caractéristiques de l'obstacle peuvent donc être directement utilisées pour modifier le mouvement, ce qui rendrait l'adaptation extrêmement simple. Cependant, ce contrôle seul risque d'être insuffisant. En effet, il est probable que la trajectoire du CoM lors du passage d'une lacune diffère de la trajectoire lors d'un pas normal. De même, en suivant le principe du pendule inversé, réaliser un pas plus grand, et donc placer l'appui plus loin, aura pour conséquence de freiner la progression du CoM vers l'avant, qu'il faudra donc compenser via les forces virtuelles appliquées au CoM. Ces forces virtuelles étant elles-mêmes calculées à partir de la trajectoire du CoM, il est probable qu'une modification de sa trajectoire de référence soit nécessaire. La trajectoire de référence antéropostérieure sera de toute façon modifiée puisque la distance parcourue par le CoM sera plus importante que pour un pas normal. Cependant, modifier la trajectoire du CoM en ne modifiant que la distance des poses clefs ne changera pas la forme de cette trajectoire. Si des données expérimentales montrent que la forme de cette trajectoire n'est pas modifiée lors d'un passage de lacune, alors le modèle statistique de base pourrait suffire. Dans le cas contraire, un modèle statistique complémentaire pourrait être nécessaire. Enfin, la relation entre la vitesse, la distance parcourue et le temps du pas devra également être adaptée. Puisque le CoM parcourra une distance plus grande, il faudra décider si la meilleure stratégie consiste à augmenter la vitesse du COM ou à augmenter le temps du pas.

## 2.2. Sol en mouvement

Lors du passage sur un sol en mouvement, il n'est pas nécessaire de modifier fondamentalement la cinématique de la marche. En suivant une cinématique similaire à celle de la marche malgré l'entraînement du modèle par le tapis, l'adaptation du contrôle se fera alors plutôt du côté de la dynamique. Le modèle statistique étant justement utilisé pour prescrire une cinématique de référence, il est là aussi possible que le modèle de base soit suffisant, à condition d'adapter certaines trajectoires. La première trajectoire qui se verra modifiée est la trajectoire du CoM et notamment sa vitesse qui passera de la vitesse de marche à la vitesse de marche plus la vitesse du tapis quand on monte dessus ou inversement quand on en descend. Une première solution serait de considérer la vitesse du CoM par

rapport au pied d'appui. La seule période critique serait alors celle du double appui, lorsqu'un pied est sur le sol fixe et l'autre sur le sol en mouvement. Si on effectue un changement de référentiel au moment de la pose du pied sur le nouvel environnement de sorte que le nouveau référentiel soit celui du nouvel environnement, alors il faudrait adapter la trajectoire du CoM de sorte qu'il accélère de la vitesse cible moins la vitesse du tapis jusqu'à sa vitesse cible dans le cas d'une montée sur le tapis, et freiner de la vitesse cible plus la vitesse du tapis jusqu'à sa vitesse cible dans le cas d'une descente du tapis. Une fois sur le tapis, le point de départ et d'arrivée du CoM dans le repère global seraient mis à jour à chaque pas de temps pour prendre en compte l'avancée due au tapis, ce qui, du point de vue du contrôleur, aurait pour conséquence de suivre exactement la même trajectoire et vitesse de CoM que sur sol fixe. Une seconde solution serait de toujours considérer la position et la vitesse du CoM dans le repère global. Là aussi, la seule période critique serait celle du double appui entre les différents environnements. L'adaptation de la vitesse du CoM serait alors de la même amplitude qu'en changeant de référentiel mais en accélérant de la vitesse cible à la vitesse cible plus la vitesse du tapis lors de la montée, et freiner de la vitesse cible plus la vitesse du tapis à la vitesse cible lors de la descente. Une fois sur le tapis, la trajectoire du CoM devrait couvrir une plus grande distance et donc suivre une vitesse plus élevée que la marche normale. Cependant, cette différence serait compensée par l'entraînement du tapis de sorte que les deltas de position et de vitesse du CoM à chaque pas de temps soient similaires à ceux de la marche classique. La seconde trajectoire à adapter est celle du talon en phase oscillante. Etant donné que cette trajectoire n'est utilisée que lors de la phase oscillante, aucune adaptation n'est requise pour la phase critique de double appui. Concernant, son adaptation une fois sur le tapis, le même raisonnement peut être appliqué que pour le CoM. Dans le cas d'un changement de référentiel, les points de départ et d'arrivée de la trajectoire peuvent évoluer pour prendre en compte l'avancée du tapis et donc fournir la même trajectoire que lors de la marche classique ou être exprimés dans le repère global, prescrivant une trajectoire plus longue et donc une vitesse plus élevée, mais compensée par l'entraînement du modèle par le tapis. Une adaptation des trajectoires du CoM et du talon en phase oscillante est toutefois potentiellement nécessaire lors de la phase qui amène le pied à se poser sur le nouvel environnement. En effet, les résultats de l'article présenté ci-dessus montrent que les sujets commencent à préparer la transition d'environnement avant d'arriver sur celui-ci et continuent de s'adapter après le lever du pied controlatéral. L'article portant sur les marges de stabilité, cette adaptation signifie donc une influence sur la trajectoire et la vitesse du CoM. L'adaptation de la trajectoire du CoM dans la simulation ne doit donc pas se faire uniquement sur la phase de double appui mais aussi avant et après. On peut aussi s'interroger sur la trajectoire du talon arrivant sur le nouvel environnement. Il est possible que la vitesse du talon s'adapte au moins en partie à la vitesse du sol du nouvel environnement, ce qui aurait pour conséquence de modifier sa trajectoire globale. Des analyses supplémentaires seront nécessaires pour déterminer la nature et l'amplitude de ces adaptations, et si celles-ci peuvent être simulées sans nouveau modèle statistique.

### 2.3. Montée ou descente de marches

Dans le cas d'une montée ou d'une descente de marches, la différence de cinématique avec la marche classique nécessitera probablement de faire appel à un modèle statistique complémentaire. La cinématique de la jambe en phase oscillante est déterminée à partir de la trajectoire du talon. Bien qu'il soit possible de modifier la hauteur du point d'arrivée de la trajectoire pour l'adapter, la forme de la trajectoire du talon est différente entre la marche et la montée ou descente d'une marche. De même, la jambe en phase d'appui doit permettre d'élever ou d'abaisser la hauteur du modèle lors du franchissement de l'obstacle, ce qui fait appel à des trajectoires articulaires différentes de la marche sur sol plat. Une solution pour se passer d'un modèle statistique prescrivant des angles articulaires serait de donner plus de poids aux forces virtuelles lors du passage de l'obstacle pour permettre au CoM de monter ou de baisser selon une trajectoire de référence. Là encore, l'analyse de données expérimentales pourrait permettre de déterminer si la création de nouvelles trajectoires de référence pour le CoM et les talons peut se faire en ne connaissant que les caractéristiques de l'obstacle ou si les trajectoires sont trop complexes et qu'un modèle statistique complet est nécessaire.

### 2.4. Transition entre marche et passage d'obstacle

Une adaptation des trajectoires de référence à chaque nouvelle foulée étant déjà implémentée, le contrôle du passage d'obstacle devrait idéalement pouvoir s'implémenter naturellement dans le contrôle en remplaçant simplement les trajectoires de référence de la nouvelle foulée par les trajectoires permettant de passer l'obstacle. Si besoin, l'adaptation peut commencer un ou plusieurs pas en amont en ajustant la longueur des pas et la vitesse du CoM pour mieux correspondre à la pose optimale de début de passage d'obstacle. De plus, les trajectoires de référence peuvent être construites de sorte que la pose de fin de passage d'obstacle corresponde à une pose de début de foulée de marche classique. Là encore, les différences entre le passage d'obstacle et la reprise de la marche subséquente seraient les trajectoires de référence qui seraient adaptées comme elles le seraient si elles suivaient une foulée de marche classique.

Une attention particulière devra tout de même être portée aux éventuelles stratégies supplémentaires. Il est par exemple possible que le torse soit utilisé de manière particulière pour aider au passage d'obstacle. Idéalement, si le contrôle est suffisamment robuste, ces stratégies supplémentaires, si elles existent, seraient présentes dans la simulation. Dans le cas contraire, c'est que ces stratégies ne sont pas des extensions des stratégies déjà implémentées et qu'il faut donc compléter le contrôleur.

# VI. Discussion

## A. Evolution du simulateur

### 1. Choix d'un contrôle de base

Les contrôleurs se basant sur la cinématique articulaire se composent généralement d'un PDC permettant de suivre un mouvement initial, plus d'autres éléments pour adapter ce mouvement ou corriger les instabilités durant la simulation. Le choix du contrôleur sur lequel baser le simulateur s'est donc fait en fonction de ces éléments supplémentaires permettant d'adapter le mouvement.

Les contrôles développés pour les modèles 2D ne sont pas envisagés comme base de travail puisqu'ils ne sont pas pensés pour gérer l'équilibre médio-latéral du modèle. Les contrôles issus du domaine de la robotique sont également pensés pour des modèles ayant des caractéristiques et des contraintes différentes d'un modèle humain. Par exemple, beaucoup font appel au principe du ZMP qui suppose que le pied ne roule pas sur le sol. Si cette particularité cinématique est acceptable pour un robot, elle ne l'est pas dans le cadre d'une simulation dont l'objectif est la biofidélité par rapport à un mouvement humain. Les méthodes de contrôle les plus à même de correspondre aux besoins de ce projet sont donc issues du domaine de l'infographie.

C'est finalement le travail de Coros (Coros et al., 2010) qui a été choisi comme base pour construire le contrôleur. En effet, les briques de contrôles présentées permettent de contrôler l'équilibre du modèle lors du prochain pas via la pose du pied comme beaucoup d'autres contrôleurs, mais aussi de contrôler l'équilibre pendant le pas actuel en jouant sur les articulations de la jambe en phase d'appui, comme lors de la marche humaine.

La première étape du développement du contrôleur a donc été de reproduire ce contrôle dans l'environnement d'OpenSim et en l'adaptant au modèle utilisé. Avec ce contrôle, le simulateur a été capable d'enchaîner cinq pas avant que le modèle ne tombe. L'instabilité était due au fait que le modèle réalisait des pas de plus en plus larges. A ce moment du développement, le réglage des paramètres se faisait exclusivement manuellement, et il est probable qu'une optimisation aurait permis de produire une marche plus stable permettant d'enchaîner plus de pas. Le choix d'utiliser ce contrôleur comme base s'est donc avéré concluant, mais des modifications étaient nécessaires pour que le simulateur réponde aux objectifs de biofidélité.

En effet, plusieurs trajectoires de référence et valeurs nécessaires au contrôle du modèle ne suivaient pas de principes biomécaniques, mais étaient plutôt le résultat de fonctions arbitraires. De ce fait, la cinématique, bien que potentiellement assurant un mouvement stable, n'était pas complètement biofidèle. De plus, le contrôle se basant uniquement sur la cinématique, la biofidélité des couples articulaires n'était pas un élément décisif dans sa conception et n'était pas semblable à ceux de la littérature. En conséquence, bien que le modèle fût capable de marcher, le fait que les couples articulaires soient différents de ceux de la littérature était la preuve que le contrôle des DoF ne se faisait pas de la même manière que chez un humain. Enfin, les briques de contrôle étant présentées dans les grandes lignes, leur implémentation dans le simulateur était simple et leur fonctionnement interne et au sein du contrôle global devait être affiné pour répondre aux objectifs.

## 2. Tentatives d'implémentations infructueuses

### 2.1. Contrôle de l'orientation du pelvis

Le pelvis étant le segment de référence du modèle, c'est lui qui détermine l'orientation globale du bipède. Afin de contrôler cette orientation, une brique de contrôle dédiée a été implémentée. Cette brique de contrôle utilisait les DoF des hanches des jambes d'appui pour appliquer un couple issu d'un PDC suivant les trajectoires de référence du pelvis. Cependant, pour que cette brique ait un effet notable sur l'orientation du pelvis, les couples produits devaient être trop importants et venaient à gêner les autres briques de contrôle. Ainsi, au vu des avantages et des inconvénients, l'implémentation de cette brique a été abandonnée. De plus, le fait que ce segment ne soit pas directement contrôlé, son mouvement au cours de la simulation pourra alors servir d'indicateur sur la qualité du contrôle et donc être un terme de la fonction de coût pour l'optimisation des paramètres du contrôleur.

### 2.2. Contrôle proposé par Hof (2008)

L'une des valeurs à déterminer via un principe de la biomécanique est la position de pose du pied en phase oscillante. Dans le contrôle de base, cette position est déterminée via le principe du pendule inversé en calculant la position nécessaire pour que le CoM se place à la verticale du CoP Eq. (4). La position est alors décalée vers l'arrière pour permettre au CoM de dépasser le CoP dans la direction antéropostérieure tandis qu'elle est décalée latéralement pour empêcher le CoM de dépasser le CoP dans la direction médio latérale Eq. (5). Cette méthode permet d'obtenir des résultats stables, mais le décalage permettant de déterminer la pose finale du pied n'est pas issu d'un principe biomécanique.

Une solution à ce problème est d'utiliser le concept de marge de stabilité. Hof ayant proposé ce concept en 2005 (Hof et al., 2005), il proposa plus tard un contrôle de l'équilibre basé sur la marge de stabilité (Hof, 2008). Le contrôle proposé consiste à placer le CoP du prochain pas à une distance déterminée par le XCoM plus une marge de stabilité Eq. (7). Dans le cas où une perturbation se

produirait, le XCoM n'aurait pas la même position que d'habitude par rapport au CoM, et la longueur de pas s'en retrouverait changée. Or, la vitesse de marche et la longueur et le temps de pas étant liés, si la longueur de pas venait à changer, le modèle pourrait rester stable mais avec une vitesse définitivement modifiée. Ainsi, le temps de pas est également modifié en fonction de la marge et de la longueur de pas pour assurer que les valeurs initiales de vitesse et de longueur et le temps de pas soient conservées lors des pas subséquents :

$$T_n = \frac{1}{\omega_0} \ln \left( \frac{s}{XCoM_x - CoP_x} + 1 \right)$$

Avec  $T_n$  le temps de pas et  $s$  la longueur de pas. De même, une perturbation dans la direction médio-latérale aurait pour résultat un décalage définitif de la position moyenne du XCoM et un terme supplémentaire est également ajouté en plus de la marge de stabilité pour recentrer le XCoM sur une valeur souhaitée.

Ce contrôle a été implémenté et testé sur un modèle numérique de pendule inversé parfait et a montré des résultats convaincants. Cependant, bien qu'ayant des résultats prometteurs, son implémentation dans le cadre d'une simulation utilisant un modèle de bipède biofidèle n'a pas pu produire de marche stable pour deux raisons. Premièrement, le fait d'utiliser un modèle de pendule permet de placer le CoP du prochain pas n'importe où et à n'importe quel moment sans prendre en compte l'évolution dynamique du système avant que le CoP n'ait besoin d'être placé. Dans une simulation utilisant un bipède, le pied en phase oscillante suit une trajectoire en partie définie par son point d'arrivée, et donc par équivalence, par le placement du CoP du prochain pas, ce qui implique de connaître à l'avance le point d'arrivée en question. Même en définissant un point d'arrivée en fonction de la position prévue du XCoM en fin de pas, le contrôleur n'est pas assez robuste pour pouvoir changer rapidement et précisément la trajectoire du pied pour que le point et le moment d'arrivée du pied soient aussi précis qu'en utilisant un modèle de pendule. Deuxièmement, le modèle utilisé est un modèle de pendule inversé parfait dans lequel la marche est modélisée par une succession de phases de simple appui sans phase de double appui. Le contrôle proposé n'est donc pas conçu pour un modèle dans lequel des doubles appuis sont présents, et il s'avère après tentative d'implémentation du contrôle que la différence entre les situations est trop importante pour que le contrôle puisse continuer de fournir une marche stable.

### 3. Modifications pérennes du contrôle de base

Les tentatives d'implémentation précédentes n'ayant pas donné de résultats satisfaisants, d'autres modifications ont dû être développées pour améliorer la biofidélité de la simulation.

Un premier point à améliorer concerne notamment le contrôle de l'équilibre. Comme expliqué précédemment, les deux aspects du contrôle de l'équilibre, à savoir les forces virtuelles appliquées au CoM et la position de la pose du pied, sont déterminés et contrôlés via des valeurs ou fonctions n'ayant

pas de fondement biomécanique. Une solution pour remédier à ce problème a été d'également utiliser le modèle statistique pour déterminer les trajectoires du CoM et du talon en phase oscillante.

Dans le contrôle de base, la vitesse cible du CoM est considéré constante alors qu'elle ne l'est pas lors de la marche. En ciblant une vitesse constante, les forces virtuelles appliquées au CoM vont alors tenter d'accélérer le CoM lorsqu'il est censé ralentir et inversement. Utiliser le modèle statistique pour déterminer la trajectoire du CoM permet de prendre en compte ces variations de vitesse, et donc d'éviter des contrôles qui auront pour conséquence l'opposé de ce qu'il doit normalement se passer. De plus, déterminer à l'avance la trajectoire du CoM permet également de déterminer la trajectoire du XCoM. Ce faisant, il est possible de déterminer la position de pose d'appui du pied en utilisant le concept de marge de stabilité en prenant en compte la position de référence du XCoM à la fin du pas. A ce stade de développement, et pour commencer par un contrôle le plus simple possible, la position d'arrivée du pied dans la direction antéropostérieure est considérée comme fixe et n'utilise donc pas le concept de marge de stabilité. Fixer à la fois la longueur et le temps de pas ne serait pas stable dans le cas d'un modèle de pendule inversé. Cependant, le simulateur n'étant pas prévu pour tester la réponse à des fortes perturbations imprévues, les seules perturbations présentes seront dues aux erreurs numériques. Ces perturbations étant normalement moins intenses que des perturbations physiques, le contrôle du CoM devrait pouvoir les compenser. Cependant, la longueur de pas devait tout de même être déterminée via un principe biomécanique. Cette nécessité a alors permis de déterminer une relation entre longueur de jambe, vitesse de marche et longueur de pas, décrite dans la partie contrôle EQ. (14).

La détermination des trajectoires du talon en phase oscillante est également importante, à la fois pour la biofidélité du mouvement, mais aussi pour contrôler la vitesse d'arrivée du pied sur le sol. La trajectoire du talon dans la direction antéropostérieure finit par une pente plus douce qu'avec une trajectoire linéaire et termine donc avec une vitesse plus faible. Cela permet notamment de réduire la force de réaction du sol à la pose du pied. La trajectoire médio-latérale quant à elle, tend à éloigner le pied vers l'extérieur avant de le ramener vers l'intérieur puis de le poser. Le fait que la vitesse du pied soit dirigée médialement permet pendant un court instant à la force de réaction d'être dirigée latéralement, soit dans le même sens que l'accélération du CoM à ce moment-là, avant d'appliquer une force dans le sens contraire pour renvoyer le CoM médialement. Cela permet un meilleur amortissement de la vitesse du CoM que si la force de réaction était directement orientée médialement et donc de faciliter le transfert du poids d'une jambe sur l'autre. Sans cette adaptation, le CoM était renvoyé de l'autre côté trop rapidement pour maintenir une marche stable.

