THÈSE

Pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITÉ DE POITIERS
UFR des sciences fondamentales et appliquées
Pôle poitevin de recherche pour l'ingénieur en mécanique, matériaux et énergétique – PPRIME (Poitiers)
(Diplôme National – Arrêté du 25 mai 2016)

École doctorale : Sciences et ingénierie des matériaux, mécanique, énergétique et aéronautique – SIMMEA (Poitiers)
Secteur de recherche : Génie mécanique, productique, transport

Présentée par :

Jean-Baptiste Riccoboni

Modélisation Dynamique d'Arborescences – application à la robotique et à l'étude du mouvement humain

Directeur(s) de Thèse :
Antoine Eon, Tony Monnet

Soutenue le 16 décembre 2022 devant le jury

Jury :

Rôle	Nom	Institution	CNU	Grade
Rapporteur	David Daney	INRIA Bordeaux		DR
Rapporteur	Bruno Watier	Université de Toulouse	74	PU
Examinatrice	Sophie Sakka	INSHEA	71	PU
Examinateur	Olivier Bonneau	Université de Poitiers	60	PU
Examinateur	Antoine Eon	Université de Poitiers	60	MCF
Examinateur	Tony Monnet	Université de Poitiers	74	PU

Pour citer cette thèse :
Jean-Baptiste Riccoboni. *Modélisation dynamique d'arborescence – application à la robotique et à l'étude du mouvement humain* [En ligne]. Thèse Génie mécanique, productique, transport. Poitiers : Université de Poitiers, 2022. Disponible sur Internet http://theses.univ-poitiers.fr

Remerciements

Mon doctorat fut plutôt chouette et ça c'est grâce aux personnes qui m'ont accompagné tout au long de cette aventure mais également celles qui m'y ont amené.

Ma première pensée est adressée à Antoine, qui fait partie des deux catégories en même temps. Tout d'abord pour le très bon enseignant et guide qu'il sut être pour moi, me permettant à la fois de décrocher une place à l'École Nationale Supérieure d'Arts et Métiers lorsque j'étais à l'IUT, puis pour l'excellent directeur qu'il a été, aux conseils toujours avisés et avec qui la communication fut d'une simplicité plus qu'agréable. Cette thèse n'aurait pas été le dixième de ce qu'elle est devenue sans lui. T'es vraiment un chic type. Merci l'ami !

Tout de suite après vient Arnaud le gars sûr, avec qui on s'est bien amusé, que ce soit devant un gymnaste qui se luxe les épaules (en simulation) ou devant un bon thé chaud à blablater. Tu m'as dépanné et instruit sur MATLAB un paquet de fois, merci encore. Par contre va vraiment falloir faire un truc avec tes codes, c'est plus possible.

Un grand merci à Tony également, pour sa direction, sa gentillesse et sa bienveillance, en stage comme en doctorat. L'été de la rédaction de notre (premier ?) article commun fut un été dont je me souviendrai, et en bien. Je n'ai toujours pas compris comment il faut compter les points avec le panier de basket de la cafétaria mais ça n'a pas vraiment d'importance tant que ça te va.

Je souhaite remercier bien évidemment Madame Sophie Sakka, Professeure à l'Institut national supérieur de formation et de recherche pour l'éducation des jeunes handicapés et les enseignements adaptés, Monsieur Olivier Bonneau, Professeur à l'Université de Poitiers, et en particulier les rapporteurs Monsieur David Daney, Directeur de Recherche à l'Institut national de recherche en sciences et technologies du numérique, et Monsieur Bruno Watier, Professeur à l'Université de Toulouse, pour avoir accepté de faire partie du jury de cette thèse.

Ensuite il y a beaucoup de monde ! Dans le désordre : Bastien le super cobur'squet peintre à ses heures, Célestin la balance au grand cœur, Andrian le stalker aux sens gustatifs défaillants mais bon gars, Romain T. et nos discussions philosophiquo-existentielles, Leo le calimero, Raphaël au modèle dynamique le plus improbable que je connaisse mais dont la bonté est sans limite, Pierre et ses judicieuses remarques, Pascal pour son humour bien à lui, Jean-Pierre pour sa beaufit- bonhommie, Romain B. le meilleur taxi et le meilleur safety, Marien pour ses analyses grinçantes mais pas si loin de la réalité, Laurence et sa bonne humeur, Mathieu M. pour son support technique irréprochable, Quentin avec qui on ne sait plus ce qui est sujet et hors-sujet, Margot et ses convictions, Laetitia et son amour pour le chocolat, Mathieu D. et nos passes à l'ultimate, Maeva et l'absence de passes au volley, Aurélie pour son énigmatisme savamment dosé… Merci à vous et les autres pour la chaleureuse ambiance à laquelle vous contribuez tous !

Merci à ma mère Valérie, ma grand-mère Jacqueline, ma sœur Camille et le reste de ma famille pour son indéfectible soutien dans les bons moments comme dans les moins bons. Le sang de la veine en somme.

Je remercie enfin Sarah, la source infinie de bonheur et de motivation qui a transformé la fin de ce doctorat.

À celles et ceux qui font de leur mieux,

Table des matières

Remerciements 3

I. Le balancement des membres supérieurs lors de la marche humanoïde et les façons de les modéliser 11

1. Un mouvement qui n'est pas nécessaire… mais pas inutile non plus 11
2. Des outils existent… mais aucun n'est adapté 15
3. Explication du plan 17

II. La Modélisation Dynamique d'Arborescences, description de l'outil et de son utilisation 19

1. Ce que fait la MDA en résumé 19
2. Liste des fichiers utilisés 20
3. Les étapes pour utiliser la MDA 20
4. Décrire l'arborescence 21
   a. Informations sur l'arborescence modélisée 21
   b. Formes du modèle 23
   c. Traitement des équations 26
   d. Génération des fichiers 29
   e. Éléments à calculer 29
   f. Description de la morphologie du système matériel 33
   g. Paramètres fournis par l'utilisateur 34
   h. Paramètres inertiels et géométriques 35
   i. Paramètres de position et d'orientation 36
   j. Paramètres des actions mécaniques extérieures ponctuelles 39
5. Initialiser des paramètres 42
   a. Informations sur l'individu 42
   b. Initialisation des paramètres fournis par l'utilisateur 42
6. Exécuter le code source 43
7. Explication de l'architecture du code source 43
   a. Génération du modèle dynamique 44
   b. Vérification optionnelle des données 45
   c. Génération des fichiers 46
8. Exemple de performances 47
9. Proposition d'une méthode numérique 48
   a. Application au pendule pesant 48
   b. Application à l'oscillateur harmonique amorti 61

III. La figure du passement filé, exemple d'application à l'étude du mouvement humain 75

1. Origine et objectifs du projet BE-MORE 75

2. Identification des paramètres 78
a. Données disponibles 78
b. Description du mouvement étudié 81
c. Modélisation du gymnaste « réel » sous forme de mécanisme 81
d. Processus d'optimisation 83
e. Résultats 86
f. Discussion 88
3. Synthèse du geste optimisé 88
a. Modélisation du gymnaste « exemplaire » sous forme de mécanisme 88
b. Processus d'optimisation 91
c. Résultats 92
d. Discussion 95
4. Perspective de développement 96
IV. La bipédie d'ORHRO, exemple d'application à la robotique 99
1. Présentation de l'existant 101
2. Ce qui est mesuré 103
3. Modélisation du robot actuel et des membres supérieurs virtuels 105
4. Cas étudiés 107
5. Description du processus d'optimisation 108
6. Résultats 110
a. Passif avec l'articulation du coude bloquée 112
b. Passif avec l'articulation du coude libre 118
c. Actif commandé en proportion de l'angle de flexion de la hanche 124
d. Actif commandé en proportion de l'angle de lacet du tronc 130
7. Comparaison et discussion 136
a. Valeurs optimisées des paramètres inertiels et géométriques 136
b. Coefficients de frottement maximaux nécessaires 143
c. Écarts entre les positions du PAMB et les références 145
V. Le résumé de ce qui a été fait et de ce qu'il reste à faire 147
VI. Annexes 151
a. La Modélisation Dynamique d'Arborescences, description de l'outil et de son utilisation 151
b. La figure du passement filé, exemple d'application à l'étude du mouvement humain 151
c. La bipédie d'ORHRO, exemple d'application à la robotique 152
VII. Bibliographie 155
VIII. Productions scientifiques de l'auteur pendant la thèse : 161

I. Le balancement des membres supérieurs lors de la marche humanoïde et les façons de les modéliser

Dans cette première partie est présentée une description, introductive et appuyée par la littérature, du phénomène du balancement des membres supérieurs lors de la marche humanoïde, notamment vis-à-vis de ses causes et de ses conséquences. Elle est suivie d'un état de l'art des outils qui permettraient de le modéliser.

1. Un mouvement qui n'est pas nécessaire… mais pas inutile non plus

C'est une expérience que beaucoup peuvent faire. Se tenir debout, mettre les mains dans les poches et un pied devant l'autre. Les bras ne se balancent pas, ou très peu, et pourtant il est tout-à-fait possible de marcher. En effet, ce mouvement des membres supérieurs, observable naturellement dans la bipédie humaine, n'est pas nécessaire à la locomotion. Alors pour quelles raisons existe-t-il ? Et quels sont les mécanismes le générant ? Ces questions ont fait l'objet de nombreuses recherches, ciblant tour à tour les différents aspects du balancement des bras dans la marche humaine.

Puisqu'il est important de définir et de décrire un sujet d'étude avant toute autre chose, il semble judicieux de citer des travaux comme ceux de Murray et al. en 1967 [1], afin que tout un chacun puisse visualiser le mouvement considéré. Les valeurs en degrés des flexions/extensions à l'épaule et au coude en fonction du pourcentage de complètement du cycle de marche pour deux vitesses de marche, une libre et une rapide, y sont données par exemple ; allant de -21 à 7 degrés pour l'épaule et de 22 à 44 degrés pour le coude en vitesse de marche libre et comptant positivement la flexion et négativement l'extension.