Un deuxième point à améliorer est la trajectoire des couples articulaires. Ceux-ci n'étant pas directement contrôlés, il est nécessaire de déterminer comment adapter le contrôle pour que les couples soient biofidèles. Comme évoqué précédemment, même si la cinématique est biofidèle, si les couples articulaires ne le sont pas, c'est que le contrôle ne reproduit pas correctement ce qu'il se passe lors de la marche humaine. L'étude des couples articulaires est donc un bon indicateur pour savoir si la méthode de contrôle est correcte ou non. Néanmoins, cet indicateur n'a pas été choisi comme un terme de la fonction de coût pour l'optimisation des paramètres du contrôle. Il aurait fallu pour cela construire un modèle statistique des couples puisque la dynamique inverse ne peut pas être directement calculée à partir des modèles statistiques de la cinématique articulaire et des forces de réaction du sol.

Une première série de solutions pour gérer les couples consiste à gérer les transitions de phase, notamment pour éviter les discontinuités et les pics de couples. L'une de ces solutions consiste justement en l'adaptation des trajectoires cinématiques de référence initialement construites et suivies par le PDC pour qu'elles soient continues. Cette stratégie d'adaptation a par la suite également été implémentée en début de phase oscillante pour adapter les trajectoires du talon, à la différence que celles-ci sont continues avec les valeurs de position et vitesse du talon. Le fait que les positions et vitesse de référence du talon soient identiques à la position et à la vitesse de simulation en début de phase oscillante permet de commencer cette phase avec des couples nuls provenant du SFC. Si ce n'était pas le cas, même en utilisant une adaptation des couples totaux, ces derniers pourraient être suffisamment influencés par le nouveau contrôle et créer des irrégularités dans la courbe de couple total. L'adaptation des couples totaux est également une solution apportée à cette problématique. En effet, même si le PDC suit des trajectoires continues à chaque début de foulée grâce à l'adaptation des trajectoires, le choc du contact du pied sur le sol peut provoquer des variations de vitesse importantes que le PDC tentera de compenser via l'application de couples importants. De plus, les flexions-extensions et adductions-abductions de hanche du membre inférieur en phase oscillante n'étant pas contrôlées par le PDC, il est possible que leur trajectoire au moment où le PDC reprend leur contrôle soit suffisamment éloignée de la référence pour que les couples calculés provoquent un changement brusque dans la trajectoire des couples. Enfin, les couples issus des CC et SFC, bien que commençant à zéro, peuvent voir leur contribution augmenter rapidement après un changement de phase, au point de pouvoir provoquer un pic de couple. L'adaptation linéaire des couples, à partir de leur valeur au changement de phase jusqu'à la valeur prescrite par le contrôleur sur une période plus ou moins longue, permet alors de limiter ce phénomène. Plusieurs variations de cette solution ont été testées avant de s'arrêter sur l'une d'elle. La solution finale adapte les couples totaux de toutes les articulations d'une jambe lors du *toe-off* (TO) ou des deux jambes lors du *heel-strike* (HS). Une autre solution testée adaptait directement les couples dans les briques de contrôle en diminuant progressivement le couple calculé par l'ancien contrôle et en augmentant de manière inversement proportionnelle le couple du nouveau contrôle. Une autre solution n'adaptait que les couples les plus à risque de montrer des pics comme la flexion-extension et l'adduction-abduction de hanche et différentes combinaisons entre ces solutions ont également été testées. La solution finale a été retenue dans un souci de cohérence globale de traitement des différents DoF et de simplicité de compréhension du fonctionnement du contrôleur.

Une seconde série de solutions a consisté en l'utilisation de gains variables au cours du cycle. La nécessité d'utiliser des gains variables a d'abord été due au SFC. La position antéropostérieure et verticale du pied étant à la fois contrôlée par la flexion-extension de hanche et le genou, mais à des moments différents de la phase oscillante, utiliser des gains fixes n'aurait pas pu permettre de traduire cette variation du rôle des DoF. Un autre avantage à l'utilisation de gains variables est que cela permet de moduler l'action d'un DoF. Même sans changer fondamentalement de fonction, un DoF peut être plus utilisé à un moment spécifique du cycle de marche et moins à un autre. Dans ce cas, un gain fixe permettrait la bonne utilisation de ce DoF à un moment du cycle mais pas sur le reste. L'évolution des gains du SFC est déterminée par trois valeurs pour les mêmes raisons qui ont poussé à l'utilisation de gains variables. Il y a au minimum trois rôles que joue la flexion-extension de hanche au cours de la phase oscillante : avancer le pied en début de phase, gérer la hauteur du pied lorsque le tibia est

parallèle au sol et que le genou ne peut plus la contrôler, et enfin, de nouveau contrôler l'avancée du pied. Pour s'assurer que ces rôles puissent être remplis, autant de valeur de référence sont donc utilisées pour décrire l'évolution des gains. Idéalement, et dans un souci de cohérence entre les briques de contrôle, les gains du CC devraient également être décrits par trois valeurs. Cependant, cela aurait mené à un trop grand nombre de paramètres à optimiser, rendant la convergence de l'optimisation beaucoup plus compliquée. C'est pourquoi les gains du CC ne sont décrits que par deux valeurs. Dans la continuité de l'utilisation de gains pour améliorer le contrôle, le fait de les appliquer directement à chaque élément lié à une direction avec un gain différent selon la direction contrôlée permet non seulement de donner la priorité du contrôle d'une direction à un ou plusieurs DoF particuliers, mais aussi de faire en sorte qu'un DoF soit principalement utilisé pour une direction en particulier. En effet, si les gains étaient uniquement appliqués au calcul de l'amplitude des forces virtuelles, l'utilisation des jacobiniennes permettrait certes de n'utiliser un DoF que pour contrôler une direction qu'il peut effectivement contrôler, mais les éléments issus des autres directions pourraient perturber l'élément important, et surtout, tous les DoF pouvant agir sur une direction seraient utilisés de la même manière, même si, là aussi, modulé par l'utilisation des jacobiniennes. Enfin, le fait de pouvoir faire varier les gains peut également être utilisé pour améliorer les transitions de phase. Les couples dus au CC ne sont utilisés que pendant les phases d'appui. Cependant, leur valeur dépend de la différence entre le XCoM de référence et de la simulation. Le XCoM de référence étant adapté de la même manière que les trajectoires de DoF, une différence est déjà présente lors de la pose du pied et les couples résultants provoqueraient donc une discontinuité ou un pic de couple. En débutant et finissant la variation de ces gains à zéro, on s'assure alors que les couples issus du CC commencent et finissent également à zéro.

En utilisant ces différentes solutions pour adapter les couples articulaires, certains couples subissent des « doubles adaptations ». Cela est le cas à la pose du pied où non seulement les couples issus du CC commencent à zéro dû aux gains commençant à zéro, mais où en plus le couple total est adapté. C'est également le cas au lever du pied où les couples du SFC commencent à zéro grâce à l'adaptation des trajectoires, et où en plus une adaptation est appliquée au couple. Initialement, ces doubles adaptations étaient évitées pour ne pas introduire de redondance dans le contrôle. Cependant, il s'avère qu'une seule adaptation peut ne pas suffire à éviter les pics de couple et il a donc été décidé d'utiliser ces doubles adaptations.

Enfin, une dernière adaptation a été de définir directement dans le contrôle le rôle de certains DoF. C'est le cas de la rotation interne-externe de hanche qui n'est pas utilisée pour contrôler le pied en phase oscillante afin de plutôt assurer son orientation à la pose du pied. C'est également le cas de la flexion-extension du genou en phase oscillante qui, en fin de phase, a le rôle précis de tendre la jambe avant le contact du pied avec le sol. Pour assurer le passage d'un contrôle à l'autre, les différentes méthodes précédemment citées ont là aussi été testées. Mais contrairement à l'adaptation entre les phases, c'est la solution consistant à faire évoluer les couples de manière inversement proportionnelle entre l'ancien et le nouveau contrôle qui a été gardée pour cette transition car c'est elle qui présentait le moins de perturbations.

## B. Difficultés rencontrées

Tout au long du développement du simulateur, différentes difficultés ont été rencontrées. Certaines de ces difficultés telles que les problèmes d'algorithme, les erreurs de programmation informatique, l'implémentation de l'optimisation en calcul parallèle ou les tentatives d'implémentations infructueuses, étaient attendues et sont considérées comme propre à ce projet et normales. Cependant, d'autres difficultés rencontrées étaient à la fois suffisamment importantes pour gêner la conception du simulateur sur la durée et de manière plus globale, mais étaient également dues à des choix méthodologiques. Ces problèmes étant moins propres à ce projet, ils sont de fait plus facilement évitables avec l'expérience acquise, et sont présentés ci-après.

### 1. Modèle de contact

La principale difficulté rencontrée au cours du développement concerne le modèle de contact, que ce soit pour la modélisation des forces verticales ou horizontales. Le premier modèle de contact implémenté était celui utilisé dans les exemples d'OpenSim. Ce modèle comporte trois sphères de contact situées au talon et à la base des orteils. Ce modèle de contact était suffisant au modèle de bipède puisque celui-ci ne prenait pas en compte l'articulation métatarso-phalangienne. Cependant, le fait de ne pas modéliser de contact aux orteils avait pour conséquence une perte prématurée du contact et donc un changement de phase lui aussi prématuré. De plus, la force de réaction antéropostérieure était également diminuée par rapport à ce qui était attendu. Ce sont ces deux problèmes qui ont été déterminants sur le choix de placer des sphères de contact sur les orteils, et par nécessité, de considérer l'articulation métatarso-phalangienne.

Une fois ce choix acté, trois modèles de contact ont été implémentés et testés. Le premier s'inspirait du modèle de contact de base d'OpenSim en ajoutant une sphère de contact au bout du pied. La quatrième sphère avait les mêmes propriétés que l'une des sphères à la base des orteils et était à la même hauteur que les sphères à la base des orteils. Sa position sur le plan horizontal a quant à elle été placée manuellement de sorte que la base de support couvre le plus de surface du pied possible. Le deuxième modèle implémenté est celui présenté par Porsa et al. (Porsa et al., 2016). L'avantage de ce modèle est qu'ayant été développé via les classes d'OpenSim, il était directement applicable au modèle de bipède. De plus, les caractéristiques du modèle avaient été optimisées pour la marche et la course. Ce contrôleur ne devant simuler que la marche, il est possible que des caractéristiques déterminées à la fois pour de la marche et de la course ne soient pas les plus optimales possibles. Cependant, la méthode pour déterminer les caractéristiques du modèle de contact de base d'OpenSim utilisant trois sphères n'est pas explicitée et est utilisée dans un exemple de chute sur une surface. Il est possible que les caractéristiques du modèle de Porsa et al. soient plus adaptées à la marche que celles du modèle de base d'OpenSim. Enfin, le troisième modèle implémenté est celui présenté par Anderson et Pandey (Anderson & Pandey, 1999). La particularité de ce modèle par rapport aux deux autres est qu'il utilise des éléments

viscoélastiques plutôt que des géométries de contact, ce qui permet de facilement l'implémenter dans le contrôleur.

Le modèle de contact d'Anderson et Pandy est celui ayant donné les moins bons résultats puisqu'il ne permettait pas de tenir une position debout statique comme les deux autres. Plusieurs raisons peuvent expliquer cet échec. Premièrement, le modèle de contact n'était pas entièrement décrit et il est possible que la manière de l'implémenter dans le contrôleur n'ait pas été assez fidèle pour qu'il fonctionne. Et secondement, le modèle était utilisé pour une simulation de saut, et il est donc possible que les paramètres n'étaient tout simplement pas compatibles avec la tâche de la marche. Les deux autres modèles de contact permettant de réaliser un mouvement de marche, le modèle d'Anderson et Pandy a rapidement été abandonné au profit des deux autres afin de ne pas perdre de temps en détermination de paramètres corrects.

Pour déterminer le meilleur des modèles parmi les deux restants, leur utilisation a été alternée au cours du développement du contrôleur. Cependant, le passage d'un modèle à l'autre présentait un obstacle pour le développement. En effet, les sphères de contact n'ayant pas le même rayon, la hauteur de sa trajectoire de référence devait être adaptée à chaque changement de modèle. Au cours de ces utilisations alternées, aucun modèle ne semblait particulièrement plus performant que l'autre et les deux provoquaient les mêmes erreurs.

A la pose du pied, la force de réaction verticale formait un pic pouvant atteindre 2 N/Kg, là où la littérature montre que la force de réaction maximale ne dépasse généralement pas 1.2 N/Kg. Ce pic de force était alors suivi par une diminution brutale de la force avant de remonter avant la fin de phase de double appui. Par la suite, en phase de simple appui, la force de réaction pouvait continuer d'osciller plus ou moins fortement. Ce comportement ressemblant à une oscillation a été perçu comme un rebond du pied sur le sol. Pour tenter d'amoindrir ce phénomène, les paramètres de contact, notamment les rigidités et dissipations, étaient continuellement modifiées au cours du développement.

Les forces de réactions horizontales étaient originellement calculées via un modèle de friction de Coulomb. Au début du développement du simulateur, le glissement du pied sur le sol permis par le modèle n'était pas suffisant pour être considéré comme un problème à gérer en priorité. Dans la direction antéropostérieure, la pose du pied provoquait un pic important de force négative sur une courte durée. La force se stabilisait alors aux alentours de zéro avant de produire à nouveau un pic important de force positive sur une courte durée également. Il est probable que le pic de force initial était suffisamment important pour stopper rapidement le pied et ainsi éviter le glissement de celui-ci sur le sol. De même, les forces de réaction médio-latérales montraient un pic important dirigé latéralement à la pose du pied avant de se stabiliser.

Le travail sur le modèle de contact au début du développement visait donc surtout à améliorer la force verticale. Celui présentant le moins de rebond à la pose du pied était le modèle de Porsa et al. Pour fluidifier le développement du contrôleur, il a donc été décidé de définitivement choisir ce modèle spécifique pour le reste du développement.

Une amélioration significative des forces de réaction verticales n'a cependant pas été le résultat d'une modification des caractéristiques du modèle, mais celui de la prise en compte des couples

articulaires dans la détermination des paramètres du contrôleur. En effet, le réglage des paramètres se faisait initialement uniquement en étudiant la cinématique des résultats de simulation. Lorsque les couples ont commencé à être étudiés, il s'est avéré qu'ils étaient trop importants par rapport à ceux attendus. Des gains trop élevés avaient pour conséquence cette amplitude trop importante, mais il en résultait aussi des réponses trop fortes à des perturbations, provoquant des pics et irrégularités dans les courbes de couples. En prenant cet élément en considération, les paramètres ont été ajustés pour tenter de les diminuer. Lorsque les couples ont atteint des valeurs plus standards et que les irrégularités ont été amoindries, les irrégularités présentes dans les forces de réaction l'ont été également. Le premier pic de force verticale a fortement diminué, ainsi que le rebond qui suivait et les oscillations présentes en phase de simple appui.

Cependant, c'est également à partir de cet instant que les problèmes de glissement du pied ont commencé à apparaître. En effet, les pics de forces horizontales ont également diminué avec l'amélioration des couples articulaires. Cela a eu pour conséquence que le pied n'était plus stoppé aussi fortement et qu'une vitesse horizontale était toujours présente. Cette vitesse était trop faible pour produire une force de réaction capable de stopper le pied, mais suffisante pour que le glissement du pied soit significatif. Une hypothèse concernant les modèles de contact a alors été formulée. Les modèles de contact utilisant des géométries de contact et un modèle de friction de Coulomb ont leurs paramètres déterminés par optimisation à partir d'une cinématique connue (Brown & McPhee, 2018; Pàmies-Vilà et al., 2014; Shourijeh & McPhee, 2015). Lors de ces optimisations, les forces sont calculées à partir du mouvement, mais le mouvement n'est pas influencé par la force en retour. En simulation directe, les forces sont calculées à partir du mouvement mais le mouvement est alors influencé par ces forces en retour. Si le mouvement n'est pas parfait, alors les forces de réaction ne le sont pas non plus, et elles influenceront alors de façon péjorative le reste du mouvement. Il semblerait que le modèle de friction de Coulomb ne soit pas suffisamment performant pour la simulation en dynamique directe, ou en tout cas à la phase de développement d'un simulateur en dynamique directe. De plus, d'un point de vue microscopique, les forces de frottement sont causées par les irrégularités des surfaces agissant comme des arrêts. Un frottement sans glissement peut alors être assimilé à un système de ressort amortisseur. En adoptant l'hypothèse d'une marche sans glissement du pied, il est alors possible de modéliser les forces de réaction horizontales par des éléments viscoélastiques, permettant aux forces d'être calculées en fonction d'une position, et donc d'éviter un glissement continu du pied sur le sol. On retrouve là des principes du modèle d'Anderson et Pandy.

Comme montré dans la partie présentant les résultats, bien que les forces aient une forme globalement similaire aux courbes de la littérature, des imperfections sont tout de même présentes, et l'amplitude des forces horizontales est plus importante qu'attendu. Les imperfections sont notamment dues à l'imperfection du mouvement lui-même, que ce soit le pic de force vertical ou les irrégularités qui suivent. Quant aux amplitudes importantes des forces du modèle de contact, elles sont nécessaires pour assurer que le pied reste fixe. En effet, comme évoqué dans la partie présentant les résultats, il est possible de trouver par optimisation des paramètres permettant de reproduire plus fidèlement les forces de réaction horizontales, mais celles-ci ont tendance à laisser le pied glisser sur le sol. Cette incohérence est alors considérée comme la preuve d'un problème dans le contrôle et le mouvement résultant.

## 2. Méthode d'amélioration du contrôleur

Comme expliqué précédemment, des modifications devaient être apportées au contrôleur de base pour que celui-ci s'appuie sur des concepts biomécaniques et produise un résultat biofidèle. Cependant, une certaine série de modifications a évolué en une mauvaise méthodologie d'amélioration du contrôleur. Il est donc intéressant de comprendre pourquoi cette méthodologie était une mauvaise piste pour la suite du développement.

Pendant toute une partie du développement, les modifications apportées visaient principalement à améliorer les couples articulaires. Les problèmes ou différences avec les données de la littérature étaient traités dans l'ordre chronologique de leur apparition dans le cycle de marche et faisaient chacun l'objet d'une étude pour comprendre ses causes. Les résultats de ces études montraient généralement que le contrôleur n'était pas adapté ou incomplet pour correctement gérer cette spécificité de la marche, et la correction apportée pouvait alors être perçue comme l'ajout d'une exception au contrôle de base. Cependant, après avoir réglé un problème du mieux possible, un nouveau était alors découvert nécessitant à son tour une exception au contrôle de base. A force de corrections, le contrôle de base était de plus en plus parsemé d'exceptions au point d'ouvrir la voie à un nouvel objectif. Les exceptions au contrôle de base touchaient généralement au PDC et au suivi des trajectoires articulaires de référence. A force de voir cet aspect du contrôle être de moins en moins prédominant, l'idée de s'en passer le plus possible a fini par s'imposer dans la méthodologie de développement du contrôleur. Pour ce faire, le PDC serait remplacé par une série continue d'exceptions qui deviendraient alors des sortes de mini-briques de contrôle ayant chacune pour tâche de contrôler un aspect précis à un moment précis de la marche. L'avantage aurait premièrement été de pouvoir se passer le plus possible de données de référence, et donc d'adapter le mouvement au passage d'obstacle, de lacune, ou sur tapis roulant potentiellement plus facilement. Un second avantage était que pour fonctionner, l'entièreté de la marche devait être décrite par des fonctions remplissant un objectif précis. Poursuivre et atteindre cet objectif aurait donc permis de déterminer quelles étaient ces fonctions, et donc de mieux comprendre les mécanismes du contrôle de la marche.

Cependant, cet objectif s'est révélé impossible à atteindre dans le cadre de ce projet. La complexité du modèle avait pour conséquence que l'ajout d'une exception entraînait des modifications à d'autres endroits du modèle. Implémenter cet objectif aurait donc nécessité de comprendre toutes les interactions entre chaque exception. De plus, chaque ajout d'exception compliquait la compréhension du contrôle, ce qui aurait pu finir par desservir l'un des objectifs du simulateur, à savoir facilement modéliser des pathologies, puisque cela aurait alors potentiellement requis de revoir l'intégralité du contrôle.

Il a donc été décidé de retirer les adaptations faites pendant cette période et de trouver une autre méthode pour améliorer le contrôleur. C'est notamment à partir de cet instant que les modifications concernant les gains et les transitions de contrôle ont été implémentées ou améliorées.

### 3. Réglage manuel des paramètres

L'optimisation des paramètres est une étape importante du développement puisque c'est elle qui permet de déterminer les paramètres à utiliser une fois le contrôle défini. Le processus d'optimisation commence par la détermination d'une solution initiale la plus proche possible du résultat attendu. Cela permet à l'optimisation de trouver plus rapidement une solution optimale et potentiellement d'éviter des optimaux locaux. Dans le cas de ce simulateur, une solution initiale convenable consistait en la réalisation d'une foulée entière sans que le modèle ne tombe complètement. Cette solution initiale doit se trouver manuellement et passe donc par un processus d'essai erreur. Encore une fois, la difficulté rencontrée provenait de la complexité du modèle. Malgré de nombreux essais aucun réglage manuel ne produisait de solution initiale acceptable.

La situation avançant difficilement, il a été décidé de commencer le processus d'optimisation à partir d'une solution initiale moins performante. Celle-ci ne se déroulant que sur un pas, les valeurs du CC étaient toutes mises à zéro, et seuls quelques gains du SFC étaient non nuls pour permettre au pied en phase oscillante d'être contrôlé un minimum. Une particularité de cette solution initiale est que le pied en phase oscillante montait trop haut et se posait trop loin et trop tard pour pouvoir entamer un second pas de manière stable. Comme évoqué dans la partie sur les résultats, la toute première optimisation réalisée a servi de test et visait à optimiser les paramètres du SFC et du modèle de contact. Mais celle-ci comportait une erreur qui avait pour conséquence de réduire l'amplitude de recherche pour les paramètres du SFC à un ordre  $10^0$  autour de la solution initiale alors que les paramètres du SFC avaient un ordre de grandeur de  $10^2$ . Pourtant, malgré cette erreur, l'optimisation a tout de même réussi à améliorer la trajectoire du pied en phase oscillante. Ce résultat a donc permis de montrer que le contrôle était en fait extrêmement sensible aux paramètres, et que la recherche manuelle d'une solution initiale aussi avancée que celle souhaitée était impossible.

## C. Perspectives

Tous les objectifs n'ayant pas été atteints, le simulateur peut continuer à être développé et amélioré afin de fournir une dynamique plus biofidèle et pour implémenter la simulation de pathologies.

### 1. Amélioration du contrôle de base

Bien que les critères de contrôle de l'équilibre initialement formulés soient respectés, la cinématique globale et les couples articulaires qui en découlent ne sont pas toujours biofidèles. Des pics de couple peuvent survenir, notamment lors de changements de phase, ou la forme globale des courbes

peut ne pas correspondre avec celles de la littérature provenant de la dynamique inverse et provoquer des problèmes dans la cinématique comme avec l'inclinaison sagittale du pelvis. Une piste d'amélioration du contrôleur serait donc de chercher les modifications nécessaires à l'obtention de couples articulaires biofidèles et permettant de respecter la cinématique globale.

### 1.1. Modèle de contact

Théoriquement, si le contrôleur permet une cinématique similaire à celle de la littérature avec un modèle aux propriétés similaires à celles d'un humain, alors c'est que la dynamique doit elle-même être similaire à la littérature, au moins dans la forme globale des courbes. Si ce n'est pas le cas, une raison possible est une mauvaise modélisation des forces de contact. En effet, le comportement du modèle est en partie déterminé par les forces de contact. Si celles-ci ne sont pas modélisées correctement, alors le modèle ne se comportera pas complètement comme attendu, et le contrôleur, en devant corriger cette différence de comportement, produira des couples articulaires différents de la littérature.

Une première cause possible de mauvaise modélisation des forces de contact est le modèle lui-même. Le modèle de Hunt-Crossley utilisé dans ce simulateur a été choisi car nativement présent dans OpenSim. Cependant, comme évoqué dans l'état de l'art, la modélisation du contact est un sujet de recherche à part entière. D'autres modèles ayant pour objectif de mieux prendre en compte les propriétés des matériaux ou de déterminer de meilleures équations décrivant le contact ont été développés. Tester ces modèles, bien que plus compliqués à implémenter dans le contrôleur, pourrait permettre d'améliorer la modélisation des forces de contact. De même, les forces horizontales sont modélisées par de simples systèmes de ressorts amortisseurs. Bien que l'utilisation d'un élément basé sur la position en plus d'un élément basé sur la vitesse semble pertinente pour éviter les glissements du pied, il est probable qu'un simple modèle de ressort amortisseur linéaire ne soit pas suffisant pour correctement modéliser les forces transversales. Par exemple, pour éviter que la vitesse du pied lors du contact ne provoque une discontinuité des forces horizontales, un coefficient est appliqué à la force pour s'assurer qu'elle commence à zéro. Il serait cependant possible de s'inspirer de la méthode utilisée pour éviter ce phénomène dans le calcul des forces verticales en liant la force calculée par la dissipation à la distance entre le point de référence et le point d'application des forces. Ainsi, l'élément limitant la force de réaction ne serait pas issu d'une fonction arbitraire mais d'une réalité liée à l'état du modèle. De même, le calcul des forces horizontales utilise actuellement une formule linéaire alors que les propriétés réelles du contact et que le calcul des forces de Hunt-Crossley ne le sont pas. La formule pour calculer les forces horizontales peut donc être modifiée.

Une seconde cause possible de mauvaise modélisation des forces de contact est la détermination des paramètres du modèle. La détermination des paramètres de contact dans la

littérature se fait généralement par optimisation faisant appel à un mouvement et des forces de réactions connues (Brown & McPhee, 2018; Pàmies-Vilà et al., 2014; Shourijeh & McPhee, 2015). Le résultat est un modèle adapté à un mouvement expérimental et pouvant servir lors de calculs de dynamique inverse si les forces de contacts ne sont pas connues. Cependant, le contrôle en dynamique directe fait appel à nombre d'hypothèses simplificatrices résultant en un mouvement ne présentant alors pas toutes les subtilités d'un mouvement expérimental ou avec des couples non biofidèles comme dans ce simulateur. Les paramètres des éléments viscoélastiques dans ce simulateur ont également été déterminés par optimisation. Cependant, l'optimisation en question ne portait pas directement sur le respect des forces de contact puisque comme l'a montré le résultat de l'une des optimisations, optimiser les paramètres du modèle pour que les forces calculées correspondent au modèle statistique génère des forces permettant au pied de glisser sur le sol. Cependant, utiliser des valeurs trop élevées pour assurer le blocage des pieds provoquerait une oscillation du pied sur le sol qui impacterait le reste de la cinématique du modèle, et donc, sa dynamique. Il serait donc intéressant de déterminer une fonction de coût pouvant améliorer les paramètres de contact sans détériorer la cinématique du reste du modèle.

## 1.2. Transitions de phases

Les transitions de phases représentent des moments particulièrement prompts aux problèmes de biofidélité des couples articulaires, notamment lorsque le contrôle des DoF passe du PDC au SFC et inversement. A ces moments-là, les couples avant et après le changement de phase ne sont pas les mêmes et les discontinuités sont évitées en passant linéairement des couples au changement de phase aux couples du nouveau contrôle sur une durée donnée. Cependant, les couples prescrits par le nouveau contrôle sont calculés de sorte à traquer des trajectoires de référence. Dès lors, ne pas directement appliquer le couple prescrit fait que les DoF ou la position du talon convergent moins vite vers leurs trajectoires de référence. Le nouveau contrôle a alors tendance à surcompenser plus longtemps provoquant un pic de couple sans discontinuité et moins important que sans adaptation, mais sur une plus longue période.

Une première solution pour éviter ces pics de couple serait d'améliorer le système d'adaptation. Tel quel, le simulateur permet de converger rapidement vers les nouvelles trajectoires de référence au prix d'un pic de couple articulaire. Il pourrait être envisageable d'utiliser une méthode d'adaptation différente provoquant moins de pics de couples articulaires au prix d'un moins bon suivi des trajectoires de référence en début de phase comme avec la minimisation des premiers gains du SFC. Un juste équilibre serait alors à trouver entre les aspects continuités des couples et suivi des trajectoires.

Une seconde solution serait d'utiliser la cinématique inverse plutôt que les forces virtuelles pour contrôler la position du talon. La jambe en phase oscillante serait alors contrôlée par le PDC sans changement de contrôle lors du changement de phase, et donc sans discontinuité. La jambe en phase oscillante était initialement contrôlée par le PDC via des trajectoires déterminées par cinématique inverse jusqu'à ce qu'un essai démontre que les forces virtuelles permettaient de mieux traquer la trajectoire du talon. La priorité ayant été placée sur la précision de la position de la pose du talon, et

donc sur la précision du suivi de la trajectoire, la méthode de contrôle du talon par forces virtuelles a finalement été gardée. Cependant, cet essai a été effectué avant les premières optimisations. Il est donc possible que le contrôle de la jambe en phase oscillante par cinématique inverse avec des paramètres optimisés permette d'obtenir un contrôle de précision équivalente au contrôle par forces virtuelles. Si cela s'avère être le cas, alors changer de méthode de contrôle améliorerait grandement la qualité de la simulation.

Enfin, il semblerait que la perte de contact du pied après le *heel strike* (HS) provient majoritairement du contrôle. En effet, une fois le contact établi et le changement de phase effectué, le contrôle de la jambe passe entièrement au PDC. Cependant, cette solution semble ne pas être suffisamment compliant et le mouvement dicté par le PDC provoque alors une élévation du pied et une perte de contact. Il serait donc intéressant de déterminer une méthode de transition de phase capable de corriger ce problème. De plus, les résultats semblent montrer que le transfert du poids lors du double appui ne s'effectue pas correctement. Il est possible que cette imperfection soit en partie à l'origine des mauvaises forces de réaction lors des phases de double appui, notamment dans la direction médio-latérale.

### 1.3. Trajectoires en plusieurs parties

Certaines courbes de couple articulaire montrent une variation brusque à des moments correspondant à la transition entre les deux splines de la trajectoire. Comme évoqué précédemment, lorsque la spline décrivant une trajectoire de référence change, la continuité des valeurs et de la première dérivée est assurée en utilisant les conditions de la spline précédente comme conditions initiales de la nouvelle spline, que ce soit lors du passage à un nouveau cycle où lorsque la trajectoire de la courbe est séparée en deux parties pour assurer la valeur de la dérivée à un instant donné. Cependant, lors de cette transition, la seconde dérivée n'est, elle, pas continue, et provoque des changements brusques de la vitesse de référence, se répercutant en changement brusque des couples articulaires. Cet effet étant inévitable lors du passage d'un cycle au suivant, le contrôleur doit y être robuste. Ainsi, la perturbation a également été jugée acceptable pour assurer la valeur de dérivée à des instants précis de certaines trajectoires. Il serait cependant intéressant d'investiguer sur la capacité du modèle statistique à produire des trajectoires de référence n'utilisant qu'une seule spline sans détériorer la cinématique pour diminuer au maximum ces perturbations supplémentaires, notamment au niveau de la trajectoire verticale du talon en phase oscillante.

### 1.4. Gains du PDC

Comme évoqué précédemment, les gains actuels du PDC ont été déterminés par optimisation. Ils sont ensuite restés inchangés lors des optimisations suivantes pour diminuer le nombre de paramètres à optimiser et pour mieux percevoir les changements apportés par les différentes optimisations.

Cependant, en procédant ainsi, les gains du PDC ne sont plus optimaux par rapport aux paramètres des autres briques de contrôle. De plus, les gains des PDC sont généralement choisis comme étant les plus faibles possibles, plus une marge de sécurité, pour assurer le suivi des trajectoires sans risquer d'erreurs numériques. Dans ce simulateur, le PDC fonctionnant conjointement avec d'autres briques de contrôles, les contrôles antagonistes pour un DoF sont fréquents lorsque les autres briques de contrôle tendent à trop éloigner la trajectoire du DoF de sa trajectoire de référence. Cela peut générer des briques de contrôle prescrivant des couples d'amplitude trop élevés par rapport à la littérature pour contrebalancer les couples des autres briques. De plus, si une brique doit régulièrement fortement contrer les prescriptions du PDC, les paramètres de cette brique, trouvés par optimisation, risquent d'être inutilement élevés par rapport aux besoins du reste du cycle, provoquant ainsi des réponses trop fortes à ces phases. Il serait donc intéressant d'effectuer des optimisations ayant pour objectif d'optimiser conjointement les gains des différentes briques de contrôle en les diminuant au maximum et ainsi diminuer ce phénomène.

### 1.5. Position de la pose du pied

Actuellement, la position de la pose du pied est invariable. La gestion de la vitesse normalement attribuée à l'adaptation de la longueur de pas est alors entièrement contrôlée par les CC. De même, l'équilibre médio-latéral normalement contrôlé par la largeur de pas est également laissé au CC en faisant l'hypothèse qu'il est assez efficace pour suivre précisément la trajectoire de XCoM de référence.

Autoriser la modification de la position d'arrivée du pied dans la direction médio-latérale permettrait à la simulation d'être plus robuste aux perturbations. Cependant, cela demande de déterminer une méthode limitant la perturbation de cette modification sur la cinématique du pied. Pour conserver la cinématique globale de la jambe et son effet sur le reste du modèle, il serait possible de n'adapter la position d'arrivée qu'à partir de la fin de phase oscillante mais la modification devrait alors se faire sur un temps trop court et ne serait pas forcément réalisable. Il serait alors possible de modifier dès le début de la phase oscillante le point d'arrivée du pied en fonction de la différence entre le XCoM de référence et de la simulation. Cependant, si la trajectoire du XCoM de la simulation connaît trop d'irrégularités, modifier continuellement le point d'arrivée du pied en fonction de la position du XCoM de la simulation risque de perturber à son tour la trajectoire du pied.

La position d'arrivée du pied dans la direction antéropostérieure devra également être modifiable, ne serait-ce que pour pouvoir adapter la longueur de pas avant de passer un obstacle. Cependant, comme évoqué précédemment, la longueur de pas est liée à la vitesse de marche et au temps de pas. Pour pouvoir autoriser la longueur de pas à être modifiée, il faudra donc déterminer quel impact cette modification aura sur les paramètres spatio-temporels et sur la cinématique et la dynamique du mouvement. De plus, la correction de l'inclinaison sagittale du pelvis impactera probablement le suivi du XCoM dans la direction antéropostérieure. Si celui-ci n'est plus à même de suffisamment contrôler cette avancée pour fixer à la fois la longueur et le temps de pas, alors il sera indispensable d'utiliser la longueur de pas comme contrôle complémentaire à la vitesse de marche.

## 1.6. Orientation globale du modèle

Comme évoqué précédemment, l'orientation globale du modèle n'est pas directement contrôlée et est utilisée comme indicateur de qualité du contrôleur. L'une de ces orientations est la rotation du pelvis et détermine dans quelle direction fait face le modèle. Un problème pouvant apparaître si le contrôle permet d'enchaîner plus de pas est l'accumulation de l'erreur sur la rotation axiale droite-gauche du pelvis. En effet, il est attendu que le pelvis ne suive pas ses trajectoires de référence aussi fidèlement que les autres DoF. Mais n'étant pas directement contrôlé, rien n'empêchera alors une erreur sur la rotation axiale droite-gauche du pelvis de s'accumuler au cours des pas. Or, si le pelvis ne fait plus face à la direction antéropostérieure alors que les trajectoires du CoM et du pied en phase oscillante sont dirigées sur cette direction, alors le contrôle ne sera pas stable. Un DoF agissant tout particulièrement sur la rotation axiale droite-gauche du pelvis est la rotation interne-externe de hanche. Ainsi, si le pied est orienté convenablement, alors le suivi de la trajectoire de référence de la rotation interne-externe de hanche devrait permettre de limiter les erreurs sur l'orientation globale du pelvis. Actuellement, la rotation interne-externe de hanche du membre inférieur en phase oscillante est contrôlée par le PDC justement pour s'assurer que le pied arrive sur le sol avec une bonne orientation. Cependant, cette trajectoire étant relative au pelvis, la justesse de l'orientation du pied part de l'hypothèse que le suivi de la trajectoire du pelvis est suffisamment précis. Les derniers résultats montrant que ce suivi n'est pas suffisamment précis et qu'une erreur peut s'accumuler, cette solution n'est finalement pas optimale. Une solution pour assurer la bonne orientation du pied lors du contact avec le sol, et donc par la suite d'un meilleur suivi de l'orientation du pelvis, consisterait à contrôler la rotation interne-externe de hanche en phase oscillante non pas par rapport au pelvis, mais directement par rapport à son orientation dans le repère global.

## 1.7. Modèle statistique basé sur la dynamique

Une dernière piste d'amélioration serait de changer radicalement d'approche concernant la « solution initiale ». Actuellement, la solution initiale qui permet de produire un mouvement de base suffisamment proche du mouvement souhaité est basée sur la cinématique via des trajectoires de référence traquées par un PDC. Il pourrait être envisageable de la remplacer par une solution basée sur la dynamique en utilisant un modèle statistique non pas pour déterminer des trajectoires angulaires de référence mais des trajectoires de couples articulaires de référence. Théoriquement, cette méthode présenterait plusieurs avantages. Premièrement, cela permettrait d'assurer que les trajectoires des couples aient une forme similaire à celles de la littérature. Les couples calculés par le contrôle du CoM et pour le contrôle du talon en phase oscillante seraient quant à eux toujours déterminés via la cinématique pour adapter les courbes initiales des couples. Deuxièmement, cela permettrait d'éviter les contrôles antagonistes entre le PDC et le contrôle du CoM. Comme évoqué précédemment, les couples du PDC et du CC sont additionnés. Cependant, si le CC tend à trop faire s'éloigner un DoF de sa trajectoire de référence, le PDC, se basant sur la cinématique, compensera de plus en plus et amoindrira les corrections du CC. Cependant, si la solution initiale était une trajectoire de couple invariable, le

couple additionnel produit par le CC ne provoquerait pas de contrôle antagoniste, évitant tous les problèmes cités précédemment. Enfin, la dérivée de la spline n'ayant pas à être utilisée, la discontinuité de la seconde dérivée lorsque la trajectoire est décrite par plusieurs splines ne serait plus un problème. Les trajectoires de couples pourraient alors plus facilement être décrites par plusieurs splines et donc être décrites de manière plus précise que les trajectoires d'angles articulaires. Cependant, cette méthode comporte tout de même une limitation importante. Si les forces de contact calculées lors de la simulation sont trop éloignées de la réalité, alors les trajectoires de référence des couples articulaires pourraient ne pas être suffisamment adaptées, rendant la simulation plus instable que via un contrôle par la cinématique. De même, les corrections appliquées par le CC pour compenser ces instabilités pourraient fortement modifier la trajectoire des couples articulaires, ce qui annulerait une partie des bénéfices de l'utilisation de couples de référence.

## 2. Implémentation des pathologies

L'objectif initial du simulateur étant de simuler des passages d'obstacle, de lacune et de montée/descente de tapis roulant par des sujets présentant une marche pathologique, la dernière étape consistera à implémenter des solutions permettant de reproduire des pathologies. Celles-ci peuvent aussi bien être d'origine musculaire que neurologique et une partie importante de cette étape sera de déterminer comment reproduire une pathologie dans le contrôleur.