En 1965, Ballesteros et al. [2] s'interrogeaient sur les causes de ce balancement, ayant mesuré des potentiels d'action au niveau des muscles des bras et de l'épaule au cours de la marche, y compris lorsque le mouvement des membres supérieurs était empêché. Cette étude postulait qu'il y a une part active non négligeable dans la génération du balancement des bras.

À l'aide d'électromyogrammes, Basmajian et Tuttle (1973 [3]) mirent en avant de surprenantes similitudes dans les principes contrôlant les contractions musculaires impliquées dans la locomotion de l'homme et du gorille, aux allures pourtant clairement différentes. Ils notèrent que les muscles étaient sollicités lors de l'apparition d'un mouvement dynamique mais rarement dans sa suite. Des mécanismes posturaux étaient soupçonnés d'autoriser la relaxation des muscles dès lors que leur contribution n'était plus nécessaire, autrement dit, que le comportement passif des corps suffisait au mouvement. Leurs travaux posaient parmi les premières bases de la réflexion sur les contributions au cours du mouvement des parties passives et actives du balancement. Bases que Jackson et al. ont utilisées en 1978 [4] lorsqu'ils proposèrent un modèle mathématique intégrant, entre autres données biomécaniques, les profils de l'activité électromyographique de certains muscles du bras et les accélérations de l'articulation de l'épaule enregistrés lors de marches humaines à diverses vitesses. Ils essayèrent ainsi de prédire le mouvement des bras résultant et avancèrent que la partie active du

balancement aurait pour but de réduire les saccades provoquées par la partie passive. Il est intéressant de noter ici qu'en 1983, Jackson [5] pensait que ce mouvement était contrôlé par de très vieux (au sens évolutionnaire du terme) mécanismes neuraux dont l'être humain aurait hérité d'animaux utilisant leurs membres supérieurs pour se déplacer et qu'il n'aidait pas à la minimisation de la rotation autour de l'axe longitudinal du corps ; conclusions sur lesquelles la quatrième partie de ce document reviendra.

Eke-Okoro et al. firent en 1997 [6] le raisonnement à l'envers, c'est-à-dire d'imposer des changements délibérés au balancement des bras (un ou deux bras attachés au corps, bras parallèles, bras synchrones avec les jambes, etc.) et d'étudier les altérations que cela causait sur la marche. Leur principal résultat fut que dans tous les cas de changements, la vitesse de marche fut diminuée. Dans une optique similaire, Topuz et al. étudièrent plus tard (2019 [7]) les conséquences du balancement des membres supérieurs sur les caractéristiques spatio-temporelles de la marche chez les amputés unilatéraux transhuméraux. Ils observèrent que les longueurs des pas et de la foulée de ces derniers étaient plus courtes que celles du groupe contrôle, de même, leurs vitesse et fréquence de marche étaient plus faibles.

Park proposa en 2008 [8] une méthode de synthèse du mouvement de balancement des membres supérieurs basée sur l'effort de rotation autour de la verticale généré par le déplacement des membres inférieurs lors de la marche. Cet effort, détaillé dans la quatrième partie de ce document, est lié à un risque de glissement des pieds entraînant un pivotement incontrôlé. Il est donc considéré comme un élément préjudiciable à la marche. En imposant que cet effort soit limité, Park est parvenu à reproduire un mouvement de balancement des bras, en opposition du mouvement des jambes comme c'est le cas dans la marche humaine. Ces travaux présentent l'une des premières conséquences positives de ce mouvement et apportent donc un élément de réponse sur les raisons de son existence.

D'autres conséquences positives furent ensuite présentées, Umberger indiquait en 2008 [9] que la dépense énergétique brute et nette augmentaient de manière significative lorsque les sujets de son étude marchaient sans balancer leurs bras. Résultats que Yizhar et al. confirmaient l'année d'après (2009 [10]).

Pontzer et al. (2009 [11]) proposèrent à partir de données expérimentales une modélisation dans laquelle le mouvement du haut du corps était essentiellement dû aux mouvements du bas du corps. Le tronc et les épaules agissaient principalement comme liens élastiques entre le bassin, la ceinture scapulaire et les bras. Ces derniers jouant le rôle d'amortisseurs réduisant la rotation de la tête et du tronc. Cette approche mit en avant la contribution de la partie passive de la génération du balancement des bras comme partie majoritaire.

Les travaux de Collins et al. en 2009 [12] firent un lien entre consommation d'énergie métabolique diminuée et génération passive du balancement, supposant que cette dernière permettait de bénéficier à moindre coût de la réduction de l'effort de rotation autour de la verticale.

Cependant, Barthelemy et Nielsen rappelaient l'année d'après (2010 [13]) le rôle de la partie active au travers de leur recherche sur la contribution du cortex moteur à l'activité électromyographique dans les muscles des bras.

C'est en 2012 que Kuhtz-Buschbeck et al. [14], en mesurant l'activité électromyographique pour plusieurs types de balancement des bras, proposèrent de ne plus opposer les parties actives et passives mais plutôt de considérer leurs contributions relatives et de poser la question de leurs proportions.

Ainsi, Goudriaan et al. (2014 [15]) utilisèrent un modèle musculosquelettique à l'aide du logiciel OpenSim [16] afin de générer des simulations dynamiques de marche avec et sans excitation des muscles des membres supérieurs. Ils observèrent que les amplitudes et la phase relative des balancements uniquement passifs étaient faibles, au point que les bras pouvaient se retrouver en phase avec le mouvement des jambes (et non en opposition de phase comme il est possible de l'observer chez l'humain). Ils en déduisirent que l'activité musculaire était nécessaire afin d'augmenter l'amplitude des balancements et de modifier la phase relative. Néanmoins leur modèle ne parvint pas à restituer un mouvement semblable à celui des bras humains.

Un modèle statistique fut proposé par Hejrati et al. en 2016 [17] générant des balancements plus fidèles à la réalité. Hélas, n'étant pas basé sur la dynamique, ce modèle n'apportait pas de réponse quant à la question de la proportion des parties actives et passives.

En 2018, Canton et MacLellan [18] réalisèrent une expérience durant laquelle ils firent marcher des sujets dont ils restreignirent les mouvements au niveau du bassin. Dans cette configuration, l'amplitude du balancement des membres supérieurs diminua, apportant selon eux la preuve que la partie passive de la génération du balancement est un facteur significatif du contrôle de ce dernier. Il est intéressant de souligner que, même dans cette configuration, des motifs d'activité musculaire et un couplage temporel du mouvement des membres furent conservés, suggérant un maintien actif sous-jacent d'un modèle de locomotion via des réseaux neuraux liés entre les membres inférieurs et supérieurs. Les recherches de Weersink et al. (2019 [19]), qui portaient sur l'analyse d'électroencéphalogrammes de l'aire motrice supplémentaire du cerveau, pointaient sur la contribution de cette dernière à des mouvements cycliques en opposition de phase des membres inférieurs et supérieurs ; supportant donc l'hypothèse d'un soutien cortical du balancement des bras dans la marche.

Pour aller plus loin, un article de revue de Meyns et al publié en 2013 [20] reprend une partie des éléments présentés ici tout en élargissant le débat, comme par exemple sur la valeur thérapeutique de la compréhension du phénomène du balancement des membres supérieurs dans la marche. Ce champ de recherche fut alimenté par la suite via notamment l'étude de Romkes et Bracht-Schweizer en 2017 [21] qui avait pour but d'aider à mieux dissocier les effets de la vitesse de marche de ceux d'une pathologie sous-jacente sur les balancements des bras. La question de la stabilité de la marche est également intéressante et les résultats publiés par Bruijn et al. en 2010 [22] suggéraient que le balancement des membres supérieurs rendait la marche plus sensible aux perturbations mais que les

mouvements de récupération qu'ils permettaient contribuaient à la stabilité générale de la marche humaine. Les travaux de Cheng et al. (2014 [23] et 2015 [24]) allèrent également dans ce sens.

Il apparaît clair à présent que, bien que l'étude du balancement des membres supérieurs lors de la marche de l'humanoïde ait fait l'objet d'une recherche active et ce depuis de nombreuses années, la communauté est loin d'avoir fait le tour de tout ce qu'il y a à comprendre à son sujet. Ses conséquences furent les premières à être identifiées, du moins en partie. Ses causes, en revanche, suscitent encore de nombreuses questions et parmi elles, la nature même de sa génération : quelles sont les proportions respectives des parties passive et active du balancement des bras ? À quel point dépendent-elles de l'individu ?

La création d'un modèle suffisamment fidèle et personnalisable pourrait bien être la prochaine étape dans l'obtention des réponses à ces questions.

La problématique concernant la conception d'un modèle adapté ne concerne pas que l'étude de la bipédie humaine. Des travaux tels que ceux de Kaddar et al. (2015 [25]) se penchèrent sur les effets du mouvement des bras d'un robot humanoïde lors de sa marche. Les impacts de la coïncidence de la fréquence naturelle des bras avec celle de la marche sur l'amplitude du balancement des bras, obtenue via l'ajout de ressorts, y sont mentionnés.

L'idée d'utiliser les robots humanoïdes pour étudier l'humain n'est pas récente. Atkeson et al. publiait à ce sujet en 2000 [26]. Les mêmes thématiques de recherche s'y retrouvent, comme la notion d'économie d'énergie avec Collins et Ruina (2005 [27]), la génération de mouvement à partir du balancement des membres supérieurs avec Osaku et al. (2006 [28]) ou la détermination du balancement optimal avec Aoustin et Formal'skii (2008 [29]).

Certains travaux établissent clairement une passerelle entre humains et robots. Boutin et al. proposaient en 2011 [30] d'utiliser des enregistrements de mouvements humains comme données d'entrée pour commander la locomotion de robots humanoïdes. Zhang et al (2013 [31]) suggérait d'utiliser la logique de compensation de l'effort de rotation autour de la verticale chez l'humain pour piloter le mouvement des robots humanoïdes.