### 2.1. Faiblesse musculaire

Une première pathologie à implémenter serait la faiblesse musculaire. Celle-ci peut notamment être liée à l'âge et renvoie à la problématique de l'accessibilité des personnes âgées. L'avantage de cette pathologie dans le cadre du développement du simulateur est qu'elle est intuitive à implémenter. Deux solutions peuvent être implémentées pour simuler cette pathologie. Tout d'abord, il est possible de limiter l'amplitude maximum des couples articulaires pour les faire correspondre à celles d'une personne présentant cette pathologie. Mais il est également possible de limiter la vitesse à laquelle évoluent les couples pour limiter l'explosivité des muscles. Bien que ces limitations puissent être appliquées à la marche saine pour notamment éviter les changements brusques des couples articulaires actuellement présents dans les simulations, elles ne le sont pour l'instant pas. En effet, les trajectoires de couples articulaires lors de la marche saine ne sont pas censées être affectées par ces limitations, et améliorer la trajectoire des couples de cette façon ne ferait que s'attaquer aux symptômes et non aux causes du problème. En revanche, elles seront toutes indiquées pour reproduire des faiblesses musculaires.

## 2.2. Amplitude articulaire réduite

Une deuxième pathologie à implémenter est la réduction des amplitudes articulaires. Elles peuvent être dues à des douleurs articulaires ou des traumatismes. L'implémentation de cette pathologie est également intuitive et deux solutions peuvent être implémentées pour simuler cette pathologie. Tout d'abord, il serait possible de modifier les trajectoires de référence pour s'assurer qu'elles ne dépassent pas les amplitudes maximales. En effet, un sujet atteint de cette pathologie évitera naturellement des mouvements le faisant atteindre l'amplitude maximale, et donc directement modifier le mouvement de référence serait cohérent. Ensuite, il serait également possible d'implémenter un contrôle empêchant de dépasser une valeur d'angle articulaire en appliquant un couple supplémentaire pour faire office de butée. De manière similaire aux limitations des couples articulaires, ces limitations ne sont pas actuellement présentes dans le contrôleur car la marche saine n'est pas censée atteindre ces limites.

## 2.3. Boiterie

Le fait de boiter peut être une conséquence des pathologies précédentes mais requiert des solutions spécifiques pour être simulé. La caractéristique principale d'une boiterie est l'asymétrie de la cinématique et des paramètres spatiotemporels de la marche entre les jambes. Le simulateur considérant actuellement la marche comme symétrique, il devra être adapté pour contrôler chaque côté via des trajectoires de référence différentes, nécessitant potentiellement un nouveau modèle statistique dans lequel le degré d'asymétrie serait une variable indépendante.

## 2.4. Marche avec une prothèse

L'impact de l'amputation d'une partie du membre inférieur et de son remplacement par une prothèse est la perte de contrôle d'une ou plusieurs articulations. Simuler la marche avec une prothèse revient donc à modifier le contrôle des DoF perdus pour qu'ils reproduisent le comportement passif des articulations de la prothèse. La complexité dans le fait d'implémenter ces nouveaux contrôles réside dans la pluralité des méthodes de conceptions des prothèses, et donc des comportements pour les différents DoF.

Les prothèses de cheville sont généralement des articulations élastiques permettant d'emmagasiner de l'énergie à la pose du pied pour la restituer lors de la propulsion et compenser ainsi en partie la perte de muscle contrôlant l'articulation. Ce comportement pourrait être simulé via un PDC dont la position de référence est fixe et la vitesse de référence est nulle comme cela a été déjà fait pour l'articulation métatarso-phalangienne dans le simulateur. Les gains seraient alors déterminés par les caractéristiques de la prothèse. Cependant, certaines prothèses ne possèdent pas d'articulation usuelle et font plutôt appel à des éléments déformables. La simulation utilisant un modèle de corps rigides, un

travail devrait être effectué pour déterminer la meilleure solution à adopter pour simuler ce segment et son contact avec le sol.

Les prothèses de genou ont également différentes conceptions. La plus simple consiste en une liaison pivot non contrôlée qui nécessite d'être freinée et bloquée avant la pose du pied pour que le genou soit fixé durant la phase d'appui. Ce blocage se faisant via un balancement en avant de la prothèse, implémenter cette solution requerrait d'adapter la cinématique de la jambe pour intégrer ce balancement, ainsi qu'un contrôle du genou basé sur la détection des événements le faisant passer de non contrôlé à verrouillé et inversement. Un autre type de genou prothétique consiste à contrôler l'articulation via un microprocesseur et un servomoteur pour permettre au genou de suivre une trajectoire plus biofidèle, d'être plus résistant lors de la pose du pied, et donc théoriquement d'améliorer la marche. Dans ce cas, le contrôle du genou devrait mimer celui du microprocesseur.

Enfin, des modifications devront être apportées au modèle lui-même. Les caractéristiques des segments humains devront être remplacées par celles des segments de la prothèse. De plus, une amputation transtibiale ou transfémorale a également pour conséquence de diminuer la taille des muscles agissant sur les articulations proximales, et donc sur la puissance maximale qu'ils peuvent développer. Les solutions évoquées dans la partie évoquant les faiblesses musculaires peuvent alors être appliquées pour prendre en compte cet aspect de la marche avec amputation.

## 2.5. Troubles neurologiques

Enfin, il est possible de simuler des troubles neurologiques en dégradant les parties du contrôleur gérant l'équilibre. Il serait par exemple possible d'altérer la position d'arrivée du pied pour simuler une erreur dans son appréciation. Les gains ou les forces virtuelles utilisés pour suivre les trajectoires du XCoM et du pied pourraient également être altérés pour simuler un problème de coordination.



# VII. Conclusion

## A. Complexité du contrôle

Le résultat principal qui ressort de ce travail est la complexité d'un contrôle basé sur des principes biomécaniques. En effet le contrôle de base était capable de réaliser plusieurs pas avant de tomber, et ce, même sans optimisation. C'est en modifiant le contrôle pour lui faire suivre une cinématique et une dynamique plus biofidèles qu'il a perdu en stabilité au point de ne plus pouvoir réaliser que deux pas après optimisation, et en terminant le second pas avec une cinématique dégradée.

Une première explication possible est que cela est dû à une sur-régulation du contrôleur. Afin de fournir un résultat biofidèle, le contrôleur est conçu pour suivre le plus fidèlement possible différentes données de référence, à savoir les trajectoires articulaires, les trajectoires du pied et les trajectoires du CoM, ainsi que des paramètres spatio-temporels tels que la longueur et le temps de pas. Ces trajectoires sont suivies grâce à l'action de couples articulaires appliqués aux DoF. Cependant, certains DoF doivent être simultanément utilisés pour suivre plusieurs trajectoires et il est donc possible que les couples prescrits par différentes briques pour un même DoF soient antagonistes. L'un des rôles du processus d'optimisation était justement de déterminer les priorités à donner aux différentes briques selon l'avancée du cycle de marche. Le fait que l'optimisation ne soit pas capable de déterminer des paramètres capables de produire une marche stable peut être alors considéré comme une preuve que le contrôleur dans son état actuel n'en est tout simplement pas capable.

Une seconde explication possible est que la complexité du contrôle facilite le fait de trouver des minimums locaux lors de l'optimisation. En effet, lors des optimisations, les priorités à donner sont exprimées dans les grandes lignes via la solution initiale mais étant donné le nombre de termes à respecter, les fonctions de coût n'empêchent pas complètement l'obtention de gains élevés pour ceux que l'on voudrait minimiser. Des fonctions de coût plus orientées pourraient contraindre l'optimisation à privilégier certaines solutions et empêcher de tomber dans certains minimums locaux.

On peut prendre comme exemple le résultat actuel du simulateur dans lequel le pelvis se penche trop en avant lors du deuxième pas. La cinématique du mouvement montre que cela est dû à la flexion trop importante de la hanche d'appui et la dynamique montre que la différence de couple entre le premier et le deuxième pas est surtout marquée par l'action du CC. Le couple de flexion-extension de hanche prescrit par le CC est intégralement dû à la force virtuelle antéropostérieure. Cette force contrôle également grandement la flexion-extension de genou et la dorsi-plantiflexion de cheville et une solution aurait pu être de minimiser davantage les gains appliqués à la flexion-extension de hanche, quitte à compenser par une augmentation des gains du genou et de la cheville. Cependant, réduire manuellement ce couple a pour résultat un moins bon suivi de la trajectoire médio-latérale du CoM et l'optimisation n'est pas non plus capable de le réduire sans altérer le suivi du XCoM. Peut-être qu'une

fonction de coût plus orientée serait à même de corriger ce problème ou peut-être que cela nécessite de modifier directement le contrôleur.

La complexité du développement du contrôleur démontre finalement la complexité intrinsèque du contrôle de la marche chez l'humain. La faisabilité d'un tel contrôle pour un résultat cinématique biofidèle n'est donc pas encore avérée mais montre tout de même des résultats intéressants et une modification plus ou moins partielle du contrôleur pourrait donner de meilleurs résultats.

## B. Modèle de contact

L'une des plus grosses limitations à ce simulateur est le modèle de contact. La précision requise pour obtenir des couples articulaires biofidèles est difficilement atteignable du fait des irrégularités des forces de réaction qui influent directement sur l'état du modèle et donc indirectement sur le contrôle. La principale difficulté en dynamique directe vient du fait que les forces de réaction calculées sont elles-mêmes influencées par le contrôle. Il peut donc être difficile d'identifier la source d'une irrégularité particulière et déterminer les paramètres avec simultanément tous les termes de la fonction de coût déjà existants plus des termes pour améliorer les forces de réaction risque d'être trop compliqué pour l'optimisation. Ce projet portant sur le contrôle et non sur la modélisation du contact, les modifications qui lui ont été apportées étaient limitées. Il pourrait cependant être nécessaire de faire de la détermination d'un modèle de contact compatible avec cette méthode de simulation une priorité avant de continuer le développement du simulateur trop en avant.

## C. Spécialisation du modèle

La complexité du contrôle amène également une potentielle future limitation à son développement. En effet, les différentes modifications et adaptations appliquées au contrôle de base ont été faites avec un objectif de marche. De plus, comme évoqué précédemment, le résultat semble être sensible aux paramètres du contrôle et les optimisations ont également été réalisées pour une tâche de marche. Or, un modèle trop spécialisé pour une tâche donnée pourra avoir du mal, voire être incapable d'être utilisé pour une tâche différente. Etant donné les difficultés rencontrées pour implémenter des modifications capables de simuler la marche classique et pour déterminer des fonctions de coût suffisamment performantes, il est probable que le passage d'une simulation de marche à une simulation de passage d'obstacle ou la prise en compte de pathologies doive se faire par une modification importante du contrôle. Cependant, cette particularité n'est pas forcément une mauvaise chose en soit et peut au contraire être vue comme un résultat à part entière.

## D. Avantages de la méthode de contrôle

De même que pour la tâche de la marche, la faisabilité et l'utilité d'un simulateur utilisant ce type de contrôleur pour simuler la marche pathologique lors de passages d'obstacles ne peuvent pas être avérées avec les résultats obtenus.

Cependant, un avantage de ce type de contrôleur réside dans son développement. En effet, les briques de contrôle utilisées sont intuitives dans le rôle qu'elles jouent. Développer un contrôleur avec cette méthode de contrôle demande donc de déterminer comment ces briques doivent fonctionner et donc de déterminer le rôle de chaque élément du corps dans la propulsion, le maintien de l'équilibre ou d'autres objectifs. Un contrôleur fonctionnel est alors une preuve que ces rôles ont été correctement déterminés et implémentés. Cela permet d'étudier les mécanismes de la marche via une approche différente de ce qui est généralement fait dans le domaine de la biomécanique où l'on détermine d'abord l'action d'un muscle via optimisation par dynamique inverse ou dynamique directe suivant un mouvement de référence, puis en étudiant les activations nécessaires au mouvement. Par exemple, il est difficile de se passer d'un contrôle prescrivant une solution initiale (PDC) pour le remplacer par des contrôles plus spécifiques et à court terme comme présentés en section IV.B.2. Cela peut être considéré comme indice supplémentaire du fait que le contrôle de la marche chez l'être humain se fait également par adaptation d'un mouvement de base. Un autre exemple concerne l'utilisation de la hanche en phase de simple appui. Dans le simulateur, elle est à la fois utilisée par le contrôle du CoM, c'est-à-dire pour le contrôle de l'équilibre et de la vitesse pendant un pas, mais aussi par le contrôle du talon en phase oscillante. Les résultats des optimisations ont montré que la hanche est simultanément utile à ces deux aspects du contrôle. Cependant, les résultats actuels ne permettent pas de terminer une foulée dans une position correcte. Cela permet de savoir que le contrôle de la hanche est soit incomplet, soit pas encore correctement paramétré. Dans les deux cas, il est possible d'améliorer notre compréhension des stratégies en jeu. En effet, s'il manque un contrôle agissant sur la hanche, alors le meilleur moyen d'améliorer le contrôleur est d'identifier correctement le mécanisme manquant. Et si le problème vient des paramètres, alors en déterminant les bons paramètres il sera possible de savoir quel DoF est utilisée pour quel objectif puisque les couples agissant sur les différents aspects de l'équilibre proviennent de contrôles spécifiques à ces aspects.

De même, les modifications à réaliser dans le contrôleur pour passer d'une tâche de marche à des tâches différentes peut également permettre de gagner des connaissances sur les mécanismes permettant d'adapter la marche chez l'humain. La quantité et le type de modifications à apporter serait un indice de l'effort que l'on doit produire pour adapter le mouvement. Réussir à adapter un passage d'obstacle en ajustant simplement le point d'arrivée du pied serait la preuve que le mouvement en question ne demande que peu d'adaptations chez l'humain aussi. Réussir à adapter un passage d'obstacle en devant modifier les gains serait la preuve du besoin d'une adaptation supplémentaire chez l'humain, voire d'un changement plus important de stratégie si les gains adaptés au passage d'obstacle sont fondamentalement différents de ceux de la marche. Enfin, la nécessité de changer directement des briques dans le contrôleur serait alors une preuve que la stratégie utilisée est radicalement différente.

## E. Contributions

Bien que le contrôleur ne parvienne pas encore à produire une marche stable, son développement permet tout de même de tirer plusieurs conclusions.

Premièrement, il est possible que la sur-régulation du modèle pour produire un mouvement biofidèle soit un obstacle majeur à sa stabilité. En effet, c'est après avoir introduit des éléments pour produire une marche biofidèle que le contrôle a perdu en stabilité. Il se peut notamment que le suivi simultané et précis des trajectoires du CoM et du talon demande un paramétrage trop précis et donc trop peu stable aux instabilités. Une solution serait alors de permettre des imprécisions sur les suivis des trajectoires pour permettre aux optimisations de déterminer de meilleurs ensembles de paramètres. Le problème revient alors à savoir quelles trajectoires il est possible de suivre moins précisément et à quel point. Par exemple, le suivi de la trajectoire médio-latérale du talon immédiatement après le TO posait initialement problème car le contrôle provoquait des oscillations du talon autour de sa trajectoire de référence et diminuer les couples pour éviter ces oscillations détériorait le reste du suivi. Cependant, le suivi de cette trajectoire en tout début de phase oscillante peut être considéré comme non prioritaire permettant aux DoFs contrôlant la trajectoire d'être utilisés pour autre chose. Au contraire, la trajectoire verticale du talon doit être précise. Une trajectoire biofidèle fait passer le talon proche du sol et une imprécision de contrôle à ce moment-là risquerait de provoquer le contact du pied avec le sol.

Deuxièmement, l'utilisation des forces virtuelles dans le cadre du contrôle d'un bipède est limitée. Le résultat de l'application des forces virtuelles sur la chaîne cinématique peut différer selon que la chaîne est ouverte (phase de simple appui) ou fermée (phase de double appui). De plus, la non prise en compte des segments se trouvant de l'autre côté de la chaîne cinématique entre les DoFs utilisés et le point d'application des forces peut également altérer l'action des forces virtuelles sur le modèle. La solution adoptée dans ce travail est d'autoriser l'utilisation de gains négatifs. Ainsi, la brique du CC n'utilise pas le vrai principe de force virtuelle mais plutôt un principe basé sur les jacobiniennes.

Troisièmement, l'utilisation de forces virtuelles pour le contrôle du talon s'avère être une alternative viable à la cinématique inverse quant à la précision du suivi des trajectoires. Les seuls problèmes liés à cette méthode sont les potentiels pics de couples en début de phase d'envol. Comme évoqué précédemment, des améliorations du contrôle lors des changements de phases sont à déterminer et pourraient corriger ce problème.

Quatrièmement, les briques de contrôle actuellement implémentées semblent produire des résultats attendus lors des phases de simple appui et oscillantes. En revanche, elles semblent insuffisantes pour produire des couples aussi biofidèles lors des phases de double appui que ce soit au niveau des couples, des réactions du sol ou du transfert de poids. Une attention particulière doit donc être portée sur le contrôle des phases de double appui.

Enfin, bien qu'elle soit perfectible, il semble que la modélisation des forces de réactions horizontales par des éléments viscoélastiques produise des résultats raisonnables dans le cadre de ce simulateur.

## F. Mesures expérimentales

Bien que le contrôleur ne soit pas encore prêt pour l'implémentation de passages d'obstacles, les résultats de l'expérimentation ont tout de même permis d'obtenir des résultats intéressants pour son développement futur. En effet, l'objectif de l'expérimentation était d'obtenir une base de données pour servir à la construction d'un modèle statistique utilisable dans le simulateur. Cependant, les résultats acquis et présentés dans l'article ont permis de mettre en avant un mécanisme d'adaptation au passage sur tapis roulant dont il sera possible de s'inspirer lors de l'implémentation du passage d'obstacle dans le contrôleur.

De plus, ces résultats ne traitaient que des marges de stabilité mais il est possible que le traitement des données cinématiques et de forces permette également de mettre en avant d'autres mécanismes qui pourront servir à l'amélioration du contrôleur, et ce, également pour le passage de lacune.

## G. Synthèse

Pour conclure, le développement d'un contrôleur utilisant des briques de contrôle et capable de produire un résultat biofidèle tant au niveau de la cinématique que de la dynamique se révèle être compliqué et exploratoire. Ce n'est pas l'approche traditionnellement employée en biomécanique où on voit plutôt se développer des approches par contrôle optimal. Enormément de paramètres sont à prendre en compte, il existe un nombre de contrôles possibles très large et la précision du résultat est fortement limitée par la modélisation de l'interaction du modèle avec son environnement. Ce travail de thèse a permis d'orienter certains choix et de faire ressortir les principaux points saillants. Même s'il n'est pas encore possible d'affirmer que cette méthode de contrôle puisse être utilisable pour les objectifs initiaux, son développement en lui-même s'avère être intéressant et capable d'offrir une nouvelle approche à l'étude de la marche.

# VIII. Annexe

## A. Classe pour la détermination des trajectoires de référence

La première classe à être instanciée se nomme *ReferenceTrajectories* et fournit les trajectoires de référence à chaque pas de temps, et les adapte lors des changements de phases. Elle a comme donnée les paramètres spatiotemporels de la simulation et les coefficients de mise à l'échelle. Elle a aussi une donnée par trajectoire ou partie de trajectoires. Ces données sont un *size\_t* pour contenir le nombre de valeurs dans les *vector* renvoyés par la fonction *statisticalModel* (ex : *hipAddKeyPosesNumber*), deux *vector* pour les positions et les valeurs initiales des poses (ex : *hipAddTime* et *hipAddValue*), un nombre réel pour la dérivée en fin de trajectoire (ex : *hipAddEndSlope*), deux *vectors* pour les positions et valeurs adaptées des poses (ex : *hipAddAdaptedTime* et *hipAddAdaptedValue*), et un nombre réel pour la dérivée en début de trajectoire (ex : *hipAddBegSlope*).

La trajectoire médio latérale du CoM étant calculée différemment des autres trajectoires, trois données concernent cette trajectoire. La première est *comzAmplitude* pour stocker l'amplitude de la trajectoire du CoM. Les deux autres sont une paire de coefficients d'interpolations linéaires utilisées pour adapter la trajectoire du CoM de référence lorsque cela est nécessaire (*aCoMzPosition*, *bCoMzPosition*).

Une donnée nommée *MoSz* est utilisée pour stocker la marge de stabilité qui sera utilisée pour la détermination de la position d'arrivée du pied en phase oscillante. Un tableau 2D de trois lignes et trois colonnes nommé *comReferenceTrajectories* est utilisé pour stocker les poses de référence du CoM à chaque pas de temps. Les trois lignes représentent les trois directions tandis que les trois colonnes représentent respectivement la valeur et la vitesse du CoM et la position du XCoM.

Un tableau 2D de 31 lignes et deux colonnes nommé *dofReferenceTrajectories* est utilisé pour stocker les poses de référence des DoF à chaque pas de temps. Les 31 lignes représentent les DoF tandis que les deux colonnes représentent respectivement la valeur et la vitesse. Un tableau de trois éléments nommé *swingHeelFinalTarget* est utilisé pour stocker la position cible du pied en phase oscillante dans les trois directions. Enfin, un tableau 2D de trois lignes et deux colonnes nommé *heelReferenceTrajectories* est utilisé pour stocker les poses de référence du talon en phase oscillante. Les trois lignes représentent les trois directions tandis que les deux colonnes représentent la valeur et la vitesse.

La classe possède ensuite 30 fonctions membres pour initialiser les données (ex : *setHipAddKeyPoses*), prenant comme paramètres les *vector* et tableaux fournis par *statisticalModel*. Une fonction nommée *keyPosesAdaptation* est définie pour adapter les poses clés, et donc les trajectoires, lors des changements de phase. Une fonction nommée *computeReferenceTrajectories* est définie pour calculer les poses de référence au pas de temps actuel et les stocker dans *comReferenceTrajectories* et *dofReferenceTrajectories*. Une fonction nommée *computeSwingHeelReferenceTrajectories* est définie

pour calculer la position finale cible et la pose de référence du pied en phase oscillante et les stocker respectivement dans *swingHeelFinalTarget* et *heelReferenceTrajectories*. Enfin, trois fonctions nommées respectivement *getComReferenceTrajectories*, *getDofReferenceTrajectories* et *getHeelReferenceTrajectories* sont définies pour renvoyer les tableaux de stockage correspondant.

	Nom	Type	Commentaire
<b>Variables</b>	velocity, strideDuration, etc...	double	Ensemble de 11 valeurs contenant les paramètres spatio-temporels
	scaleCoef	tableau 1D [9]	Coefficients de mise à l'échelle
	*dof*KeyPoseNumber	size_t	Nombre de valeur nécessaire pour chaque trajectoire
	*dof*Time	vector	Temporalité initiale des poses clefs pour chaque trajectoire
	*dof*Value	vector	Valeur initiale des poses clefs pour chaque trajectoire
	*dof*EndSlope	double	Dérivé en fin de trajectoire pour chaque trajectoire
	*dof*AdaptedTime	vector	Temporalité adaptée des poses clefs pour chaque trajectoire
	*dof*AdaptedVakue	vector	Valeur adapté des poses clefs pour chaque trajectoire
	*dof*begSlope	double	Dérivé en début de trajectoire pour chaque trajectoire
	comzAmplitude	double	Amplitude de la trajectoire médio-latéral du CoM
	aCoMz	double	Coefficients pour l'adaptation de la trajectoire du CoM
	bCoMz	double	
	mosz	double	Valeur de la marge de stabilité à appliquer
	comReferenceTrajectories	tableau 2D [3][3]	Valeurs de référence (position, vitesse, XCoM) du CoM dans les trois directions
	dofReferenceTrajectories	tableau 2D [31][2]	Valeurs de référence (position, vitesse) des DoFs
	swingHeelFinalTarget	tableau 1D [3]	Position de pose cible du pied en phase oscillante
	heelReferenceTrajectories	tableau 2D [3][2]	Valeur de référence (position, vitesse) du talon en phase oscillante

Table 11 : Variables de la classe *ReferenceTrajectories*

	Nom	Type	Commentaire
Fonctions	set* <i>dof</i> *keyPoses	void	Prend les valeurs de poses clefs et les stocke dans les variables correspondantes pour chaque trajectoire
	keyPosesAdaptation	void	Adapte les poses clefs
	computeReferenceTrajectories	void	Calcul les valeurs de référence des DoFs et CoM et les stocke dans les variables correspondantes
	computeSwingHeelReferenceTrajectories	void	Calcul les valeurs de référence du talon en phase oscillante
	getComReferenceTrajectories	tableau [3][4]	Renvoie les valeurs de référence du CoM
	getDofReferenceTrajectories	tableau [31][2]	Renvoie les valeurs de référence des DoFs
	getHeelReferenceTrajectories	tableau [3][2]	Renvoie les valeurs de référence du talon en phase oscillante

Table 12 : Fonctions de la classe *ReferenceTrajectories*

## B. Classe du contrôleur proportionnel dérivé

La deuxième classe à être instanciée est nommée *PDController* et détermine les couples articulaires à appliquer pour suivre les trajectoires articulaires de référence grâce à un PDC. Elle possède un tableau de 14 éléments nommé *gainParameters* contenant les gains initiaux sur les positions, et un tableau 2D de 25 lignes et deux colonnes nommé *gain* pour stocker les gains qui seront utilisés par le PDC à chaque pas de temps. Les 25 lignes correspondent aux articulations tandis que les deux colonnes représentent les gains sur la position et sur la vitesse. La classe possède une troisième donnée qui est un tableau de 25 éléments nommé *torque* dans lequel sont stockés les couples articulaires calculés par le PDC pour être renvoyés au contrôleur. Enfin, une donnée nommée *SfcToPdTorqueTransferTime* contient une valeur réelle utilisée pour indiquer la temporalité du passage du contrôle par le SFC au PDC pour le genou en phase oscillante..

Une première fonction de la classe est nommée *updGain* et prend comme paramètre la phase du cycle de marche et le temps en pourcent de la phase oscillante pour mettre à jour *gain* en fonction des valeurs de *gainParameters* et de l'état de la simulation. Une seconde fonction nommée *getTorque* prend comme paramètre les poses de référence des DoF, les poses actuelles des DoF, la phase de marche et le temps en pourcent du cycle du membre inférieur droit, du membre inférieur gauche ainsi que le temps en pourcent de la phase oscillante (i.e. temps évoluant de 0 à 100 entre le TO et le moment du HS subséquent prédit par le modèle statistique) pour calculer les couples du PDC et les renvoyer au contrôleur.

	Nom	Type	Commentaire
<b>Variables</b>	gainParameters	tableau 1D [14]	Gains initiaux en position du PDC
	gain	tableau 2D [25][2]	Gains (position, vitesse) du pas de temps actuel pour chaque DoF
	torque	tableau 1D [25]	Couples articulaires calculées par le PDC
	sfcToPdTorqueTransferTime	double	Valeur temporelle à laquelle le contrôle du genou passe progressivement du SFC au PDC
<b>Fonctions</b>	updGain	void	Met à jour la valeur des gains du PDC
	getTorque	tableau 1D [25]	Calcul et renvoie la valeur des couples du PDC

Table 13 : Membres de la classe *PDController*

## C. Classe du contrôle du talon en phase oscillante

La troisième classe à être instanciée est nommée *SwingFootControl* et contrôle la position du pied en phase oscillante pour suivre sa trajectoire de référence via l'application de forces virtuelles sur le pied.

La classe possède un tableau de neuf éléments nommé *scaleCoef* stockant les coefficients de mise à l'échelle. Elle possède également un tableau 2D de 21 lignes et trois colonnes nommé *gainParameters* stockant les valeurs décrivant l'évolution des gains appliqués au calcul des forces virtuelles. Le gain pour chaque articulation est défini par trois valeurs différentes pour permettre à celui-ci de varier au cours de la phase oscillante. Les articulations utilisées pour contrôler le talon via les forces virtuelles sont la flexion-extension et l'adduction-abduction de la hanche, la flexion-extension du genou, mais également les trois DoF de la hanche controlatérale. Les 21 lignes représentent donc sept trios de gain, tandis que les trois colonnes représentent les directions. La troisième donnée est un tableau 2D de 12 lignes et trois colonnes nommé *gain* stockant les valeurs de gain à utiliser pour le pas de temps actuel. Puisque six articulations sont utilisées pour contrôler un talon, les 12 lignes représentent les gains à utiliser pour les deux talons, tandis que les trois colonnes représentent les directions. Les deux données suivantes sont des nombres réels nommés *a* et *b* et sont utilisés pour stocker les coefficients d'interpolation linéaire utilisés pour faire varier la valeur des gains lors de la phase oscillante. La classe possède aussi deux tableaux 2D de 10 lignes et trois colonnes nommés *rJt* et *lJt*. Ils représentent les matrices transposées des jacobiniennes utilisées pour calculer les forces virtuelles à appliquer au talon. Il y a également trois tableaux de trois éléments nommés *positionChange*, *velocityChange* et *force*. Les deux premiers sont utilisés pour stocker la différence de position et de vitesse du talon entre sa pose réelle et

sa pose de référence qui serviront à déterminer la force à appliquer au talon. Trois valeurs réelles nommées *midGainTime*, *SfcToPdTorqueTransferTime*, *SfcToPdTorqueTransferCoef* sont utilisées pour stocker des valeurs nécessaires lors des transitions de contrôle, à savoir le temps auquel la seconde valeur de gain est atteinte, le temps à partir duquel le contrôle du genou en phase oscillante passe du SFC au PDC, et la valeur du coefficient permettant ce transfert. Enfin, la dernière donnée est un tableau de 25 éléments nommé *torque* dans lequel sont stockés les couples articulaires calculés par le contrôle du talon pour être renvoyés au contrôleur.

La première fonction de cette classe se nomme *setScaleCoef*. Elle prend comme paramètre un tableau de neuf éléments et est utilisée pour transmettre et initialiser les valeurs de mise à l'échelle. La deuxième fonction se nomme *updGain*. Elle prend comme paramètres la phase du cycle de marche et le temps en pourcent de la phase oscillante pour déterminer les gains à appliquer aux forces virtuelles et les stocker dans *gain* à chaque pas de temps, en fonction des données de *gainParameters*. La dernière fonction se nomme *getTorque*. Elle prend comme paramètre la pose de référence du talon, la pose actuelle du talon en phase oscillante, les valeurs de DoF, la phase du cycle de marche, et le temps en pourcent de la phase oscillante, du cycle du membre inférieur droit et du cycle du membre inférieur gauche. Elle calcule alors les couples articulaires à appliquer pour le contrôle du talon et les stocke dans *torque* pour les renvoyer au contrôleur.

	Nom	Type	Commentaire
<b>Variables</b>	scaleCoef	tableau 1D [9]	Coefficients de mise à l'échelle
	gainParameters	tableau 2D[21][3]	Trois gains initiaux par DoF (7*3) décrivant l'évolution des gains qui seront utilisées lors de la simulation (1 par direction)
	gain	tableau 2D [12][3]	Gains du pas de temps actuel pour chaque DoF (1 par direction)
	a	double	Coefficients contrôlant l'évolution des gains pendant la phase oscillante
	b		
	rJt	tableau 2D [10][3]	Matrices transposées des jacobiennes
	lJt		
	force	tableau 1D [3]	Valeur des forces virtuelles à appliquer au talon
	midGainTime	double	Temporalité à laquelle les gains atteignent la valeur du second gain initial
	sfcToPdTorqueTransferTime	double	Valeur temporelle à laquelle le contrôle du genou passe progressivement du SFC au PDC
	sfcToPdTorqueTransferCoef	double	Coefficient utilisé pour diminuer le contrôle du genou par le SFC
torque	tableau 1D [25]	Couples articulaires calculées par le SFC	
<b>Fonctions</b>	setScaleCoef	void	Stocke les valeurs de coefficients de mise à l'échelle dans la variable correspondante
	updGain	void	Calcul la valeur des gains au pas de temps actuel
	getTorque	tableau 1D [25]	Calcul et renvoie la valeur des couples du SFC

Table 14 : Membres de la classe *SwingFootControl*

## D. Classe du contrôle du CoM

La quatrième classe à être instanciée est nommée *ComControl* et contrôle la vitesse du modèle et une partie de son équilibre via l'application de forces virtuelles sur le CoM.

La première donnée de la classe est un tableau de neuf éléments nommé *scaleCoef* stockant les coefficients de mise à l'échelle. La deuxième est un tableau 2D de 15 lignes et trois colonnes nommé *gainParameters* qui permet de stocker les valeurs décrivant l'évolution des gains appliqués aux forces virtuelles. Pour chaque articulation des membres inférieurs exceptées l'articulation métatarso-phalangienne, le gain à utiliser est décrit par deux valeurs pour permettre au gain d'évoluer. Les 15 lignes représentent donc six duos de gains plus les trois gains associés aux articulations des lombaires qui eux ont une valeur fixe, tandis que les trois colonnes représentent les directions. La troisième donnée est un tableau 2D de 17 lignes et trois colonnes nommé *gain*. C'est dans ce tableau que sont stockés les gains à utiliser à chaque pas de temps. Les 17 lignes représentent les 15 articulations utilisées pour les forces virtuelles plus les deux articulations métatarso-phalangiennes inclus dans le tableau pour simplifier sa lecture, tandis que les colonnes représentent les directions. La classe possède également deux nombres réels nommés *a* et *b* utilisés pour stocker les coefficients d'interpolation linéaire utilisés pour faire varier la valeur des gains lors de la phase oscillante. Il y a également un tableau de deux éléments nommé *feetParameters*. Il stocke des gains initiaux à appliquer à un contrôle complémentaire aux forces virtuelles. Ce contrôle complémentaire s'applique aux articulations de la cheville. Un tableau de quatre éléments nommé *feetGain* stocke les gains à appliquer au contrôle complémentaire à chaque pas de temps. Deux tableaux 2D de 10 lignes et trois colonnes nommés *rJt* et *lJt* représentent les matrices jacobiniennes utilisées pour le calcul des forces virtuelles. Les six nombres réels suivants : *timeStepNumber*, *targetVelocityAddition*, *meanTargetVelocity*, *simuVelocityAddition*, *meanSimuVelocity* et *velocityForceParameter* sont utilisés pour déterminer une partie de la force virtuelle à appliquer au CoM. Il y a aussi un tableau de trois éléments nommé *force* dans lequel est stocké la force à appliquer au CoM à chaque pas de temps. L'avant dernière donnée est un tableau de 25 éléments nommé *torque* dans lequel sont stockés les couples articulaires déterminés par la brique de contrôle du CoM pour être renvoyé au contrôleur globale. Enfin, la dernière donnée est un tableau de 25 éléments nommé *feetTorque* dans lequel sont stockées les valeurs de couples du contrôle complémentaire.

La première méthode de cette classe se nomme *setScaleCoef*. Elle prend comme paramètre un array de neuf éléments et est utilisé pour transmettre et initialiser les valeurs de mise à l'échelle.

La deuxième méthode se nomme *updGain*. Elle prend comme argument la phase du cycle de marche, le temps en pourcent de la phase oscillante, et les valeurs des forces de réaction du sol des pieds droit et gauche. C'est elle qui met à jour les gains à utiliser à chaque pas de temps.

La troisième méthode se nomme *getVirtualForceTorque*. Elle prend comme argument les poses de référence du CoM, les poses actuelles du CoM, les valeurs des DoF, la phase du cycle de marche, le temps en pourcent de la phase oscillante, et les valeurs des forces de réaction du sol des pieds droit et gauche. Elle calcul les couples articulaire à appliquer dues aux forces virtuelles, les couples articulaires du contrôle complémentaire, et renvoie l' donnée *torque*.

La dernière méthode se nomme *getFeetTorque* et renvoie l' donnée *feetTorque* calculé précédemment.

	Nom	Type	Commentaire
<b>Variables</b>	scaleCoef	tableau 1D [9]	Coefficients de mise à l'échelle
	gainParameters	tableau 2D[15][3]	Deux gains initiaux par DoF du membre inférieur utilisé (6*2) plus un gain par DoF des lombaires décrivant l'évolution des gains qui seront utilisées lors de la simulation (1 par direction)
	gain	tableau 2D [17] [3]	Gains au pas de temps actuel (1 par direction contrôlée)
	a	double	Coefficients contrôlant l'évolution des gains pendant la phase oscillante
	b		
	HSGrf	tableau 1D [2]	Stocke les forces de réaction verticales aux HS
	rJt	tableau 2D [10] [3]	Matrices transposées des jacobiennes
	lJt		
	timeStepNumber	double	Nombre de pas de temps depuis le premier HS
	targetVelocityAddition	double	Addition de la vitesse antéropostérieur de référence à chaque pas de temps
	meanTargetVelocity	double	Vitesse de référence moyenne du CoM depuis le premier HS
	simuVelocityAddition	double	Addition de la vitesse antéropostérieur de simulation à chaque pas de temps
	simuVelocityAddition	double	Addition de la vitesse antéropostérieur de simulation à chaque pas de temps
	meanSimuVelocity	double	Vitesse de simulation moyenne du CoM depuis le premier HS
	force	tableau 1D [3]	Valeur des forces virtuelles à appliquer au CoM
torque	tableau 1D [25]	Couples articulaires calculées par le CC	
<b>Fonctions</b>	setScaleCoef	void	Stocke les valeurs de coefficients de mise à l'échelle dans la variable correspondante
	setHSGrf	void	Stocke les valeurs de forces de réaction verticales dans la variable correspondante
	updGain	void	Calcul la valeur des gains au pas de temps actuel
	getTorque	tableau 1D [25]	Calcul et renvoie la valeur des couples du CC

Table 15 : Membres de la classe ComControl

## E. Classe de compensation de la gravité

La cinquième classe à être instanciée est nommée *GravityCompensation*. C'est cette classe qui permet de calculer les couples articulaires nécessaires pour compenser l'effet de la gravité sur un segment et donc permettre aux autres briques de contrôle d'utiliser des gains moins élevés pour contrôler ces segments.

Elle a comme premier donnée l'objet *model* défini précédemment pour lui permettre d'accéder à la masse de chaque segment.

Le deuxième donnée est un array de neuf éléments nommé *scaleCoef* stockant les coefficients de mise à l'échelle.

Le troisième donnée est un array de 15 éléments nommé *segmentForce* dans lequel est stocké le poids de chaque segment du modèle, excepté le segment du pelvis utilisé comme lien initial.

Les 15 données suivants sont des array de nombre d'éléments variable dans lequel sont stockées les valeurs des matrices jacobienne de chaque élément (ex : *rFemur*, *rTibia*, etc...). Comparé aux classes précédentes, les jacobienne sont ici stockées dans des array à une dimension. Les forces virtuelles utilisées pour déterminer les couples articulaires n'utilisent que des forces dans la direction verticale. Pour alléger les calculs, seuls les éléments nécessaires au calcul des forces verticales sont écrits dans la classe.