Au-delà de sa relative ressemblance à l'être humain, le robot humanoïde présente deux qualités très importantes :

1. Sa modélisation est simple car il est constitué de matières homogènes, dont les propriétés mécaniques sont connues avec précision. Il en va de même pour le comportement de ses articulations.

2. Il exécute ce qu'il lui est demandé. Si la commande est qu'il ne doit pas penser à ses bras et qu'il doit les laisser « ballants », c'est ce qu'il fera. C'est un cas quasiment impossible à obtenir chez l'être humain.

Étudier le balancement des bras d'un robot humanoïde pendant sa marche pourrait apporter des réponses transposables aux questions concernant l'humain.

2. Des outils existent… mais aucun n'est adapté

Il faut tout d'abord mentionner que le modèle devra nécessairement prendre en compte la réalité physique des membres supérieurs mais aussi du reste du corps, que ce soit celui d'un humain ou d'un robot humanoïde. En effet, impossible de comprendre leurs comportements et leurs interactions sans intégrer à cette étude la dynamique, partie de la mécanique établissant le lien entre un mouvement et les forces qui en sont à l'origine.

La recherche de l'outil se cantonnera donc aux générateurs de modèles dynamiques. Deux familles peuvent être distinguées : les générateurs numériques et les générateurs symboliques. Dans les faits, ces deux familles sont à même de répondre à notre problématique mais elles le font différemment.

La différence ayant le plus d'importance dans le cas présent réside dans le moment où l'utilisateur doit renseigner les valeurs numériques du système qu'il étudie.

Les générateurs numériques exigent que ces valeurs soient renseignées avant la création du modèle. Si l'utilisateur souhaite par exemple piloter un robot afin qu'il réalise un mouvement donné, l'utilisateur devra connaître, entre autres choses, les valeurs des masses et des longueurs des pièces constituant ce robot avant que le générateur ne puisse lui donner la valeur des efforts à appliquer aux articulations du robot. Le principal avantage d'un générateur numérique est que, puisqu'il manipule uniquement des nombres – ce qu'un ordinateur sait très bien faire –, fournir ce genre de réponse lui est très facile. Il est donc capable de le faire très rapidement même si c'est la première fois qu'il modélisait ce robot. Beaucoup de générateurs numériques existent, tels que Havok [32], DART [33], PhysX [34], Pinocchio [35], MuJoCo [36], RBDL [37], ODE [38], Simbody [39]… Mais comment procéder si l'une des pièces du robot est en cours de conception et que les valeurs numériques de ses paramètres n'ont pas été choisies ? Ou que l'utilisateur souhaite légitimement les choisir vis-à-vis du comportement dynamique du robot ? Ou encore qu'il souhaite modéliser un être humain pour lequel ces valeurs sont connues avec une part d'incertitude ? Deux options sont possibles : relancer le générateur à chaque changement de valeur ou utiliser les générateurs symboliques.

Les générateurs symboliques ne demandent les valeurs numériques qu'après la création du modèle. Comment ? En manipulant des symboles plutôt que des nombres. En reprenant l'exemple du robot, un modèle symbolique ne donnera non pas les valeurs des efforts à appliquer aux articulations du robot mais leurs expressions sous forme mathématique, une formule composée de symboles qui seront remplacés par leurs valeurs au moment où l'utilisateur en aura besoin. Cela permet de tester, une fois le modèle créé, beaucoup de valeurs différentes en très peu de temps, encore moins que pour les modèles numériques. La détermination de valeurs numériques proches des valeurs réelles est communément appelée identification et sera une étape cruciale pour la problématique présentée ici. Évidemment cette méthode de génération a un inconvénient : générer un modèle symbolique prend plus de temps qu'un modèle numérique justement car le générateur manipule des symboles et non

des nombres. Chaque famille de générateur a ses avantages et ses inconvénients et l'usage de l'une par rapport à l'autre dépend des besoins. Pour une application telle que celle de l'étude des causes du balancement des membres supérieurs dans la marche humanoïde, les modèles symboliques sont les plus appropriés du fait de l'important nombre d'inconnues que comportera la modélisation de l'être humain.

Et là aussi il en existe quelques-uns. Les publications de Racković et al. (1998 [40]), McPhee et al. (2004 [41]), Dean-Leon et al. (2012 [42]), Dallali et al. (2013 [43], COMAN), Docquier et al. (2013 [44], ROBOTRAN) et Khalil et al. (2014 [45], OpenSYMORO), pour ne citer qu'eux, sont tout autant de générateurs de modèles dynamiques symboliques. Alors lequel choisir ? Ce sont les fonctionnalités proposées qui importent. L'outil doit obligatoirement :

- permettre de calculer les modèles dynamiques direct et inverse (correspondant respectivement aux parties passives et actives du balancement des bras) mais également de calculer n'importe quelle grandeur telle que la position dans l'espace au cours du temps d'un point d'intérêt du système étudié (comme un centre d'inertie par exemple), le moment cinétique d'un de ses solides, l'action mécanique globale exercée par l'extérieur, etc. tout cela afin de pouvoir considérer toutes les pistes potentielles permettant d'expliquer les causes du balancement des bras.

- autoriser le formalisme symbolique sur l'intégralité des paramètres du système étudié, y compris la possibilité d'une dépendance temporelle compte tenu des propriétés mécaniques du corps humain.

- bénéficier d'une flexibilité la plus grande possible dans les méthodes de description des paramètres inertiels, géométriques, de positionnement, d'orientation et des actions mécaniques extérieures pour maximiser la finesse des modèles conçus.

- générer des modèles ne dépendant d'aucune bibliothèque de calcul extérieure afin de pouvoir être implémentés sur des automates industriels, étape nécessaire à la validation expérimentale sur les robots disponibles au laboratoire des résultats obtenus par la simulation.

- générer des modèles dont l'évaluation, c'est-à-dire le remplacement des symboles par des valeurs numériques, soit la plus rapide possible afin de minimiser le temps des simulations.

Hélas il semblerait qu'au terme de notre recherche, aucune des solutions existantes ne remplissait toutes les conditions susmentionnées. Le besoin d'un tel générateur de modèles dynamiques étant partagé par de nombreux projets portés par l'équipe Robotique Biomécanique Sport Santé (RoBioSS), équipe au sein de laquelle les travaux présentés dans ce document eurent lieu, la décision de concevoir cet outil fut prise.

3. Explication du plan

Afin de répondre à la problématique, voici le plan de recherche qui fut élaboré :

Numéro de l'étape	Description de l'étape	Commentaire
1	Concevoir l'outil de modélisation adéquat.	L'algorithme réalisé et son fonctionnement font l'objet de la deuxième partie de ce document.
2	Validation de la capacité de l'outil à l'identification de paramètres.	Un exemple d'identification et de génération de mouvement optimisé chez l'humain est présenté dans la troisième partie de ce document.
3	Validation de la capacité de l'outil à modéliser finement et rapidement.	Un exemple de modèle d'humanoïde équipé de membres supérieurs entièrement paramétrables et de ses performances de calcul est détaillé dans la quatrième partie de ce document.

Tableau 1. Plan de recherche

Cette première partie étant terminée, la partie suivante présentera la Modélisation Dynamique d'Arborescences, l'algorithme de génération de modèles dynamiques symboliques développé pour répondre aux exigences de l'étude du phénomène du balancement des membres supérieurs lors de la marche humanoïde.

II. La Modélisation Dynamique d'Arborescences, description de l'outil et de son utilisation

Dans cette deuxième partie est expliqué l'essentiel de ce qu'il y a à savoir sur la Modélisation Dynamique d'Arborescences (MDA), de la façon dont il faut procéder pour l'utiliser au détail de l'architecture de son code source.

1. Ce que fait la MDA en résumé

La MDA génère des modèles cinématiques, cinétiques et dynamiques de mécanismes pouvant être représentés comme des arborescences à chaînes ouvertes. Ces modèles se présentent sous la forme de fichiers contenant les expressions des grandeurs demandées, qui peuvent être écrites symboliquement ou au sein d'une fonction pour calculer leurs valeurs numériques.

L'utilisateur commence par renseigner la description morphologique de la structure à modéliser, à laquelle il rajoute les paramètres géométriques et inertiels, les paramètres de pilotage et les paramètres des actions mécaniques exercées par le milieu extérieur sur l'arborescence. Cette description se fait à l'aide d'un fichier texte.

Ce fichier est lu par le logiciel Maple. Si certains paramètres nécessaires aux calculs à effectuer n'ont pas été renseignés par l'utilisateur, ils seront alors automatiquement créés. Ils constitueront par ailleurs une partie des arguments des fonctions calculant les valeurs numériques des expressions.

Maple utilise ensuite les paramètres renseignés pour définir les éléments mathématiques nécessaires : matrices de passage, vecteurs rotations et de déplacement, torseurs cinématiques, cinétiques, dynamiques, d'actions mécaniques extérieures…

Une fois tous ces éléments réunis, l'algorithme écrit le Principe Fondamental de la Dynamique. Il obtient ainsi les équations de la dynamique qui permettront d'exprimer les éléments de réduction des torseurs des actions mécaniques de liaison.

En fonction du souhait de l'utilisateur, ce système d'équations sert de base à la création des modèles direct et/ou inverse du système étudié. Il est par ailleurs possible de demander l'expression de n'importe quelle grandeur définie par la MDA, telles que la position d'un point de l'arborescence dans le repère de l'étude, les composantes d'une matrice de passage…

Ces modèles sont ensuite exportés dans des fichiers TEX pour obtenir les expressions formelles des grandeurs demandées ou dans des fichiers C++ comme fonctions permettant d'en calculer les valeurs numériques. Ces derniers ne dépendent d'aucune bibliothèque externe et ne requièrent pas d'allocations dynamiques de mémoire ; ils peuvent donc être nativement utilisé par n'importe quel système capable de compiler du code C++.

2. Liste des fichiers utilisés

Pour pouvoir utiliser la MDA, deux fichiers sont nécessaires :

1. Le code source écrit dans un fichier intitulé « VXpX_Modelisation_Dynamique_Arborescence.mw » où VXpX désigne la version de la MDA. À l'heure de la rédaction de ce document, le code est en version 1.85 et le fichier s'intitule donc « V1p85_Modelisation_Dynamique_Arborescence.mw ».

2. La description de l'arborescence à modéliser. Un fichier prérempli, contenant des explications sur les différents éléments à renseigner et la façon de le faire, est disponible pour faciliter le travail de description. Il est nommé « VXpX_VIERGE_Description_arborescence.txt ».

Un troisième fichier, optionnel, peut être utilisé :

3. L'initialisation de paramètres utilisés par le fichier de description. Ce fichier permet de modéliser l'arborescence en apportant des modifications – le plus souvent des simplifications – sans avoir à modifier le fichier de description qui peut ainsi être générique. Là encore, il existe un fichier prérempli avec la même fonction que le précédent. Il est nommé « VXpX_VIERGE_Parametres_utilisateur.txt ».

3. Les étapes pour utiliser la MDA

Il y a quatre étapes :

1. Ouvrez le fichier de description d'arborescence, remplissez-le et sauvegardez-le.

2. Si vous le souhaitez, ouvrez le fichier d'initialisation de paramètres, remplissez-le et sauvegardez-le.

3. Ouvrez le fichier du code source et exécutez l'entièreté de la feuille de calcul.

4. Une fenêtre de sélection s'ouvre. Choisissez votre fichier de description et cliquez sur « Lire ».

Débute alors la génération du modèle puis celle des fichiers que vous avez demandés. Une fois l'écriture de ces derniers terminée, vous pouvez les utiliser.

4. Décrire l'arborescence

Nous avons fait le choix de passer en revue ici le fichier de description vierge. Malgré le fait que cette approche paraît s'apparenter à un mode d'emploi, ce qui n'est pas habituel dans un document exposant des travaux de recherche, cette méthode nous permet de montrer toutes les capacités de la MDA et la grande flexibilité de son code. Des captures d'écran du fichier « V1p85_VIERGE_Description_arborescence.txt » seront présentées à des fins d'illustration.

En guise d'exemple simple de description, le mécanisme suivant, composé d'une liaison glissière et d'une liaison pivot, a été décrit puis modélisé :

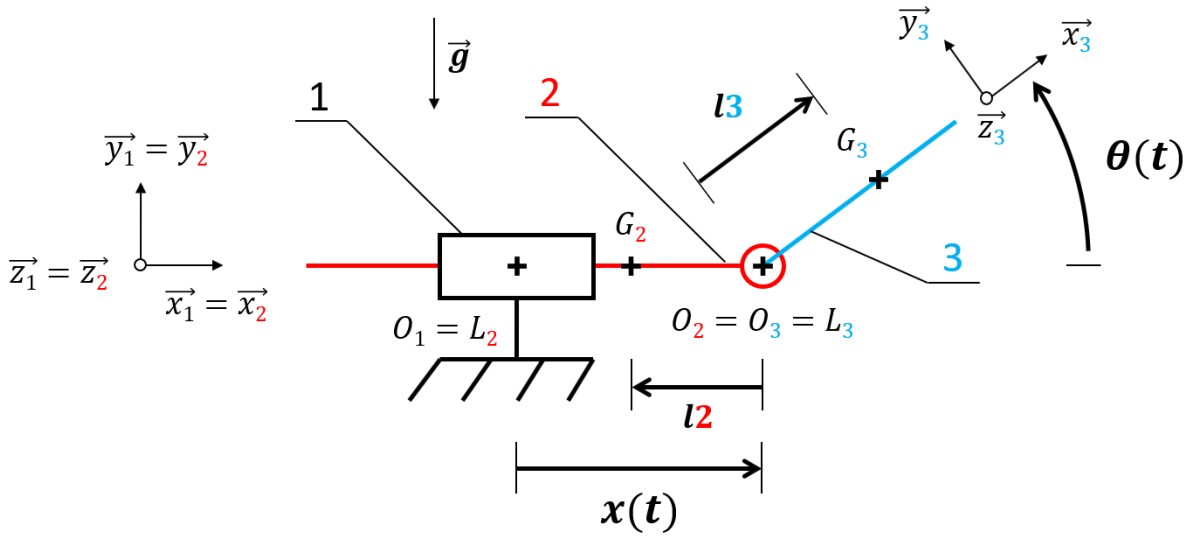


Figure 1. Schéma cinématique d'un mécanisme simple

Le fichier « MS_Description_arborescence.txt », disponible en annexe, traduit ce mécanisme pour la MDA. Les fichiers de description utilisés pour le gymnaste à la barre et pour ORHRO sont également fournis en annexe pour illustrer des exemples plus complexes.

a. Informations sur l'arborescence modélisée

La première chose à vérifier est que le fichier est compatible avec la version du code source que vous utiliserez.

```
# Ce fichier est compatible avec la version 1.85 de la MDA.
```

Indiquez ensuite les éléments qui vous permettront d'identifier l'arborescence modélisée par les fichiers obtenus, ainsi que sa version.

```
# Nommez l'arborescence :

Arborescence[Nom] := "":

# Donnez le sigle du nom de l'arborescence :

Arborescence[Sigle] := "":

# Décrivez l'arborescence :

# NULL

# Indiquez la version de l'arborescence :

Arborescence[Version] := "0":
# Changelog : NULL
```

Comme mentionné précédemment, il est possible d'initialiser des paramètres utilisés par le fichier de description. La méthode est décrite ci-après :

```
# L'arborescence peut-elle modéliser plusieurs individus ?

# Si non, renseignez autant de valeurs de paramètres que possible lorsque cela vous
sera demandé afin d'accélérer la création du modèle et de minimiser le nombre de
paramètres demandés lors de son utilisation.

# Si oui, deux options s'offrent à vous :

# 1) Vous souhaitez obtenir un modèle unique que vous personnaliserez lors de son
utilisation -> vous êtes libre de renseigner ou non les valeurs des paramètres. Tous
ceux qui n'auront pas été initialisés et qui sont nécessaires à l'évaluation du
modèle seront générés automatiquement et demandés lorsque vous utiliserez le modèle.

# 2) Vous souhaitez obtenir le modèle d'un individu en particulier -> renseignez des
noms personnalisés pour les paramètres ("PU_[...]", plus de détails sont donnés dans
la section dédiée aux paramètres fournis par l'utilisateur) puis créez dans le
dossier de la MDA un fichier initialisant ces noms avec les valeurs numériques
correspondant à l'individu concerné. Supprimez ci-dessous le symbole "#" devant
"Arborescence[Individu]" et recopiez le nom (avec son extension) de ce fichier entre
les guillemets. Vous pourrez ainsi répéter ce processus pour tous les individus dont
vous souhaitez le modèle personnalisé.

Arborescence[Individu] := "V1p85_VIERGE_Parametres_utilisateur.txt":

# Vous pouvez renseigner une chaîne de caractères qui sera intégrée aux noms des
fichiers générés :

Arborescence[Prefixe] := "":
```

b. Formes du modèle

L'utilisateur doit ensuite indiquer la ou les formes de modèle que la MDA doit générer. Plusieurs formes peuvent être générées simultanément. Comme ce sera le cas à de nombreuses reprises par la suite, le chiffre 1 active la fonctionnalité et le chiffre 0 la désactive.

```
# Sous quelle(s) forme(s) souhaitez-vous obtenir le modèle dynamique ?

# Il existe deux formes principales :

# - Le modèle dynamique inverse calcule les actions mécaniques de liaison à fournir
pour obtenir un mouvement donné de l'arborescence.

# - Le modèle dynamique direct calcule le mouvement obtenu lorsque des actions
mécaniques données sont fournies aux liaisons de l'arborescence.

# Il existe quatre formes secondaires pour le modèle dynamique inverse complet et il
est également possible de n'en obtenir qu'une partie en spécifiant les actions
mécaniques de liaison à calculer.

# Pour le modèle dynamique direct, il faut spécifier les paramètres de position et
d'orientation à calculer et la méthode numérique de résolution.

# Enfin, une forme combinée regroupant les modèles dynamiques partiels spécifiés
direct puis inverse est disponible pour des raisons de praticité.
```

Le modèle inverse complet est intéressant pour les problèmes de dimensionnement car il permet d'obtenir simplement toutes les expressions des éléments de réduction des torseurs des actions mécaniques de liaison pour toutes les liaisons de l'arborescence.

```
# Renseignez le chiffre 1 pour chaque forme souhaitée, le chiffre 0 sinon :

# 1) Les actions mécaniques à calculer sont celles exercées sur le solide enfant par
le solide parent (exprimées dans le repère du solide parent).

ObtentionMD[Inverse][Complet][EPRP][CE][Souhait] := 0:

# 2) Les actions mécaniques à calculer sont celles exercées sur le solide enfant par
le solide parent (exprimées dans le repère du solide enfant).

ObtentionMD[Inverse][Complet][EPRE][CE][Souhait] := 0:

# 3) Les actions mécaniques à calculer sont celles exercées sur le solide parent par
le solide enfant (exprimées dans le repère du solide enfant).

ObtentionMD[Inverse][Complet][PERE][CE][Souhait] := 0:

# 4) Les actions mécaniques à calculer sont celles exercées sur le solide parent par
le solide enfant (exprimées dans le repère du solide parent).

ObtentionMD[Inverse][Complet][PERP][CE][Souhait] := 0:
```

Le modèle inverse partiel est un des modèles de base, utilisé par exemple pour calculer uniquement les couples à fournir aux liaisons d'un robot 6 axes afin de réaliser un mouvement donné.

```
# Renseignez le chiffre 1 pour obtenir, entre autres, une fonction ne calculant que
les actions mécaniques spécifiées plus bas, le chiffre 0 sinon :

ObtentionMD[Inverse][Partiel][Specifie][CE][Souhait] := 0:
```