Les deux données suivants sont des *double* nommés *SfcToPdTorqueTransferTime* et *SfcToPdTorqueTransferCoef*. Ils sont utilisés pour stocker des valeurs nécessaires lors des transitions de contrôle.

Le dernier donnée est un array de 25 éléments nommé *torque* dans lequel sont stockées les couples articulaires déterminés par la brique de contrôle pour être renvoyé au contrôleur.

La première méthode se nomme *segmentForceComputation* et est utilisé pour calculer le poids de chaque segment et les stocker dans l'donnée *segmentForce*.

La deuxième méthode se nomme *updJacobian*. Elle prend comme arguments les valeurs des DoF et la phase du cycle de marche pour mettre à jour les valeurs des jacobienne.

La dernière méthode se nomme *getTorque*. Elle prend comme argument les valeurs des DoF, la phase du cycle de marche et le temps en pourcent de la phase oscillante pour calculer et renvoyer les valeurs de couple articulaire de la brique de contrôle.

	Nom	Type	Commentaire
<b>Variables</b>	model	Model	Référence vers le modèle utilisé
	scaleCoef	tableau 1D [9]	Coefficients de mise à l'échelle
	segmentForce	tableau 1D [15]	Poids de chaque segment
	<i>*nom du segment*</i>	tableau 1D [*]	Matrices jacobiennes de chaque segment. Seule la ligne correspondante à la direction verticale est gardée. Le nombre d'élément dépend de la position du segment)
	sfcToPdTorqueTransferTime	double	Valeur temporelle à laquelle le contrôle du genou passe progressivement du SFC au PDC
	sfcToPdTorqueTransferCoef	double	Coefficient utilisé pour diminuer le contrôle du genou par le SFC
	torque	tableau 1D [25]	Couples articulaires calculées par le GC
<b>Fonctions</b>	segmentForceComputation	void	Calcul le poids de chaque segment et le stocke dans la variable correspondante
	updJacobian	void	Met à jour les matrices jacobiennes
	getTorque	tableau 1D [25]	Calcul et renvoie la valeur des couples du GC

Table 16 : Membres de la classe GravityCompensation

## F. Classe de calcul des forces de réaction horizontales

La sixième classe à être instanciée est nommée *GroundReactionForces*. C'est dans cette classe qu'est défini le modèle de contact permettant de calculer les forces de frictions.

Le premier donnée de la classe est un array de neuf éléments nommé *scaleCoef* stockant les coefficients de mise à l'échelle.

Les données suivant sont des int et se nommes *rStancePhase* et *lStancePhase*. Ils sont utilisés pour stocker les phases d'appui des pieds droit et gauche.

Les six données suivants sont des array de quatre éléments nommés *X0*, *Y0*, *Z0*, *Xr0*, *Yr0* et *Zr0*. Ils sont utilisés pour stocker les positions et angles de référence des éléments viscoélastiques. Les quatre éléments représentent les quatre points d'application des forces de réaction.

Les deux données suivant sont des array de quatre éléments nommés *XOAdapted* et *ZOAdapted*. Ils sont utilisés pour stocker les positions de référence des éléments viscoélastiques antéropostérieurs et média latéral en fonction du roulement des pieds sur le sol.

L'donnée suivant est un array de trois éléments nommé *gain*. Il est utilisé pour stocker les raideurs des éléments viscoélastiques.

Les deux derniers données sont des array de huit et quatre éléments nommés *force* et *moment*. Ils sont utilisés pour stocker les valeurs des forces et moment de réaction de réaction.

La première méthode se nomme *updXOZO*. Elle prend comme argument la position des points d'application des forces de réaction dans le repère global, les phases d'appui des pieds droit et gauche, et le temps de la simulation en seconde pour mettre à jour les valeurs de référence des éléments viscoélastiques.

La deuxième méthode se nomme *getForce*. Elle prend comme arguments la position des points d'application des forces de réaction dans le repère global, la phase du cycle de marche, le temps en pourcent du cycle des membres de droite, *rTime*, et des membres de gauche, *lTime*, le temps en pourcent de la phase oscillante et les valeurs de force de réaction du sol vertical des pieds droit et gauche. Elle calcul les forces et moments de réaction du sol et renvoie les forces.

La troisième méthode se nomme *getMoment*. Elle renvoie les moments de réactions précédemment calculées.

	Nom	Type	Commentaire
<b>Variables</b>	scaleCoef	tableau 1D [9]	Coefficients de mise à l'échelle
	rStancePhase	int	Indice de la phase d'appui
	lStancePhase		
	X0	tableau 1D [4]	Position et angles de référence de chaque élément viscoélastique
	Y0		
	Z0		
	Xr0		
	Yr0		
	Zr0		
	X0Adapted		
	Z0Adapted		
	gain	tableau 1D [3]	Raideur des éléments viscoélastiques
	force	tableau 1D [8]	Forces calculées par le modèle de contact
moment	tableau 1D [4]	Moments calculés par le modèle de contact	
<b>Fonctions</b>	updXOZO	void	Calcul les positions de références adaptées
	getForce	tableau 1D [8]	Calcul et renvoie les forces calculées par le modèle de contact
	getMoment	tableau 1D [4]	Renvoie les couples calculés par le modèle de contact

Table 17 : Membres de la classe *GroundReactionForces*

## G. Classe du contrôleur global

La dernière classe à être instanciée est nommée *GlobalController*. Elle est définie comme une classe enfant de la classe *Controller* présente dans la bibliothèque d'OpenSim. La classe *Controller* est notamment responsable de transmettre la valeur des couples à appliquer aux actionneurs. La classe *GlobalController*, développé pour ce projet, étant une classe enfant de la classe *Controller*, elle a également ce rôle mais est également utilisé pour stocker les données sur l'état du système et appeler les fonctions des différentes classes de contrôle à chaque pas de temps.

Les six premiers données concernent le modèle et les actionneurs. Le premier est nommé *model* et est un pointeur vers un objet de classe *Model*. Les quatre données suivants sont des array de taille différentes contenant les pointeurs vers les objets appliquant une force sur model. *torqueTable* est un array de 25 éléments contenant des pointeurs vers des *TorqueActuator*. *HpartieorceTable* est un array de 16 éléments contenant des pointeurs vers des *HuntCrossleyForce*. *frictionForceTable* est un array de huit éléments contenant des pointeurs vers des *PointActuator*. Et *frictionMomentTable* est un array de quatre éléments contenant des pointeurs vers des éléments *TorqueActuator*. Un sixième donnée nommé *scaleCoef* est un array de neuf éléments pour contenir les coefficients de mise à l'échelle.

Les six données suivant concernent les briques de contrôle et sont des pointeurs vers des objets définis par les classes présentées précédemment : *referenceTrajectories*, *pdcontroller*, *swingFootControl*, *comControl*, *gravityCompensation* et *groundReactionForces*.

Les 10 données suivant concernent les forces s'appliquant sur le modèle. Les huit premiers sont des array de 25 éléments contenant des valeurs de couples articulaires. Les cinq premiers se nomment *torqueFromSFC*, *torqueFromPD*, *torqueFromCC*, *torqueFromGC* et *torqueFromFeet*. Ils sont utilisés pour stocker les couples qui proviennent respectivement des briques de contrôle du talon, PDC, contrôle du CoM, compensation de la gravité et du contrôle complémentaire aux forces virtuelles. Le sixième se nomme *totalTorque* et est utilisé pour stocker le couple articulaire final qui est appliqué à chaque pas de temps. Le septième se nomme *phaseTransitionTorque* et est utilisé pour stocker les valeurs de couple total lors de moments spécifiques de la simulation. Les trois données suivant sont des array de 16, huit et cinq éléments nommés *HCForce*, *frictionForce* et *frictionMoment* qui sont utilisés pour stocker les valeurs des forces de réaction du sol.

Les quatre données suivant sont des array 2D de quatre lignes et quatre colonnes nommés *T\_o\_rcalc*, *T\_o\_lcalc*, *T\_o\_rtoes* et *T\_o\_ltoes* et représentent les matrices de passage entre le repère global et les segments des pieds.

Les 13 données suivant concernent l'état du modèle. Le premier est un array 2D de trois lignes et trois colonnes nommé *comValues*. Les trois lignes représentent la position du CoM, la vitesse du CoM, et la position du XCoM, tandis que les trois colonnes représentent les directions. Le deuxième donnée est

un array 2D de 31 lignes et deux colonnes nommé *dofValues* utilisé pour stocker les valeurs des DoF à chaque pas de temps. Les 31 lignes représentent les 31 DoF tandis que les deux colonnes représentent la position et la vitesse. Les troisième et quatrième données sont des array 2D de trois lignes et deux colonnes nommés *rHeelValues* et *lHeelValues*, utilisées pour stocker la position et la vitesse des talons. Les trois lignes représentent les trois directions tandis que les deux colonnes représentent la position et la vitesse. Les cinquième et sixième données sont des array de trois éléments nommés *rHeelPrecedentPosition* et *lHeelPrecedentPosition* et sont utilisés pour stocker les positions des talons au pas de temps précédent dans les trois directions. Les septième et huitième données sont des array 2D de trois lignes et deux colonnes nommés *swingHeelInitialVlues* et *stanceHeelInitialVlues* et sont utilisés pour stocker les valeurs des talons en début de phase oscillante et de simple appui. Le neuvième donnée est un array 2D de trois lignes et deux colonnes nommé *swingheelValues*. Selon le talon en phase aérienne, il a les mêmes valeurs que *rHeelValues* ou *lHeelValues* et est utilisé pour simplifier la lecture et l'écriture du script. Le dixième donnée est un array 3D de quatre lignes, six colonnes et deux niveaux nommé *grfPointValues* utilisé pour stocker les valeurs des points d'application des forces de réaction du sol transverses. Les quatre lignes représentent les quatre points d'application, les six colonnes représentent la direction (forces puis moment), et les deux niveaux représentent la position et la vitesse. Le onzième donnée est un array 2D de quatre lignes et six colonnes nommé *grfPrecedentPosition* utilisé pour stocker les positions des points d'application des forces au pas de temps précédent. Les deux derniers données sont des *double* nommés *rGrfY* et *lGrfY* utilisés pour stoker la valeur des forces de réaction verticales du sol de chacun des pieds.

Les trois données suivant concernent les trajectoires de référence. Le premier donnée est un array 2D de trois lignes et trois colonnes nommé *comReferenceTrajectories* utilisé pour stocker les valeurs de référence du CoM. Le deuxième donnée est un array 2D de 31 lignes et deux colonnes nommé *dofReferenceTrajectories* utilisé pour stocker les poses de référence des DoF. Le troisième donnée est un array 2D de trois lignes et deux colonnes nommé *heelReferenceTrajectories* utilisé pour stocker les poses de référence du talon en phase aérienne.

Les 18 données suivant concernent les données de la simulation. Les onze premiers sont des *double* dans lesquels sont stockés les paramètres spatiotemporels de la simulation (*vitesse*, *HS1*, *CTO*, *CHS*, *TO*, *HS2*, *strideLength*, *stepLength*, *strideDuration*, *swingDuration* et *omega0*). Les quatre données suivant sont des *double* nommés *currentSimulationTime*, *precedentSimulationTime*, *timeStepSecond* et *timeStepPercent* utilisés pour stocker le temps de la simulation en seconde du pas de temps actuel et du pas de temps précédent, et le pas de temps en seconde et en pourcentage du cycle de marche entre les deux. Les deux données suivants sont un *int* et un *string* nommés *gaitPhase* et *phaseTransition* utilisés pour stocker la phase du cycle de marche et si le pas de temps actuel marque le passage d'une phase à la suivante. Le dix-huitième donnée est un *double* nommé *lastHS* utilisé pour garder en mémoire le temps en seconde du dernier HS. Le dix-neuvième donnée est un *double* nommé *strideTimeSecond* utilisé pour stocker le temps en seconde depuis le début du de la foulée la plus récente. Les vingtièmes et vingt-et-unièmes données sont des *double* nommés *rTime* et *lTime* utilisés pour stocker le temps en pourcentage du cycle de marche des foulées du pied droit et gauche. Le vingt-deuxième donnée est un *double* nommé *lastTOPercent* utilisé pour stocker le temps en pourcentage du cycle de marche du lever du pied le plus récent. Les vingt-troisième et vingt-quatrième données sont des *double* nommé

*swingTime* et *swingTimeStepPercent* utilisés pour stocker le temps et le pas de temps en pourcentage de la phase oscillante. Les vingt-cinquième et vingt-sixième données sont des *int* nommés *rStancePhase* et *lStancePhase* utilisés pour stocker la phase d'appui des pieds droit et gauche. Les vingt-septième et vingt-huitième données sont des *double* nommés *HSTorqueAdaptationTime* et *IpToPdTorqueTransferTime* utilisés lors de transferts d'un contrôle à un autre

La première méthode se nomme *computeControls* et est héritée de la classe *Control*. C'est elle qui détermine les couples à appliquer en faisant appel aux différentes briques de contrôle, puis qui transmet ces couples et les forces de réaction au solveur.

La deuxième méthode se nomme *updGainPhase*. C'est via cette fonction que les données de la simulation et du modèle sont mises à jour à chaque pas de temps et que les fonctions adaptant les trajectoires sont appelées.

Les 14 méthodes suivantes sont utilisées pour instancier les données concernant le modèle, les briques de contrôle et les array d'actionneurs (ex : *setModel*, *setTorqueTable*, *setPDController*, etc.).

Une dernière méthode nommée *getContactSpherePosition* est utilisée pour renvoyer les positions de sphères de contact à *Main* pour la détermination de l'état initial de la simulation.

	Nom	Type	Commentaire
<b>Variables liées au modèle</b>	model	Model	Référence vers le modèle utilisé
	torqueTable	tableau 1D [25]	Référence vers les objets actionneurs associés aux DoFs
	HCforceTable	tableau 1D [16]	Référence vers les objets de force de Hunt-Crossley
	frictionForceTable	tableau 1D [8]	Référence vers les actionneurs associés aux modèles de contact
	frictionMomentTable	tableau 1D [4]	
	scaleCoef	tableau 1D [9]	Coefficients de mise à l'échelle

Table 18 : Variables liées au modèle de la classe *GlobalController*

	Nom	Type	Commentaire
<b>Variables liées aux briques de contrôles</b>	referenceTrajectories	ReferenceTrajectories	Référence vers les objets brique de contrôle
	pdcontroller	PDController	
	swingFootControl	SwingFootControl	
	comControl	ComControl	
	gravityCompensation	GravityCompensation	
	groundReactionForces	GroundReactionForces	

Table 19: Variables liées aux briques de contrôle de la classe *GlobalController*

	Nom	Type	Commentaire
<b>Variables liées aux forces et moments</b>	torqueFromSFC	tableau 1D [25]	Valeurs de couple issues des briques de contrôle
	torqueFromPD		
	torqueFromCC		
	torqueFromGC		
	totalTorque		Valeurs de couple total
	phaseTransitionTorque		Valeur de couple au changement de phase précédent
	swingTime100SFCtorque		Valeurs de couple du SFC à 100% du temps de phase oscillante
	HCFforce	tableau 1D [16]	Valeurs des forces et moments calculés par les modèles de contact
	frictionForce	tableau 1D [8]	
	frictionMoment	tableau 1D [4]	

Table 20 : Variables liées aux forces et moments de la classe GlobalController

	Nom	Type	Commentaire
<b>Variables liées aux matrices de passage</b>	T_o_rcalc	tableau 2D [4][4]	Matrices de passages entre le repère global et les segments des pieds
	T_o_lcalc		
	T_o_rtoes		
	T_o_ltoes		

Table 21 : Variables liées aux matrices de passage de la classe GlobalController

	Nom	Type	Commentaire
<b>Variables liées à l'état du modèle</b>	comValues	tableau 2D [3][3]	Valeurs du CoM (position, vitesse, XCoM) dans les trois directions
	dofValues	tableau 2D [31][2]	Valeurs des DoFs (position, vitesse)
	rHeelValues	tableau 2D [3][2]	Valeurs des talons (position, vitesse)
	lHeelValues		
	rHeelPrecedentPosition	tableau 1D [3]	Position des talons au pas de temps précédent
	lHeelPrecedentPosition		
	swingHeelInitialValues	tableau 2D [3][2]	Valeurs des talons (position, vitesse) en début de phase de simple appui / phase oscillante
	stanceHeelInitialValues		
	swingheelValues		
	grfPointValues	tableau 3D [4][6][2]	Valeurs des points d'applications des forces de réactions (position, vitesse) selon les 6 degrés de libertés
	grfPrecedentPosition	tableau 2D [4][6]	Position des points d'applications des forces de réactions au pas de temps précédent
	rGrfY	double	Résultante des forces de réaction

	IGrfY	double	verticales
--	-------	--------	------------

Table 22 : Variables liées à l'état du modèle de la classe GlobalController

	Nom	Type	Commentaire
<b>Variables liées aux trajectoires de référence</b>	comReferenceTrajectories	tableau 2D [3][3]	Valeurs de référence (position, vitesse, XCoM) du CoM dans les trois directions
	dofReferenceTrajectories	tableau 2D [31][2]	Valeurs de référence (position, vitesse) des DoFs
	swingHeelFinalTarget	tableau 1D [3]	Position de pose cible du pied en phase oscillante

Table 23 : Variables liées aux trajectoires de références de la classe GlobalController

	Nom	Type	Commentaire	
<b>Variables liées aux données de la simulation</b>	velocity, strideDuration, etc...	double	Ensemble de 11 valeurs contenant les paramètres spatio-temporels	
	currentSimulationTime		Temps de simulation en seconde	
	precedentSimulationTime		Temps de simulation en seconde du pas de temps précédent	
	timeStepSecond		Pas de temps en seconde	
	timeStepPercent		Pas de temps en pourcent	
	gaitPhase		Indice de la phase de marche	
	phaseTransition	string	Indice de transition de phase	
	lastHS	double	Temps du dernier HS en seconde	
	strideTimeSecond		Temps écoulé depuis le début de foulée la plus récente en seconde	
	rTime		Temps du cycle en pourcent	
	lTime		Temps du dernier TO en pourcent	
	lastTOPercent		Temps de la phase oscillante en pourcent	
	swingTime		Pas de temps de la phase oscillante en pourcent	
	swingTimeStepPercent	int	Indice de la phase d'appui	
	rStancePhase		double	Temps d'adaptions des couples après un HS
	lStancePhase			Valeur temporelle à laquelle le contrôle du genou passe progressivement du SFC au PDC
	HSTorqueAdaptationTime			
SfcToPdTorqueTransferTime				

Table 24 : Variables liées aux données de la simulation de la classe GlobalController

	<b>Nom</b>	<b>Type</b>	<b>Commentaire</b>
<b>Fonctions</b>	<i>computeControls</i>	void	Calcul les couples et les transmet au solveur
	<i>updGainPhase</i>	void	Met à jour les variables à chaque pas de temps et appelle les fonctions nécessaires aux adaptations si besoin
	<i>set*élément*</i>	void	Transmet les références et valeurs nécessaires tel que les briques de contrôle
	<i>getContactSpherePosition</i>	Tableau 1D [6]	Calcul et renvoie la position de certaines sphères de contact

Table 25 : Fonctions de la classe *GlobalController*

# IX. Références

- Abe, Y. (2007). *Multiobjective Control with Frictional Contacts*. 10.
- Ackermann, M., & van den Bogert, A. J. (2010). Optimality principles for model-based prediction of human gait. *Journal of Biomechanics*, 43(6), Art. 6. <https://doi.org/10.1016/j.jbiomech.2009.12.012>
- Alamoudi, R., & Alamoudi, M. (2020). Development of Linear Regression Models to Estimate the Margin of Stability Based on Spatio-Temporal Gait Parameters. *IEEE Access*, 8, 19853-19859. <https://doi.org/10.1109/ACCESS.2020.2969294>
- Albert, A., & Gerth, W. (2003). Analytic Path Planning Algorithms for Bipedal Robots without a Trunk. *Journal of Intelligent and Robotic Systems*, 36(2), 109-127. <https://doi.org/10.1023/A:1022600522613>
- Allen, B. F., & Faloutsos, P. (2009). Evolved Controllers for Simulated Locomotion. In A. Egges, R. Geraerts, & M. Overmars (Éds.), *Motion in Games* (p. 219-230). Springer. [https://doi.org/10.1007/978-3-642-10347-6\\_20](https://doi.org/10.1007/978-3-642-10347-6_20)
- Al-Shuka, H. F. N., Corves, B., Zhu, W.-H., & Vanderborght, B. (2016). Multi-level control of zero-moment point-based humanoid biped robots : A review. *Robotica*, 34(11), 2440-2466. <https://doi.org/10.1017/S0263574715000107>
- Anderson, D., Madigan, M., & Nussbaum, M. (2007). Maximum voluntary joint torque as a function of joint angle and angular velocity : Model development and application to the lower limb. *Journal of biomechanics*, 40, 3105-3113. <https://doi.org/10.1016/j.jbiomech.2007.03.022>
- Anderson, F. C., & Pandy, M. G. (2001). Dynamic Optimization of Human Walking. *Journal of Biomechanical Engineering*, 123(5), Art. 5. <https://doi.org/10.1115/1.1392310>

- Anderson, & Pandy. (1999). A Dynamic Optimization Solution for Vertical Jumping in Three Dimensions. *Computer Methods in Biomechanics and Biomedical Engineering*, 2(3), Art. 3. <https://doi.org/10.1080/10255849908907988>
- Aoyagi, Y., & Shephard, R. J. (2010). Habitual physical activity and health in the elderly : The Nakanojo Study. *Geriatrics & Gerontology International*, 10(s1), S236-S243. <https://doi.org/10.1111/j.1447-0594.2010.00589.x>
- Arslan, Y. Z., Karabulut, D., Ortes, F., & Popovic, M. B. (2019). 11 - Exoskeletons, Exomusculatures, Exosuits : Dynamic Modeling and Simulation. In M. B. Popovic (Éd.), *Biomechatronics* (p. 305-331). Academic Press. <https://doi.org/10.1016/B978-0-12-812939-5.00011-2>
- Azevedo, C., Poignet, P., & Espiau, B. (2004). Artificial locomotion control : From human to robots. *Robotics and Autonomous Systems*, 47(4), 203-223. <https://doi.org/10.1016/j.robot.2004.03.013>
- Bauman, A. E. (2004). Updating the evidence that physical activity is good for health : An epidemiological review 2000–2003. *Journal of Science and Medicine in Sport*, 7(1, Supplement 1), 6-19. [https://doi.org/10.1016/S1440-2440\(04\)80273-1](https://doi.org/10.1016/S1440-2440(04)80273-1)
- Beausoleil, S., Miramand, L., & Turcot, K. (2019). Evolution of gait parameters in individuals with a lower-limb amputation during a six-minute walk test. *Gait & Posture*, 72, 40-45. <https://doi.org/10.1016/j.gaitpost.2019.05.022>
- Bessonnet, G., Sardain, P., & Chessé, S. (2002). *Optimal Motion Synthesis – Dynamic Modelling and Numerical Solving Aspects*. 22.
- Bessonnet, G., Seguin, P., & Sardain, P. (2005). A Parametric Optimization Approach to Walking Pattern Synthesis. *The International Journal of Robotics Research*, 24(7), 523-536. <https://doi.org/10.1177/0278364905055377>
- Bishop, C. M. (1994). Neural networks and their applications. *Review of Scientific Instruments*, 65(6), 1803-1832. <https://doi.org/10.1063/1.1144830>

- Blake, A. J., Morgan, K., Bendall, M. J., Dallosso, H., Ebrahim, S. B. J., ARIE, T. H. D., FENTEM, P. H., & BASSEY, E. J. (1988). FALLS BY ELDERLY PEOPLE AT HOME : PREVALENCE AND ASSOCIATED FACTORS. *Age and Ageing*, 17(6), 365-372. <https://doi.org/10.1093/ageing/17.6.365>
- Blin, O., Ferrandez, A. M., & Serratrice, G. (1990). Quantitative analysis of gait in Parkinson patients : Increased variability of stride length. *Journal of the Neurological Sciences*, 98(1), 91-97. [https://doi.org/10.1016/0022-510X\(90\)90184-O](https://doi.org/10.1016/0022-510X(90)90184-O)
- Boos, M., & McPhee, J. (2013). Volumetric Modeling and Experimental Validation of Normal Contact Dynamic Forces. *Journal of Computational and Nonlinear Dynamics*, 8(2), Art. 2. <https://doi.org/10.1115/1.4006836>
- Borzova, E., & Hurmuzlu, Y. (2004). Passively walking five-link robot. *Automatica*, 40(4), 621-629. <https://doi.org/10.1016/j.automatica.2003.10.015>
- Brown, P., & McPhee, J. (2018). A 3D ellipsoidal volumetric foot-ground contact model for forward dynamics. *Multibody System Dynamics*, 42(4), Art. 4. <https://doi.org/10.1007/s11044-017-9605-4>
- Busse, A. L., Gil, G., Santarém, J. M., & Jacob Filho, W. (2009). Physical activity and cognition in the elderly : A review. *Dementia & Neuropsychologia*, 3(3), 204-208. <https://doi.org/10.1590/S1980-57642009DN30300005>
- Carensac, S., Pronost, N., & Bouakaz, S. (2018). Physics-based control of walking virtual characters in low frequency simulations. *Proceedings of the 31st International Conference on Computer Animation and Social Agents - CASA 2018*, 77-82. <https://doi.org/10.1145/3205326.3205361>
- Cenk Güler, H., Berme, N., & Simon, S. R. (1998). A viscoelastic sphere model for the representation of plantar soft tissue during simulations. *Journal of Biomechanics*, 31(9), Art. 9. [https://doi.org/10.1016/S0021-9290\(98\)00085-2](https://doi.org/10.1016/S0021-9290(98)00085-2)
- Channon, P. H., Hopkins, S. H., & Pham, D. T. (1992). Derivation of optimal walking motions for a bipedal walking robot. *Robotica*, 10(2), 165-172. <https://doi.org/10.1017/S026357470000758X>

- Chevallereau, C., & Aoustin, Y. (2001). Optimal reference trajectories for walking and running of a biped robot. *Robotica*, 19(5), 557-569. <https://doi.org/10.1017/S0263574701003307>
- Collins, S. H., Adamczyk, P. G., & Kuo, A. D. (2009). Dynamic arm swinging in human walking. *Proceedings of the Royal Society B: Biological Sciences*, 276(1673), 3679-3688. <https://doi.org/10.1098/rspb.2009.0664>
- Collins, S. H., Wisse, M., & Ruina, A. (2001). A Three-Dimensional Passive-Dynamic Walking Robot with Two Legs and Knees. *The International Journal of Robotics Research*, 20(7), 607-615. <https://doi.org/10.1177/02783640122067561>
- Coros, S., Beaudoin, P., & van de Panne, M. (2010). Generalized Biped Walking Control. *ACM SIGGRAPH 2010 Papers*, 130:1-130:9. <https://doi.org/10.1145/1833349.1781156>
- Cruz Ruiz, A. L., Pontonnier, C., Pronost, N., & Dumont, G. (2017). Muscle-Based Control for Character Animation : Muscle-Based Control for Character Animation. *Computer Graphics Forum*, 36(6), Art. 6. <https://doi.org/10.1111/cgf.12863>
- Da Silva, M., Abe, Y., & Popović, J. (2008). Simulation of Human Motion Data using Short-Horizon Model-Predictive Control. *Computer Graphics Forum*, 27(2), 371-380. <https://doi.org/10.1111/j.1467-8659.2008.01134.x>
- da Silva, M., Abe, Y., & Popović, J. (2008). Interactive Simulation of Stylized Human Locomotion. *ACM SIGGRAPH 2008 Papers*, 82:1-82:10. <https://doi.org/10.1145/1399504.1360681>
- Davies, B., & Datta, D. (2003). Mobility outcome following unilateral lower limb amputation. *Prosthetics and Orthotics International*, 27(3), 186-190. <https://doi.org/10.1080/03093640308726681>
- de Garis, H. (1990). GENETIC PROGRAMMING : Building Artificial Nervous Systems Using Genetically Programmed Neural Network Modules. In B. Porter & R. Mooney (Éds.), *Machine Learning Proceedings 1990* (p. 132-139). Morgan Kaufmann. <https://doi.org/10.1016/B978-1-55860-141-3.50019-5>

- de Lasa, M., Mordatch, I., & Hertzmann, A. (2010). Feature-based locomotion controllers. *ACM Transactions on Graphics*, 29(4), 1-10. <https://doi.org/10.1145/1778765.1781157>
- Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., & Thelen, D. G. (2007). OpenSim : Open-Source Software to Create and Analyze Dynamic Simulations of Movement. *IEEE Transactions on Biomedical Engineering*, 54(11), Art. 11. <https://doi.org/10.1109/TBME.2007.901024>
- Donelan, J. M., Kram, R., & Kuo, A. D. (2002). Simultaneous positive and negative external mechanical work in human walking. *Journal of Biomechanics*, 35(1), Art. 1. [https://doi.org/10.1016/S0021-9290\(01\)00169-5](https://doi.org/10.1016/S0021-9290(01)00169-5)
- Dorn, T. W., Wang, J. M., Hicks, J. L., & Delp, S. L. (2015). Predictive Simulation Generates Human Adaptations during Loaded and Inclined Walking. *PLOS ONE*, 10(4), Art. 4. <https://doi.org/10.1371/journal.pone.0121407>
- Falisse, A., Serrancoí, G., Dembia, C. L., Gillis, J., Jonkers, I., & De Groote, F. (2019). Rapid predictive simulations with complex musculoskeletal models suggest that diverse healthy and pathological human gaits can emerge from similar control strategies. *Journal of The Royal Society Interface*, 16(157), Art. 157. <https://doi.org/10.1098/rsif.2019.0402>
- Faloutsos, P., van de Panne, M., & Terzopoulos, D. (2001a). Composable controllers for physics-based character animation. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '01*, 251-260. <https://doi.org/10.1145/383259.383287>
- Faloutsos, P., van de Panne, M., & Terzopoulos, D. (2001b). The virtual stuntman : Dynamic characters with a repertoire of autonomous motor skills. *Computers & Graphics*, 25(6), 933-953. [https://doi.org/10.1016/S0097-8493\(01\)00171-6](https://doi.org/10.1016/S0097-8493(01)00171-6)

- Finlayson, M., & van Denend, T. (2003). Experiencing the loss of mobility : Perspectives of older adults with MS. *Disability and Rehabilitation*, 25(20), 1168-1180. <https://doi.org/10.1080/09638280310001596180>
- Flores, P., Machado, M., Silva, M. T., & Martins, J. M. (2011). On the continuous contact force models for soft materials in multibody dynamics. *Multibody System Dynamics*, 25(3), Art. 3. <https://doi.org/10.1007/s11044-010-9237-4>
- García, C. E., Prett, D. M., & Morari, M. (1989). Model predictive control : Theory and practice—A survey. *Automatica*, 25(3), 335-348. [https://doi.org/10.1016/0005-1098\(89\)90002-2](https://doi.org/10.1016/0005-1098(89)90002-2)
- Geijtenbeek, T., & Pronost, N. (2012). Interactive Character Animation Using Simulated Physics : A State-of-the-Art Review. *Computer Graphics Forum*, 31(8), Art. 8. <https://doi.org/10.1111/j.1467-8659.2012.03189.x>
- Geijtenbeek, T., van de Panne, M., & van der Stappen, A. F. (2013). Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics*, 32(6), Art. 6. <https://doi.org/10.1145/2508363.2508399>
- Gerritsen, K. G. M., van den Bogert, A. J., & Nigg, B. M. (1995). Direct dynamics simulation of the impact phase in heel-toe running. *Journal of Biomechanics*, 28(6), Art. 6. [https://doi.org/10.1016/0021-9290\(94\)00127-P](https://doi.org/10.1016/0021-9290(94)00127-P)
- Glitsch, U., & Baumann, W. (1997). The three-dimensional determination of internal loads in the lower extremity. *Journal of Biomechanics*, 30(11), Art. 11. [https://doi.org/10.1016/S0021-9290\(97\)00089-4](https://doi.org/10.1016/S0021-9290(97)00089-4)
- Goswami, A., Thuilot, B., & Espiau, B. (1998). A Study of the Passive Gait of a Compass-Like Biped Robot : Symmetry and Chaos. *The International Journal of Robotics Research*, 17(12), 1282-1301. <https://doi.org/10.1177/027836499801701202>

- Gubina, F., Hemami, H., & McGhee, R. B. (1974). On the Dynamic Stability of Biped Locomotion. *IEEE Transactions on Biomedical Engineering, BME-21(2)*, 102-108.  
<https://doi.org/10.1109/TBME.1974.324294>
- Guess, T. M., Stylianou, A. P., & Kia, M. (2014). Concurrent Prediction of Muscle and Tibiofemoral Contact Forces During Treadmill Gait. *Journal of Biomechanical Engineering, 136(2)*, Art. 2.  
<https://doi.org/10.1115/1.4026359>
- Hak, L., Hettinga, F. J., Duffy, K. R., Jackson, J., Sandercock, G. R. H., & Taylor, M. J. D. (2019). The concept of margins of stability can be used to better understand a change in obstacle crossing strategy with an increase in age. *Journal of Biomechanics, 84*, 147-152.  
<https://doi.org/10.1016/j.jbiomech.2018.12.037>
- Hak, L., Houdijk, H., Steenbrink, F., Mert, A., van der Wurff, P., Beek, P. J., & van Dieën, J. H. (2012). Speeding up or slowing down? : Gait adaptations to preserve gait stability in response to balance perturbations. *Gait & Posture, 36(2)*, 260-264. <https://doi.org/10.1016/j.gaitpost.2012.03.005>
- Hak, L., Houdijk, H., Steenbrink, F., Mert, A., van der Wurff, P., Beek, P. J., & van Dieën, J. H. (2013). Stepping strategies for regulating gait adaptability and stability. *Journal of Biomechanics, 46(5)*, 905-911. <https://doi.org/10.1016/j.jbiomech.2012.12.017>
- Hamner, S. R., Seth, A., & Delp, S. L. (2010). Muscle contributions to propulsion and support during running. *Journal of Biomechanics, 43(14)*, Art. 14.  
<https://doi.org/10.1016/j.jbiomech.2010.06.025>
- Hamner, S. R., Seth, A., Steele, K. M., & Delp, S. L. (2013). A rolling constraint reproduces ground reaction forces and moments in dynamic simulations of walking, running, and crouch gait. *Journal of Biomechanics, 46(10)*, Art. 10. <https://doi.org/10.1016/j.jbiomech.2013.03.030>
- Hansen, N., Auger, A., Ros, R., Finck, S., & Pošík, P. (2010). Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. *Proceedings of the 12th Annual Conference*

- Comp on Genetic and Evolutionary Computation - GECCO '10*, 1689.  
<https://doi.org/10.1145/1830761.1830790>
- Hausdorff, J. M. (2005). Gait variability : Methods, modeling and meaning. *Journal of NeuroEngineering and Rehabilitation*, 2(1), 19. <https://doi.org/10.1186/1743-0003-2-19>
- Hausdorff, J. M., Cudkowicz, M. E., Firtion, R., Wei, J. Y., & Goldberger, A. L. (1998). Gait variability and basal ganglia disorders : Stride-to-stride variations of gait cycle timing in parkinson's disease and Huntington's disease. *Movement Disorders*, 13(3), 428-437.  
<https://doi.org/10.1002/mds.870130310>
- Hausdorff, J. M., Yogev, G., Springer, S., Simon, E. S., & Giladi, N. (2005). Walking is more like catching than tapping : Gait in the elderly as a complex cognitive task. *Experimental Brain Research*, 164(4), 541-548. <https://doi.org/10.1007/s00221-005-2280-3>
- Hawkins, N. M., & Atha, J. (1976). A Study of Passenger Behaviour on a Slow Speed Travellator System. *Ergonomics*, 19(4), 499-517. <https://doi.org/10.1080/00140137608931561>
- Herbert, E. (1939). *The Function of the Arms in Walking—ProQuest*.  
<https://www.proquest.com/openview/a3b1781e995800c0207b7dbb5f7522e3/1?pq-origsite=gscholar&cbl=1818403>
- Highsmith, M. J., Schulz, B. W., Hart-Hughes, S., Latlief, G. A., & Phillips, S. L. (2010). Differences in the Spatiotemporal Parameters of Transtibial and Transfemoral Amputee Gait. *JPO: Journal of Prosthetics and Orthotics*, 22(1), 26-30. <https://doi.org/10.1097/JPO.0b013e3181cc0e34>
- Hill, A. V. (1938). The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London. Series B - Biological Sciences*, 126(843), 136-195.  
<https://doi.org/10.1098/rspb.1938.0050>
- Hirai, K. (1999). The Honda humanoid robot : Development and future perspective. *Industrial Robot: An International Journal*, 26(4), 260-266. <https://doi.org/10.1108/01439919910277431>

- Hoang, I., Paire-Ficout, L., Derollepot, R., Perrey, S., Devos, H., & Ranchet, M. (2022). Increased prefrontal activity during usual walking in aging. *International Journal of Psychophysiology*, *174*, 9-16. <https://doi.org/10.1016/j.ijpsycho.2022.01.011>
- Hodgins, J. K., & Pollard, N. S. (1997). Adapting simulated behaviors for new characters. *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '97*, 153-162. <https://doi.org/10.1145/258734.258822>
- Hodgins, J. K., Wooten, W. L., Brogan, D. C., & O'Brien, J. F. (1995). Animating human athletics. *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '95*, 71-78. <https://doi.org/10.1145/218380.218414>
- Hof, A. L. (2008). The 'extrapolated center of mass' concept suggests a simple control of balance in walking. *Human Movement Science*, *27*(1), Art. 1. <https://doi.org/10.1016/j.humov.2007.08.003>
- Hof, A. L., Gazendam, M. G. J., & Sinke, W. E. (2005). The condition for dynamic stability. *Journal of Biomechanics*, *38*(1), Art. 1. <https://doi.org/10.1016/j.jbiomech.2004.03.025>
- Hof, A. L., van Bockel, R. M., Schoppen, T., & Postema, K. (2007). Control of lateral balance in walking. *Gait & Posture*, *25*(2), Art. 2. <https://doi.org/10.1016/j.gaitpost.2006.04.013>
- Hunt, K. H., & Crossley, F. R. E. (1975). Coefficient of Restitution Interpreted as Damping in Vibroimpact. *Journal of Applied Mechanics*, *42*(2), Art. 2. <https://doi.org/10.1115/1.3423596>
- Hunter, S. W., Batchelor, F., Hill, K. D., Hill, A.-M., Mackintosh, S., & Payne, M. (2017). Risk Factors for Falls in People With a Lower Limb Amputation : A Systematic Review. *PM&R*, *9*(2), 170-180.e1. <https://doi.org/10.1016/j.pmrj.2016.07.531>
- Iwasaki, T., Okamoto, S., Akiyama, Y., & Yamada, Y. (2022). Gait Stability Index Built by Kinematic Information Consistent With the Margin of Stability Along the Mediolateral Direction. *IEEE Access*, *10*, 52832-52839. <https://doi.org/10.1109/ACCESS.2022.3175409>