Le modèle direct n'existe qu'en version partielle pour le moment. Il détermine l'expression des dérivées d'ordre supérieur ou égal à deux des paramètres de positionnement et d'orientation demandés par l'utilisateur à l'instant courant (afin de laisser la possibilité à l'utilisateur de coder lui-même son schéma d'intégration pour l'instant d'après)…

```
# Renseignez le chiffre 1 pour obtenir, entre autres, une fonction ne calculant que
les dérivées d'ordre deux et plus des paramètres de position et d'orientation
spécifiés plus bas à l'instant courant, le chiffre 0 sinon :

ObtentionMD[Direct][Partiel][Specifie][CE][Souhait] := 0:

# Définition de l'ordre maximal des données de sorties.
# Renseignez le chiffre :
# - 2 pour obtenir l'accélération,
# - 3 pour obtenir l'accélération et l'à-coup,
# - 4 pour obtenir l'accélération, l'à-coup et de la dérivée de l'à-coup.

ObtentionMD[Direct][CE][Ordre] := 2:
```

…ou l'expression de ces paramètres à l'instant d'après par le biais de méthodes numériques usuelles présentées page 46.

```
# Renseignez le chiffre 1 pour obtenir, entre autres, une ou des fonctions ne
calculant que les paramètres (ainsi que leurs dérivées) de position et d'orientation
spécifiés plus bas à l'instant suivant, le chiffre 0 sinon :

ObtentionMD[Direct][Partiel][Specifie][CM][Souhait] := 0:

# Définition de l'ordre maximal des données de sorties.
# Renseignez le chiffre :
# - 1 pour obtenir la position/orientation et la vitesse,
# - 2 pour obtenir la position/orientation, la vitesse et l'accélération
# - 3 pour obtenir la position/orientation, la vitesse, l'accélération et l'à-coup.

ObtentionMD[Direct][CM][Ordre] := 1:

# Quelle(s) méthode(s) de calcul des paramètres de position et d'orientation (et
potentiellement de leurs dérivées) à l'instant d'après souhaitez-vous utiliser ?

# Renseignez le chiffre 1 pour chaque méthode souhaitée, le chiffre 0 sinon :

# Méthode d'Euler :

ObtentionMD[Direct][CME][Souhait] := 1:

# Méthode de Runge-Kutta d'ordre 4 :

ObtentionMD[Direct][CMRK4][Souhait] := 0:

# Méthode de Jean-Baptiste :

ObtentionMD[Direct][CMJB][Souhait] := 0:
```

Le modèle combiné est un modèle hybride qui calcule le modèle inverse partiel à l'instant d'après avec les paramètres de positionnement et d'orientation du modèle direct partiel.

```
# Renseignez le chiffre 1 pour obtenir, entre autres, une fonction combinant les
modèles dynamiques partiels spécifiés direct et inverse, le chiffre 0 sinon :

ObtentionMD[Combine][Partiel][Specifie][CM][Souhait] := 0:
```

Le modèle avancé est la forme qui permet d'obtenir l'expression de n'importe quelles grandeurs utilisées par la MDA. Des exemples de son utilisation seront donnés lors de l'explication du code source.

```
# Cette forme est réservée aux utilisateurs avancés de la MDA. L'activer permet de
demander n'importe quelle(s) grandeur(s) impliquée(s) dans la dynamique de
l'arborescence modélisée.

ObtentionMD[Avance][Partiel][Specifie][CE][Souhait] := 0:
```

c. Traitement des équations

L'activation des paramètres de cette partie influence sur les temps de génération des modèles et le nombre d'opérations contenues dans les fichiers.

Une partie des opérations de la MDA sont parallélisées. Il est possible de définir le nombre de processeurs logiques auxquels attribuer ces opérations.

```
# Combien de processeurs logiques souhaitez-vous allouer à la génération du modèle
dynamique ? Attention, un nombre élevé peut entraîner une demande importante de
mémoire vive.

NombreProcesseursLogiques := 1:
```

Il est possible, pour les fichiers de calcul, de choisir le nombre de chiffres significatifs avec lesquels ces calculs seront effectués. Il est conseillé de ne sélectionner le mode « simple » que s'il s'agit d'une exigence de la plateforme sur laquelle le fichier sera lu.

```
# Avec quelle précision les nombres flottants doivent-il être évalués ?

# Renseignez :
# - "simple" pour calculer avec 8 chiffres significatifs
# - "double" pour calculer avec 16 chiffres significatifs

Precision[Valeur] := "double":
```

En manipulant les expressions, la MDA fait apparaître des identités mathématiques remarquables qui peuvent être simplifiées. La recherche de ces identités demande des ressources matérielles supplémentaires et peut rallonger le temps de génération des modèles. Augmenter le niveau augmente le nombre d'expressions simplifiées par la MDA ; le premier niveau suffit néanmoins pour la grande majorité des cas. La simplification par défaut ne concerne que les relations trigonométriques tandis que la simplification complète cherche également au niveau des puissances, des logarithmes et des racines.

```
# ATTENTION ! Les options suivantes ont pour but de diminuer le nombre de calculs
que contiennent les fichiers générés par la MDA, améliorant ainsi leur rapidité
d'exécution. Leurs activations peuvent ralentir considérablement la génération des
modèles dynamiques.

# Souhaitez-vous que les expressions des éléments demandés soient simplifiées ?

# Renseignez le chiffre :
# - 0 pour ne rien simplifier
# - 1 pour simplifier les équations de mouvement (recommandé)
# - 2 pour simplifier les équations de mouvement, les expressions des positions et
des vitesses
# - 3 pour simplifier les équations de mouvement, les expressions des positions, des
vitesses et des accélérations
# - 4 pour simplifier les équations de mouvement, les expressions des positions, des
vitesses, des accélérations et des à-coups
# - 5 pour simplifier les équations de mouvement, les expressions des positions, des
vitesses, des accélérations, des à-coups, des dérivées des à-coups et des
composantes calculées par le modèle combiné.

Simplification := 0:

# Souhaitez-vous utiliser la simplification complète ?
# Renseignez le chiffre 1 le cas échéant, le chiffre 0 sinon (recommandé) :

SimplificationComplete := 0:

# L'arborescence modélisée a-t-elle beaucoup de paramètres inconnus ?
# Renseignez le chiffre 1 le cas échéant pour limiter la quantité maximale de
mémoire vive utilisée (recommandé), le chiffre 0 sinon :

SimplificationProgressive := 0:
```

Factoriser les expressions peut également réduire le nombre d'opérations.

```
# Souhaitez-vous que les expressions des éléments demandés soient factorisées ?
# Renseignez le chiffre 1 le cas échéant, le chiffre 0 sinon :

Factorisation := 0:
```

Pour éviter les problématiques liés aux nombres flottants proches de zéro, un seuil peut être défini sous lequel ces nombres seront imposés à zéro. Cette option n'est pas nécessaire dans la grande majorité des cas.

```
# Souhaitez-vous approximer les expressions des éléments demandés en annulant les
termes dont les coefficients numériques sont proches de zéro ?
# Le cas échéant, renseignez la valeur positive en dessous de laquelle les
coefficients doivent être annulés ; renseignez le chiffre 0 sinon (recommandé) :

ValeurLimiteAnnulation := 0:
```

Enfin, option presque obligatoire pour les fichiers de calcul et sans effet pour les fichiers formels, une procédure d'optimisation formelle peut être activée. Elle utilise la procédure « optimize » de la bibliothèque « codegen » développée par Maplesoft™ pour son logiciel Maple [46]. Son objectif est de repérer des opérations qui reviennent plusieurs fois au sein des expressions demandées, de les réaliser en amont et de les remplacer par le résultat pour toutes les occurrences.

```
# Souhaitez-vous que le calcul des éléments demandés soit optimisé en nombre
d'opérations ?
# Renseignez le chiffre 1 le cas échéant (fort impact mais recommandé), le chiffre 0
sinon :

Optimisation := 0:
```

d. Génération des fichiers

C'est ici que le format des fichiers est défini. Il est possible d'activer tous les paramètres simultanément. Attention néanmoins, les expressions formelles générées par les modèles dynamiques de mécanismes complexes peuvent être très, très longues…

```
# Quel(s) type(s) de fichier(s) souhaitez-vous générer ?

# Il existe trois possibilités, renseignez le chiffre 1 pour chaque type souhaité,
le chiffre 0 sinon :

# 1) Fichier LaTeX contenant les expressions formelles des éléments à calculer.

Fichier[Formel][LaTeX][Souhait] := 0:

# 2) Fichier(s) de calcul écrits en C++ destiné(s) à être utilisé(s) par un
automate.

Fichier[Calcul][Cpp][Automate][Souhait] := 0:

# 3) Fichier(s) de calcul écrits en C++ destiné(s) à être transformé(s) en MEX
Function via MATLAB.

Fichier[Calcul][Cpp][MATLAB][Souhait] := 0:
```

e. Éléments à calculer

Presque tous les paramètres précédents indiquaient à la MDA comment calculer. Voyons à présent ceux définissant quoi calculer. À noter s'il s'agit de la première description d'une arborescence :

```
# Afin de pouvoir remplir cette section facilement, il est conseillé de compléter
les sections suivantes avant. Elle est placée ici afin de pouvoir modifier
rapidement les éléments à calculer.
```

Si vous souhaitez générer un modèle dynamique inverse pour lequel vous spécifiez les éléments à calculer, vous devez remplir la liste des éléments à calculer pour le modèle inverse partiel spécifié.

```
# Remplissez la liste des composantes des éléments de réduction des torseurs des
actions mécaniques de liaison à calculer.

# Règles de saisie :

# - La première lettre désigne le nom de la composante (X, Y, Z et L, M, N pour les
composantes de la résultante et du moment respectivement selon les axes x, y, z).

# - Le premier indice indique le nom du torseur (ici TAML pour torseur de l'action
mécanique de liaison).

# - Le deuxième indice indique le solide sur lequel l'action mécanique est exercée.

# - Le troisième indice indique le solide exerçant l'action mécanique.

# - Pour les moments, l'indice suivant indique le point de réduction du torseur (ici
le point liant les deux solides).

# - Le dernier indice indique le repère dans lequel la composante est exprimée (ici
soit le repère du solide sur lequel l'action mécanique est exercée, soit le repère
du solide qui exerce l'action mécanique, soit le repère du milieu extérieur).

# Exemple de saisie :
# LEAC[Inverse][Partiel][Specifie][CE] := [X[TAML][S[1]][S[2]][R[1]],
X[TAML][S[2]][S[1]][R[1]], L[TAML][S[1]][S[2]][L[2]][R[2]],
M[TAML][S[2]][S[1]][L[2]][R[IntermediaireEntre1Et2]],
N[TAML][S[2]][S[1]][L[2]][R[0]]]:

LEAC[Inverse][Partiel][Specifie][CE] := []:

# Vous pouvez renseigner une chaîne de caractères qui sera intégrée au nom de la
fonction générée :

Fichier[SuffixeAdditionnel][Inverse][Partiel][Specifie] := "":
```

Dans le cas de l'exemple simple, demander les expressions symboliques des efforts d'actionnement du mécanisme générera les équations de la forme suivante (issues du fichier LaTeX) :

$$X(t) = -l3m3\left(\frac{d}{dt}\theta(t)\right)^2\cos(\theta(t)) - l3m3\sin(\theta(t))\left(\frac{d^2}{dt^2}\theta(t)\right) + m2\left(\frac{d^2}{dt^2}x(t)\right) + m3\left(\frac{d^2}{dt^2}x(t)\right)$$

$$N(t) = l3^2m3\left(\frac{d^2}{dt^2}\theta(t)\right) + l3\cos(\theta(t))\,m3g - \left(\frac{d^2}{dt^2}x(t)\right)\sin(\theta(t))\,l3m3 + Izz3\left(\frac{d^2}{dt^2}\theta(t)\right)$$

Équation 1

où $X(t)$ et $N(t)$ sont respectivement la force et le couple actionnant les liaisons glissière et pivot.

Si vous souhaitez générer un modèle dynamique direct, vous devez toujours remplir la liste LPPO[Direct][Partiel][Specifie][ND].

```
# Remplissez la liste des paramètres de position et d'orientation à calculer.
# Exemple de saisie :
# LPPO[Direct][Partiel][Specifie][ND] := [x[O[1]][O[2]][R[1]],phi[2,3]]:

LPPO[Direct][Partiel][Specifie][ND] := []:
```

Dans le cas de l'exemple simple, demander les expressions symboliques des dérivées secondes des paramètres de positionnement et d'orientation actionnés réalisera les opérations suivantes :

1. factorisation matricielle des expressions de $X(t)$ et de $N(t)$ par $\frac{d^2}{dt^2}x(t)$ et par $\frac{d^2}{dt^2}\theta(t)$ qui donnera une équation de la forme

$$A \cdot \begin{pmatrix} \frac{d^2}{dt^2}x(t) \\ \frac{d^2}{dt^2}\theta(t) \end{pmatrix} = B$$

Équation 2

2. inversion de la matrice A pour isoler $\frac{d^2}{dt^2}x(t)$ et $\frac{d^2}{dt^2}\theta(t)$

$$\begin{pmatrix} \frac{d^2}{dt^2}x(t) \\ \frac{d^2}{dt^2}\theta(t) \end{pmatrix} = A^{-1} \cdot B$$

Équation 3

3. produit matriciel de A^{-1} et de B pour obtenir les expressions (issues du fichier LaTeX)

$$\frac{d^2}{dt^2}x(t) = \frac{X(t)\,Izz3 + l3^2 m3 X(t) + \left(\left(\frac{d}{dt}\theta(t)\right)^2 l3^3 m3 + \left(\frac{d}{dt}\theta(t)\right)^2 Izz3 l3\right) m3\cos(\theta(t)) + \left(-g\,l3^2 m3^2\cos(\theta(t)) + N(t)\,l3m3\right)\sin(\theta(t))}{l3^2 m3^2\left(\cos^2(\theta(t))\right) + \left(l3^2 m2 + Izz3\right) m3 + Izz3m2}$$

$$\frac{d^2}{dt^2}\theta(t) = \frac{N(t)\,m2 + m3N(t) + (-m2gl3 - gl3m3)\,m3\cos(\theta(t)) + \left(\left(\frac{d}{dt}\theta(t)\right)^2 l3^2 m3^2\cos(\theta(t)) + X(t)\,l3m3\right)\sin(\theta(t))}{l3^2 m3^2\left(\cos^2(\theta(t))\right) + \left(l3^2 m2 + Izz3\right) m3 + Izz3m2}$$

Équation 4

Il y a trois paramètres supplémentaires pour le modèle direct :

1. L'inverse d'une matrice étant la transposée de sa comatrice divisée par son déterminant, il est important de vérifier que ce dernier ne s'annule pas. Activer le premier paramètre supplémentaire ajoute le calcul du déterminant de la matrice A à ceux des paramètres de positionnement et d'orientation (PPO) demandés afin d'en connaître sa valeur.

2. La désactivation du deuxième permet de modéliser un système en régime libre sans avoir à indiquer que les actions mécaniques liées aux PPO calculés sont nulles. Dans l'exemple simple, la MDA écrit que $X(t) = 0$ et que $N(t) = 0$.

3. Le troisième vous permet de définir le pas temporel auquel l'évolution des PPO est calculée par les méthodes numériques. Ne lui donnez pas de valeur numérique si vous souhaitez pouvoir le choisir lors de l'invocation du modèle.

```
# Souhaitez-vous calculer le déterminant de la matrice des coefficients multipliant
les dérivées secondes des paramètres de position et d'orientation à calculer ?
# Renseignez le chiffre 1 le cas échéant, le chiffre 0 sinon :

CalculDeterminant := 0:

# Des actions mécaniques de liaison associées aux paramètres de position et
d'orientation à calculer peuvent-elles être exercées ?
# Renseignez le chiffre 1 le cas échéant, le chiffre 0 sinon :

LiaisonsPilotees := 0:

# Renseignez la valeur numérique strictement positive du pas temporel (c'est-à-dire
le temps écoulé entre deux évaluations du modèle dynamique direct, en secondes).
# Si vous souhaitez pouvoir choisir cette valeur lors de l'utilisation du modèle,
laissez le symbole "#" devant la ligne ci-dessous.

#h := 0.001:

# Vous pouvez renseigner une chaîne de caractères qui sera intégrée au nom de la
fonction générée :

Fichier[SuffixeAdditionnel][Direct][Partiel][Specifie] := "":
```

f. Description de la morphologie du système matériel

Élément crucial du fichier de description, la liste décrivant l'arborescence permet à la MDA de comprendre comment les solides constituant l'arborescence modélisée sont liés entre eux. Il existe toujours plusieurs façons de renseigner cette liste, qui dépendent directement de l'arborescence choisie pour décrire le mécanisme. Le choix minimisant la longueur des branches et plaçant l'un des solides liés à l'extérieur à la racine est souvent le choix minimisant également la longueur des expressions obtenues grâce à la MDA.

```
# Liste décrivant l'arborescence (LDA)

# Règles de saisie :

# - Le chiffre 0 représente le milieu extérieur. Il ne peut pas être utilisé comme
numéro identifiant un solide.

# - Chaque branche doit être composée de nœuds dont les clés correspondent aux
numéros des solides, classés par ordre de généalogie descendante.

# - Les branches doivent être renseignées par ordre de profondeur croissante de
leurs racines par rapport à la racine de l'arborescence (qui est le premier nœud de
la première branche renseignée).

# - Chaque branche doit contenir au moins deux nœuds dont le premier est une racine.

# - Chaque nœud ne peut avoir qu'un parent.

# - Le dernier nœud de la dernière branche doit être une feuille.

# - Toutes les arêtes (i.e. liaisons entre solides) doivent être renseignées une
unique fois dans la liste. Cela signifie entre autres que la racine d'une branche
est également un des nœuds d'une branche précédente.

# Remplissez la liste de la manière suivante :

# LDA := [
# [NuméroSolideNoeud1,NuméroSolideNoeud2,...,NuméroSolideNoeudN,...],
# [NuméroSolideNoeudN,NuméroSolideNoeudN+1,...]
# ]:

LDA := [[1]]:
```

g. Paramètres fournis par l'utilisateur

Les utilisateurs peuvent définir le nom des paramètres de la façon qu'ils le souhaitent. Il y a néanmoins quelques règles à suivre afin de ne pas écraser d'autres paramètres et assurer le bon fonctionnement de la MDA.

```
# La MDA vous permet d'utiliser des paramètres personnalisés pour faciliter la
transcription de vos mécanismes.

# Les noms de ces paramètres doivent satisfaire trois conditions :

# - Ils doivent commencer par "PU_".

# - Ils doivent finir par "(t)" si le paramètre dépend du temps.

# - Ils doivent tous être renseignés dans la liste ci-dessous.

# Vous pourrez ensuite utiliser ces paramètres dans le reste de ce fichier.
# Exemple de nom de paramètre : PU_RaideurRessort(t).

LPU := []:
```