- Izquierdo, M., Aguado, X., Gonzalez, R., López, J. L., & Häkkinen, K. (1999). Maximal and explosive force production capacity and balance performance in men of different ages. *European Journal of Applied Physiology and Occupational Physiology*, 79(3), 260-267. <https://doi.org/10.1007/s004210050504>
- Jackson, J. N., Hass, C. J., & Fregly, B. J. (2016). Development of a Subject-Specific Foot-Ground Contact Model for Walking. *Journal of Biomechanical Engineering*, 138(9), Art. 9. <https://doi.org/10.1115/1.4034060>
- Jain, S., Ye, Y., & Liu, C. K. (2009). Optimization-based interactive motion synthesis. *ACM Transactions on Graphics*, 28(1), 1-12. <https://doi.org/10.1145/1477926.1477936>
- Jie Tan, Liu, K., & Turk, G. (2011). Stable Proportional-Derivative Controllers. *IEEE Computer Graphics and Applications*, 31(4), Art. 4. <https://doi.org/10.1109/MCG.2011.30>
- Kajita, S., Matsumoto, O., & Saigo, M. (2001). Real-time 3D walking pattern generation for a biped robot with telescopic legs. *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 3, 2299-2306 vol.3. <https://doi.org/10.1109/ROBOT.2001.932965>
- Kajita, S., & Tani, K. (1991). *Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode*. 1405,1406,1407,1408,1409,1410,1411-1405,1406,1407,1408,1409,1410,1411. <https://doi.org/10.1109/ROBOT.1991.131811>
- Kajita, S., Yamaura, T., & Kobayashi, A. (1992). Dynamic walking control of a biped robot along a potential energy conserving orbit. *IEEE Transactions on Robotics and Automation*, 8(4), 431-438. <https://doi.org/10.1109/70.149940>
- Kato, R., & Mori, M. (1984). Control method of biped locomotion giving asymptotic stability of trajectory. *Automatica*, 20(4), 405-414. [https://doi.org/10.1016/0005-1098\(84\)90099-2](https://doi.org/10.1016/0005-1098(84)90099-2)

- Khulief, Y. A., & Shabana, A. A. (1987). A continuous force model for the impact analysis of flexible multibody systems. *Mechanism and Machine Theory*, 22(3), 213-224.  
[https://doi.org/10.1016/0094-114X\(87\)90004-8](https://doi.org/10.1016/0094-114X(87)90004-8)
- Kim, H. J., Wang, Q., Rahmatalla, S., Swan, C. C., Arora, J. S., Abdel-Malek, K., & Assouline, J. G. (2008). Dynamic Motion Planning of 3D Human Locomotion Using Gradient-Based Optimization. *Journal of Biomechanical Engineering*, 130(3). <https://doi.org/10.1115/1.2898730>
- König, M., Epro, G., Seeley, J., Catalá-Lehnen, P., Potthast, W., & Karamanidis, K. (2019). Retention of improvement in gait stability over 14 weeks due to trip-perturbation training is dependent on perturbation dose. *Journal of Biomechanics*, 84, 243-246.  
<https://doi.org/10.1016/j.jbiomech.2018.12.011>
- Koopman, B., Grootenboer, H. J., & de Jongh, H. J. (1995). An inverse dynamics model for the analysis, reconstruction and prediction of bipedal walking. *Journal of Biomechanics*, 28(11), 1369-1376.  
[https://doi.org/10.1016/0021-9290\(94\)00185-7](https://doi.org/10.1016/0021-9290(94)00185-7)
- Kuo, A. D. (2001). A Simple Model of Bipedal Walking Predicts the Preferred Speed–Step Length Relationship. *Journal of Biomechanical Engineering*, 123(3), 264-269.  
<https://doi.org/10.1115/1.1372322>
- Kuo, A. D., Donelan, J. M., & Ruina, A. (2005). Energetic Consequences of Walking Like an Inverted Pendulum : Step-to-Step Transitions. *Exercise and Sport Sciences Reviews*, 33(2), 88-97.
- Landi, F., Russo, A., & Bernabei, R. (2004). PHYSICAL ACTIVITY AND BEHAVIOR IN THE ELDERLY : A PILOT STUDY. *Archives of Gerontology and Geriatrics*, 38, 235-241.  
<https://doi.org/10.1016/j.archger.2004.04.033>
- Lankarani, H. M., & Nikravesh, P. E. (2021). A Contact Force Model With Hysteresis Damping for Impact Analysis of Multibody Systems. 45-51. <https://doi.org/10.1115/DETC1989-0104>

- Laszlo, J., van de Panne, M., & Fiume, E. (1996). Limit cycle control and its application to the animation of balancing and walking. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '96*, 155-162. <https://doi.org/10.1145/237170.237231>
- Leardini, A., Sawacha, Z., Paolini, G., Ingrosso, S., Natio, R., & Benedetti, M. G. (2007). A new anatomically based protocol for gait analysis in children. *Gait & Posture*, 26(4), 560-571. <https://doi.org/10.1016/j.gaitpost.2006.12.018>
- Lee, Y., Park, M. S., Kwon, T., & Lee, J. (2014). Locomotion control for many-muscle humanoids. *ACM Transactions on Graphics*, 33(6), Art. 6. <https://doi.org/10.1145/2661229.2661233>
- Lin, Y.-C., Kim, H. J., & Pandy, M. G. (2011). A computationally efficient method for assessing muscle function during human locomotion. *International Journal for Numerical Methods in Biomedical Engineering*, 27(3), Art. 3. <https://doi.org/10.1002/cnm.1396>
- Macchietto, A., Zordan, V., & Shelton, C. R. (2009). Momentum control for balance. *ACM Transactions on Graphics*, 28(3), 1-8. <https://doi.org/10.1145/1531326.1531386>
- Mahboobin, A., Cham, R., & Piazza, S. J. (2010). The impact of a systematic reduction in shoe-floor friction on heel contact walking kinematics—A gait simulation approach. *Journal of Biomechanics*, 43(8), Art. 8. <https://doi.org/10.1016/j.jbiomech.2010.01.040>
- Matarić, M. J., Zordan, V. B., & Williamson, M. M. (1999). Making Complex Articulated Agents Dance. *Autonomous Agents and Multi-Agent Systems*, 2(1), Art. 1. <https://doi.org/10.1023/A:1010023022632>
- McGeer, T. (1990). Passive walking with knees. , *IEEE International Conference on Robotics and Automation Proceedings*, 1640-1645 vol.3. <https://doi.org/10.1109/ROBOT.1990.126245>
- McLean, S. G., Su, A., & van den Bogert, A. J. (2003). Development and Validation of a 3-D Model to Predict Knee Joint Loading During Dynamic Movement. *Journal of Biomechanical Engineering*, 125(6), Art. 6. <https://doi.org/10.1115/1.1634282>

- Miller, R. H. (2014). A comparison of muscle energy models for simulating human walking in three dimensions. *Journal of Biomechanics*, 47(6), Art. 6. <https://doi.org/10.1016/j.jbiomech.2014.01.049>
- Mitchell, W., Atherton, P., Williams, J., Larvin, M., Lund, J., & Narici, M. (2012). Sarcopenia, Dynapenia, and the Impact of Advancing Age on Human Skeletal Muscle Size and Strength; a Quantitative Review. *Frontiers in Physiology*, 3. <https://www.frontiersin.org/articles/10.3389/fphys.2012.00260>
- Moissenet, F., Leboeuf, F., & Armand, S. (2019). Lower limb sagittal gait kinematics can be predicted based on walking speed, gender, age and BMI. *Scientific Reports*, 9(1), Art. 1. <https://doi.org/10.1038/s41598-019-45397-4>
- Moreira, P., Silva, M., & Flores, P. (2009). *GROUND FOOT INTERACTION IN HUMAN GAIT: MODELLING AND SIMULATION*. 14.
- Muico, U., Lee, Y., Popović, J., & Popović, Z. (2009). Contact-aware nonlinear control of dynamic characters. *ACM Transactions on Graphics*, 28(3), 1-9. <https://doi.org/10.1145/1531326.1531387>
- Muico, U., Popović, J., & Popović, Z. (2011). Composite control of physically simulated characters. *ACM Transactions on Graphics*, 30(3), 1-11. <https://doi.org/10.1145/1966394.1966395>
- Mussa-Ivaldi, F. A. (1997). Nonlinear force fields : A distributed system of control primitives for representing and learning movements. *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. « Towards New Computational Principles for Robotics and Automation »*, 84-90. <https://doi.org/10.1109/CIRA.1997.613842>
- Neff, M., & Fiume, E. (2002). *Modeling Tension and Relaxation for Computer Animation*. 8.
- Neptune, R. R., WRIGHT, I. C., & BOGERT, A. J. V. D. (2000). A Method for Numerical Simulation of Single Limb Ground Contact Events : Application to Heel-Toe Running. *Computer Methods in*

*Biomechanics and Biomedical Engineering*, 3(4), Art. 4.

<https://doi.org/10.1080/10255840008915275>

Pàmies-Vilà, R., Font-Llagunes, J. M., Lugrís, U., & Cuadrado, J. (2012). *Two approaches to estimate foot-ground contact model parameters using optimization techniques. 2.*

Pàmies-Vilà, R., Font-Llagunes, J. M., Lugrís, U., & Cuadrado, J. (2014). Parameter identification method for a three-dimensional foot-ground contact model. *Mechanism and Machine Theory*, 75, 107-116. <https://doi.org/10.1016/j.mechmachtheory.2014.01.010>

Park, J. H., & Kim, K. D. (1998). Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, 4, 3528-3533 vol.4. <https://doi.org/10.1109/ROBOT.1998.680985>

Peasgood, M., Kubica, E., & McPhee, J. (2007). Stabilization of a Dynamic Walking Gait Simulation. *Journal of Computational and Nonlinear Dynamics*, 2(1), Art. 1. <https://doi.org/10.1115/1.2389230>

Pell, J. P., Donnan, P. T., Fowkes, F. G. R., & Ruckley, C. V. (1993). Quality of life following lower limb amputation for peripheral arterial disease. *European Journal of Vascular Surgery*, 7(4), 448-451. [https://doi.org/10.1016/S0950-821X\(05\)80265-8](https://doi.org/10.1016/S0950-821X(05)80265-8)

Porsa, S., Lin, Y.-C., & Pandy, M. G. (2016). Direct Methods for Predicting Movement Biomechanics Based Upon Optimal Control Theory with Implementation in OpenSim. *Annals of Biomedical Engineering*, 44(8), Art. 8. <https://doi.org/10.1007/s10439-015-1538-6>

Pratt, J., Chew, C.-M., Torres, A., Dilworth, P., & Pratt, G. (2001). Virtual Model Control : An Intuitive Approach for Bipedal Locomotion. *The International Journal of Robotics Research*, 20(2), Art. 2. <https://doi.org/10.1177/02783640122067309>

- Press, V., Freestone, I., & George, C. F. (2003). Physical activity : The evidence of benefit in the prevention of coronary heart disease. *QJM: An International Journal of Medicine*, *96*(4), 245-251. <https://doi.org/10.1093/qjmed/hcg041>
- Pycma*. (2022). [Python]. CMA-ES. <https://github.com/CMA-ES/pycma> (Original work published 2016)
- Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, *11*(7), 733-764. [https://doi.org/10.1016/S0967-0661\(02\)00186-7](https://doi.org/10.1016/S0967-0661(02)00186-7)
- Raibert, M. H., & Hodgins, J. K. (1991). *Animation of dynamic legged locomotion*. <https://doi.org/10.1145/122718.122755>
- Reil, T., & Husbands, P. (2002). Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Transactions on Evolutionary Computation*, *6*(2), 159-168. <https://doi.org/10.1109/4235.996015>
- Ren, L., Jones, R. K., & Howard, D. (2007). Predictive modelling of human walking over a complete gait cycle. *Journal of Biomechanics*, *40*(7), 1567-1574. <https://doi.org/10.1016/j.jbiomech.2006.07.017>
- Rezende, L. F. M. de, Rey-López, J. P., Matsudo, V. K. R., & Luiz, O. do C. (2014). Sedentary behavior and health outcomes among older adults : A systematic review. *BMC Public Health*, *14*(1), 333. <https://doi.org/10.1186/1471-2458-14-333>
- Roeles, S., Rowe, P. J., Bruijn, S. M., Childs, C. R., Tarfali, G. D., Steenbrink, F., & Pijnappels, M. (2018). Gait stability in response to platform, belt, and sensory perturbations in young and older adults. *Medical & Biological Engineering & Computing*, *56*(12), 2325-2335. <https://doi.org/10.1007/s11517-018-1855-7>
- Ryser, D. K., Erickson, R. P., & Cahalan, T. (1988). Isometric and isokinetic hip abductor strength in persons with above-knee amputations. *Archives of Physical Medicine and Rehabilitation*, *69*(10), 840-845. <https://doi.org/10.5555/uri:pii:0003999388900056>

- Saidouni, T., & Bessonnet, G. (2003). Generating globally optimised sagittal gait cycles of a biped robot. *Robotica*, 21(2), 199-210. <https://doi.org/10.1017/S0263574702004691>
- Sheridan, P. L., Solomont, J., Kowall, N., & Hausdorff, J. M. (2003). Influence of Executive Function on Locomotor Function : Divided Attention Increases Gait Variability in Alzheimer's Disease. *Journal of the American Geriatrics Society*, 51(11), 1633-1637. <https://doi.org/10.1046/j.1532-5415.2003.51516.x>
- Sherman, M. A., Seth, A., & Delp, S. L. (2011). Simbody : Multibody dynamics for biomedical research. *Procedia IUTAM*, 2, 241-261. <https://doi.org/10.1016/j.piutam.2011.04.023>
- Shourijeh, M. S., & McPhee, J. (2015). Foot-ground contact modeling within human gait simulations : From Kelvin-Voigt to hyper-volumetric models. *Multibody System Dynamics*, 35(4), Art. 4. <https://doi.org/10.1007/s11044-015-9467-6>
- Slout, L. H., & van der Krogt, M. M. (2018). Interpreting Joint Moments and Powers in Gait. In *Handbook of Human Motion* (p. 625-643). Springer International Publishing. [https://doi.org/10.1007/978-3-319-14418-4\\_32](https://doi.org/10.1007/978-3-319-14418-4_32)
- Sok, K. W., Kim, M., & Lee, J. (2007). Simulating Biped Behaviors from Human Motion Data. *ACM Transactions on Graphics*, 26(3), 10.
- Srinivasan, M., & Ruina, A. (2006). Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439(7072), Art. 7072. <https://doi.org/10.1038/nature04113>
- Thelen, D. G., & Anderson, F. C. (2006). Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. *Journal of Biomechanics*, 39(6), Art. 6. <https://doi.org/10.1016/j.jbiomech.2005.02.010>
- Thompson, D. M., & Haran, D. (1983). Living with an amputation : The patient. *International Rehabilitation Medicine*, 5(4), 165-169. <https://doi.org/10.3109/03790798309167040>

- Tinetti, M. E., Speechley, M., & Ginter, S. F. (1988). Risk Factors for Falls among Elderly Persons Living in the Community. *New England Journal of Medicine*, 319(26), 1701-1707. <https://doi.org/10.1056/NEJM198812293192604>
- Tisserand, R., Robert, T., Dumas, R., & Chèze, L. (2016). A simplified marker set to define the center of mass for stability analysis in dynamic situations. *Gait & Posture*, 48, 64-67. <https://doi.org/10.1016/j.gaitpost.2016.04.032>
- Tropp, H., & Odenrick, P. (1988). Postural control in single-limb stance. *Journal of Orthopaedic Research*, 6(6), 833-839. <https://doi.org/10.1002/jor.1100060607>
- Tsai, Y.-Y., Lin, W.-C., Cheng, K. B., Lee, J., & Lee, T.-Y. (2010). Real-Time Physics-Based 3D Biped Character Animation Using an Inverted Pendulum Model. *IEEE Transactions on Visualization and Computer Graphics*, 16(2), Art. 2. <https://doi.org/10.1109/TVCG.2009.76>
- ttk592. (2022). *Ttk592/spline* [C++]. <https://github.com/ttk592/spline> (Original work published 2015)
- van den Bogert, A. J., Gerritsen, K. G. M., & Cole, G. K. (1998). Human muscle modelling from a user's perspective. *Journal of Electromyography and Kinesiology*, 8(2), 119-124. [https://doi.org/10.1016/S1050-6411\(97\)00028-X](https://doi.org/10.1016/S1050-6411(97)00028-X)
- van de Panne, M. (1993). Sensor-actuator networks. *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '93*, 335-342. <https://doi.org/10.1145/166117.166159>
- van de Panne, M., Kim, R., & Fiume, E. (1994). *Virtual Wind-up Toys for Animation*. 8.
- Vukobratović, M., & Borovac, B. (2004). Zero-moment point—Thirty five years of its life. *International Journal of Humanoid Robotics*, 01(01), 157-173. <https://doi.org/10.1142/S0219843604000083>
- Wang, J. M., Hamner, S. R., Delp, S. L., & Koltun, V. (2012). Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Transactions on Graphics*, 31(4), Art. 4. <https://doi.org/10.1145/2185520.2185521>

- Wang, T.-Y., Bhatt, T., Yang, F., & Pai, Y.-C. (2012). Adaptive control reduces trip-induced forward gait instability among young adults. *Journal of Biomechanics*, 45(7), 1169-1175. <https://doi.org/10.1016/j.jbiomech.2012.02.001>
- Watson, F., Fino, P. C., Thornton, M., Heracleous, C., Loureiro, R., & Leong, J. J. H. (2021). Use of the margin of stability to quantify stability in pathologic gait – a qualitative systematic review. *BMC Musculoskeletal Disorders*, 22(1), Art. 1. <https://doi.org/10.1186/s12891-021-04466-4>
- Webster, K. E., Merory, J. R., & Wittwer, J. E. (2006). Gait Variability in Community Dwelling Adults With Alzheimer Disease. *Alzheimer Disease & Associated Disorders*, 20(1), 37-40. <https://doi.org/10.1097/01.wad.0000201849.75578.de>
- Wieber, P.-B. (2002). *On the stability of walking systems*. Proceedings of the International Workshop on Humanoid and Human Friendly Robotics. <https://hal.inria.fr/inria-00390866>
- Wooten, W. L., & Hodgins, J. K. (2000). Simulating leaping, tumbling, landing and balancing humans. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 1, 656-662 vol.1. <https://doi.org/10.1109/ROBOT.2000.844127>
- Wu, J., & Popović, Z. (2010). Terrain-adaptive Bipedal Locomotion Control. *ACM SIGGRAPH 2010 Papers*, 72:1-72:10. <https://doi.org/10.1145/1833349.1778809>
- Xiang, Y., Arora, J. S., & Abdel-Malek, K. (2010). Physics-based modeling and simulation of human walking : A review of optimization-based and other approaches. *Structural and Multidisciplinary Optimization*, 42(1), Art. 1. <https://doi.org/10.1007/s00158-010-0496-8>
- Yang, P.-F., Laszlo, J., & Singh, K. (2004). Layered dynamic control for interactive character swimming. *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '04*, 39. <https://doi.org/10.1145/1028523.1028529>

Young, S. (1995). Analysis of Moving Walkway Use in Airport Terminal Corridors. *Transportation Research Record*, 8.

Zhao, P., & van de Panne, M. (2005). User interfaces for interactive control of physics-based 3D characters. *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games - SI3D '05*, 87. <https://doi.org/10.1145/1053427.1053441>

Zordan, V. B. (2002). *Motion capture-driven simulations that hit and react*. 9.