h. Paramètres inertiels et géométriques

Les premiers paramètres demandés par la MDA concernent les propriétés inertielles et géométriques des solides constituant l'arborescence. Ce sont des paramètres importants qui dans la plupart des cas n'évoluent pas au cours du temps. Néanmoins cela n'est pas une exigence : il est possible de les renseigner comme paramètres utilisateurs dépendant du temps.

```
# Si l'un des paramètres suivants n'est pas initialisé ici, il sera considéré comme
une constante indéterminée. Elle agira alors comme une donnée d'entrée et sera
demandée lors de l'exécution des fichiers de calcul.

# m[S[i]], masse (en kilogrammes) du solide S[i] :

m[S[1]] := 0:

# MI[S[i]][G[i]][R[i]], matrice d'inertie (en kg.m²) de S[i] au point G[i] (centre
d'inertie de S[i]) (exprimée dans le repère R[i] associé à ce solide) :
# Si la valeur numérique (ou l'expression en fonction d'autres paramètres) d'au
moins un élément de la matrice est inconnue lors du complètement du présent fichier,
renseignez les moments et produits d'inertie dont les valeurs numériques ou les
expressions sont connues avec Ixx, Iyy, Izz, Iyz, Ixz et Ixy à la place de MI.
Attention, les valeurs des produits d'inertie renseignées ici seront multipliées par
-1 lors de la création de la matrice d'inertie par la MDA.

MI[S[1]][G[1]][R[1]] :=
Matrix(3,{(1,1)=0,(2,2)=0,(3,3)=0,(2,3)=0,(1,3)=0,(1,2)=0},shape=symmetric):

# Dep[O[i]][G[i]][R[i]], vecteur déplacement (en mètres) du point O[i] (origine de
R[i]) vers G[i] (exprimé dans R[i]) :
# Si la valeur numérique (ou l'expression en fonction d'autres paramètres) d'au
moins une composante du vecteur est inconnue lors du complètement du présent
fichier, renseignez celles dont les valeurs numériques ou les expressions sont
connues avec x, y et z à la place de Dep.

Dep[O[1]][G[1]][R[1]] := Vector([0,0,0]):

# Dep[O[i]][L[i]][R[i]], vecteur déplacement (en mètres) de O[i] vers le point L[i]
(liaison entre S[i] et son parent) (exprimé dans R[i]) :
# Si la valeur numérique (ou l'expression en fonction d'autres paramètres) d'au
moins une composante du vecteur est inconnue lors du complètement du présent
fichier, renseignez celles dont les valeurs numériques ou les expressions sont
connues avec x, y et z à la place de Dep.

Dep[O[1]][L[1]][R[1]] := Vector([0,0,0]):
```

i. Paramètres de position et d'orientation

Les deuxièmes paramètres demandés peuvent être renseignés de plusieurs façons. Dans la version 1.85 de la MDA, le positionnement et l'orientation des solides les uns par rapports aux autres peut se faire via les angles d'Euler-Tait-Bryan ou la méthode de Denavit-Hartenberg modifiée par Khalil et Kleinfinger.

```
# Comment les solides sont-ils positionnés et orientés par rapport au référentiel terrestre R[0] (considéré galiléen) ainsi que par rapport au référentiel de leur parent R[Parent] ?

# Règles de lecture :

# - Le premier indice indique par rapport à quel référentiel le solide sera positionné et orienté (R[0] ou R[Parent]).

# - Le deuxième indice indique de quelle manière (ETB ou DH).

# - ETB signifie que le solide sera positionné et orienté grâce aux angles d'Euler ou de Tait-Bryan.

# - DH signifie que le solide sera positionné et orienté grâce à la méthode de description de morphologie de Denavit-Hartenberg modifiée par Khalil et Kleinfinger.

# - Au moins un solide doit être positionné et orienté par rapport à R[0], d'une manière ou d'une autre.

# - Tous les solides, exceptée la racine de l'arborescence, doivent être positionnés et orientés par rapport à R[Parent], d'une manière ou d'une autre. Si le positionnement et l'orientation d'un solide par rapport à R[Parent] sont inconnus, ils peuvent être déterminés automatiquement si le parent du solide concerné (ou l'un de ses ascendants) et lui-même (ou l'un de ses descendants) sont positionnés et orientés par rapport à R[0].

# Remplissez les ensembles suivants avec les numéros des solides séparés par une virgule :

ENSAPO[R[0]][ETB] := {}:
ENSAPO[R[0]][DH] := {}:
ENSAPO[R[Parent]][ETB] := {}:
ENSAPO[R[Parent]][DH] := {}:
```

Il est nécessaire qu'au moins un des solides de l'arborescence soit positionné et orienté par rapport au milieu extérieur, quelle que soit la façon.

```
# Si l'un des paramètres suivants n'est pas initialisé ici, il sera considéré comme une variable temporelle (excepté pour a et alpha qui seront considérés comme des constantes indéterminées). Elle agira alors comme une donnée d'entrée et sera demandée lors de l'exécution des fichiers de calcul de dynamique inverse (et de dynamique direct si elle est une variable temporelle pilotée).
```

```
# Éléments des matrices de transformation de R[0] vers R[i] et inversement
```

```
# Angles d'Euler ou de Tait-Bryan

# Par défaut, les rotations se font d'abord autour de l'axe x du repère R[0], puis
autour de l'axe y du repère R[IntermediaireEntre0Eti] et enfin autour de l'axe z du
repère R[i].
# Vous pouvez spécifier un autre ordre des rotations parmi les douze existants.
# Exemple avec l'ordre par défaut :
# OrdreDesRotations[0,i] := [x,y,z]:

# beta[0,i], première rotation (en radians) autour du premier axe de
OrdreDesRotations[0,i] :

# rho[0,i], deuxième rotation (en radians) autour du deuxième axe de
OrdreDesRotations[0,i] :

# phi[0,i], troisième rotation (en radians) autour du troisème axe de
OrdreDesRotations[0,i] :

# Dep[O[0]][O[i]][R[0oui]], vecteur déplacement (en mètres) de O[0] vers O[i]
(exprimé dans R[0] ou dans R[i]) :
# Si la valeur numérique (ou l'expression en fonction d'autres paramètres) d'au
moins une composante du vecteur est inconnue lors du complètement du présent
fichier, renseignez celles dont les valeurs numériques ou les expressions sont
connues avec x, y et z à la place de Dep.
```

```
# Paramètres de la méthode de description de morphologie de Denavit-Hartenberg
modifiée par Khalil et Kleinfinger

# a[0,i], déplacement (en mètres) de z[0] vers z[i] le long de x[0] :

# alpha[0,i], rotation (en radians) de z[0] vers z[i] autour de x[0] :

# r[0,i], déplacement (en mètres) de x[0] vers x[i] le long de z[i] :

# theta[0,i], rotation (en radians) de x[0] vers x[i] autour de z[i] :
```

Les solides sont positionnés et orientés par rapport à leurs parents respectifs de manière similaire.

```
# Éléments des matrices de transformation de R[i-1] vers R[i] et inversement

# ATTENTION ! "i-1" a une signification différente en fonction des méthodes de
positionnement et d'orientation.

# ETB : "i-1" est seulement une façon simple d'indiquer qu'il s'agit du parent du
solide S[i]. Lorsque vous renseignez les paramètres ci-dessous, vous devez remplacer
"i-1" par le véritable numéro du parent dans la LDA (qui ne vaut pas nécessairement
i-1).

# DH : "i-1" est le numéro du parent du solide S[i] lors de la création initiale des
repères. Cela permet de déclarer les paramètres (a, alpha, r, theta) de la même
façon peu importe le sens de parcours de la branche dans la LDA. Il est alors
nécessaire de bien respecter la méthode et de s'assurer que "i-1", qui ne vaut pas
nécessairement i-1, est néanmoins inférieur à "i".
```

```
# Angles d'Euler ou de Tait-Bryan

# Par défaut, les rotations se font d'abord autour de l'axe x du repère R[i-1], puis
autour de l'axe y du repère R[IntermediaireEntrei-1Eti] et enfin autour de l'axe z
du repère R[i].
# Vous pouvez spécifier un autre ordre des rotations parmi les douze existants.
# Exemple avec l'ordre par défaut :
# OrdreDesRotations[i-1,i] := [x,y,z]:

# beta[i-1,i], rotation (en radians) autour du premier axe de OrdreDesRotations[i-
1,i] :

# rho[i-1,i], rotation (en radians) autour du deuxième axe de OrdreDesRotations[i-
1,i] :

# phi[i-1,i], rotation (en radians) autour du troisème axe de OrdreDesRotations[i-
1,i] :

# Dep[O[i-1]][O[i]][R[i-1oui]], vecteur déplacement (en mètres) de O[i-1] vers O[i]
(exprimé dans R[i-1] ou dans R[i]) :
# Il est possible de renseigner Dep[O[i-1]][L[i]][R[i-1]] à la place.
# Si la valeur numérique (ou l'expression en fonction d'autres paramètres) d'au
moins une composante du vecteur est inconnue lors du complètement du présent
fichier, renseignez celles dont les valeurs numériques ou les expressions sont
connues avec x, y et z à la place de Dep.
```

```
# Paramètres de la méthode de description de morphologie de Denavit-Hartenberg
modifiée par Khalil et Kleinfinger

# a[i-1,i], déplacement (en mètres) de z[i-1] vers z[i] le long de x[i-1] :

# alpha[i-1,i], rotation (en radians) de z[i-1] vers z[i] autour de x[i-1] :

# r[i-1,i], déplacement (en mètres) de x[i-1] vers x[i] le long de z[i] :

# theta[i-1,i], rotation (en radians) de x[i-1] vers x[i] autour de z[i] :
```

j. Paramètres des actions mécaniques extérieures ponctuelles

Les troisièmes et derniers paramètres demandés concernent les interactions mécaniques entre l'arborescence et le milieu extérieur. La première est à distance, il s'agit de la pesanteur.

```
# L'action de la pesanteur est-elle prise en compte ?

# Si oui, renseignez la valeur de l'accélération de la pesanteur « g » (en m/s²) et
remplissez le vecteur d'orientation de la pesanteur « VO[Pesanteur] », exprimé dans
R[0], de façon que sa direction ainsi que son sens soient similaires à ceux du champ
de pesanteur au voisinage de l'arborescence modélisée.

g := 9.80665:

VO[Pesanteur][R[0]] := Vector([0,0,0]):

# Si non, renseignez la valeur 0 pour l'accélération de la pesanteur.
```

D’une manière similaire aux paramètres de position et d’orientation, il faut ensuite indiquer quels sont les numéros des solides concernés par les actions mécaniques extérieures ponctuelles.

```
# Y a-t-il des solides soumis à des actions mécaniques extérieures ponctuelles
(hormis celle équivalente à l'action de la pesanteur) ?

# Règle de saisie :

# - Le deuxième indice est S[i] en remplaçant i par le numéro du solide sur lequel
ces actions mécaniques sont exercées.

# - Le troisième indice est :
#       - R[i] si les actions mécaniques sont (ou doivent être) exprimées dans le
repère du solide,
#       - R[j] si les actions mécaniques sont (ou doivent être) exprimées dans le
repère d'un autre solide de l'arborescence (en remplaçant j par le numéro de cet
autre solide),
#       - R[0] sinon.

# - Le second membre est un ensemble contenant les numéros identifiant les actions
mécaniques répondant aux trois critères ci-dessus.

# - Deux actions mécaniques différentes, exercées sur le même solide, ne peuvent
être identifiées par le même numéro.

# - Deux actions mécaniques différentes, exercées sur des solides différents,
peuvent être identifiées par le même numéro.

# - Il doit y avoir autant de numéros renseignés (tous ensembles regroupés) que
d'actions mécaniques extérieures ponctuelles appliquées au système matériel.

# Remplissez le tableau de la manière suivante :
# ENAMEP[C][S[i]][R[ioujou0]] :=
{NuméroPremièreActionMécanique,NuméroDeuxièmeActionMécanique,...}:
```

Il est possible de renseigner les expressions de la position d'un point de contact ou des composantes des éléments de réduction du torseur de l'action mécanique si ceux-ci sont connues.

```
# Si l'un des paramètres suivants n'est pas initialisé ici, il sera considéré comme
une variable temporelle. Elle agira alors comme une donnée d'entrée et sera demandée
lors de l'exécution des fichiers de calcul.

# Dep[O[ioujou0]][C[i][n]][R[ioujou0]], vecteur déplacement (en mètres) de
O[ioujou0] vers le point C[i][n] qui est le point d'application sur S[i] de l'action
mécanique numéro n (exprimé dans R[ioujou0]) :
# Si la valeur numérique (ou l'expression en fonction d'autres paramètres) d'au
moins une composante du vecteur est inconnue lors du complètement du présent
fichier, renseignez celles dont les valeurs numériques ou les expressions sont
connues avec x, y et z à la place de Dep.

# Res[TAMPC][S[i]][Ext[n]][R[ioujou0]], résultante (en kg.m/s²) du torseur de
l'action mécanique ponctuelle connue exercée sur S[i] par l'extérieur en C[i][n]
(exprimé dans R[ioujou0]) :
# Si la valeur numérique (ou l'expression en fonction d'autres paramètres) d'au
moins une composante du vecteur est inconnue lors du complètement du présent
fichier, renseignez celles dont les valeurs numériques ou les expressions sont
connues avec X, Y et Z à la place de Res.

# Mom[TAMPC][S[i]][Ext[n]][C[i][n]][R[ioujou0]], moment (en kg.m²/s²) en C[i][n] du
torseur de l'action mécanique ponctuelle connue exercée sur S[i] par l'extérieur en
C[i][n] (exprimé dans R[ioujou0]) :
# Si la valeur numérique (ou l'expression en fonction d'autres paramètres) d'au
moins une composante du vecteur est inconnue lors du complètement du présent
fichier, renseignez celles dont les valeurs numériques ou les expressions sont
connues avec L, M et N à la place de Mom.
```

5. Initialiser des paramètres

Comme pour le fichier de description, passons en revue le fichier d'initialisation de paramètres. Des captures d'écran du fichier « V1p85_VIERGE_Parametres_utilisateur.txt » seront présentées ici à des fins d'illustration. Les fichiers d'initialisation utilisés pour ORHRO et le gymnaste à la barre sont fournis en annexe en guise d'exemple.

a. Informations sur l'individu

La première chose à vérifier est que le fichier est compatible avec la version du code source que vous utiliserez.

```
# Ce fichier est compatible avec la version 1.85 de la MDA.
```

Indiquez ensuite les éléments qui vous permettront d'identifier l'individu modélisé par les fichiers obtenus, ainsi que sa version.

```
# Nommez l'individu :

Individu[Nom] := "":

# Donnez le sigle du nom de l'individu :

Individu[Sigle] := "":

# Décrivez l'individu :

# NULL

# Indiquez la version de l'individu :

Individu[Version] := "":
# Changelog : NULL
```

b. Initialisation des paramètres fournis par l'utilisateur

En renseignant dans le fichier d'initialisation les valeurs des paramètres utilisateurs plutôt que dans le fichier de description, il est possible de créer plusieurs fichiers d'initialisation et de simplement changer le nom à lire dans le fichier de description.

```
# Exemple d'initialisation de paramètre :
#PU_RaideurRessort(t) := 0: # Raideur du ressort dépendante du temps (en kg/s² ou
N/m)
```

6. Exécuter le code source

Double-cliquez sur le fichier contenant le code source, « V1p85_Modelisation_Dynamique_Arborescence.mw » dans notre cas. En cliquant ensuite sur l'icône composée de trois points d'exclamation ou en tapant la combinaison de touches Ctrl+Maj+Entrée, Maple exécutera tous les groupes d'exécution du fichier. Une fenêtre s'ouvre, vous invitant à sélectionner le fichier de description que la MDA doit lire :

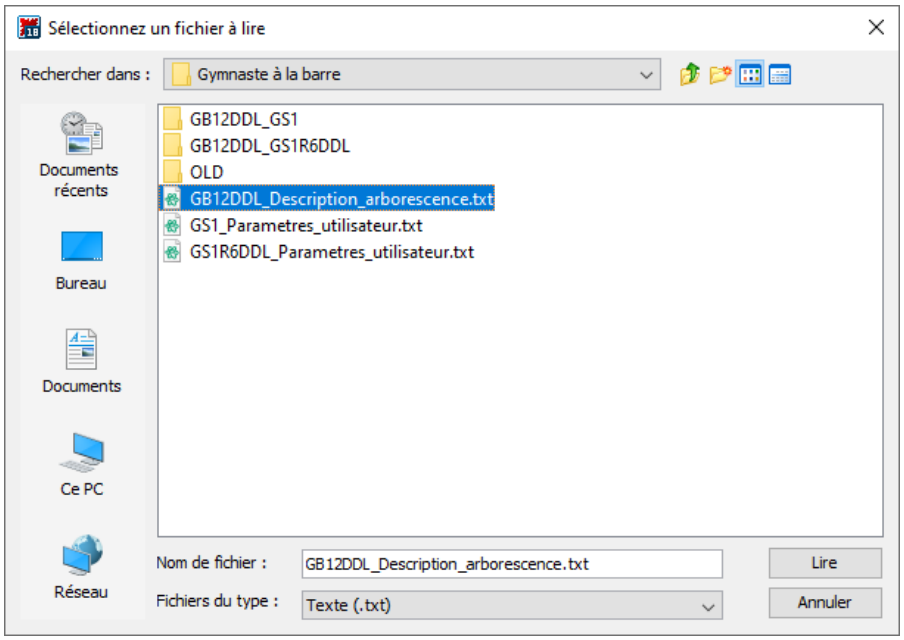


Figure 2. Capture d'écran de la fenêtre de sélection du fichier à lire

Une fois le dernier groupe exécuté, les fichiers contenant les modèles ont été écrits dans le dossier contenant le fichier de description lu, prêts à être utiliser.

7. Explication de l'architecture du code source

La MDA est constituée de trois grandes parties :

1. La génération du modèle dynamique.
2. La vérification optionnelle des données.
3. La génération des fichiers

a. Génération du modèle dynamique

La première partie a pour objectif de déterminer les expressions demandées par l'utilisateur.

1. Les **Étapes préliminaires diverses** préparent Maple en indiquant les bibliothèques de procédures à utiliser ou en générant les tableaux dans lesquels entreposer les données et paramètres renseignés par l'utilisateur par exemple.

2. L'**Analyse et la définition du souhait de l'utilisateur** a pour vocation de valider la saisie de l'utilisateur. Si certains paramètres ont mal été renseignés, la MDA les désactive et informe l'utilisateur. Cette partie nécessite encore d'être testée et enrichie, il est certainement encore possible de demander des choses impossibles ou contradictoires.

3. La **Création des procédures de traitement des expressions** définit les procédures :

    a. *Estimer*, pour connaître le coût calculatoire d'une expression.

    b. *Simplifier*, pour simplifier les expressions sur la base d'identité remarquable, pilotée par le niveau de Simplification.

    c. *Factoriser*, pour factoriser les expressions, pilotée par le niveau de Factorisation et par une liste de facteurs. Pour le moment, une seule liste a été définie sur la base d'expérimentation sur les paramètres à factoriser et l'ordre de factorisation. Une deuxième liste, générée sur la base des paramètres indéterminés et de leurs occurrences dans l'expression à factoriser, sera disponible à terme.

    d. *Approximer*, pour annuler les nombres flottants proches de zéro, pilotée par la ValeurLimiteAnnulation.

4. Le **Complètement et le rangement des paramètres** est une grosse étape durant laquelle la MDA repère les paramètres dont elle a besoin pour générer les expressions demandées par l'utilisateur, les crée si l'utilisateur ne les a pas renseignés puis les entrepose dans les tableaux correspondants.

5. La **Création des liaisons** est l'étape la plus importante de la MDA. C'est elle qui réalise tous les calculs mathématiques. Voici les sous-étapes qui la compose :

    a. Obtention des matrices de passage, des vecteurs déplacements et des vitesses de rotation des solides

    b. Création des listes de facteurs

    c. Obtention des données cinématiques, cinétiques et dynamiques

    d. Obtention des torseurs des actions mécaniques extérieures

    e. Obtention des éléments demandés pour le modèle avancé

    f. Obtention des torseurs des actions mécaniques de liaison

    g. Potentielle simplification des expressions qui seront utilisées

    h. Isolement des dérivées secondes par rapport au temps des paramètres de positionnement et d'orientation demandés entre les solides liés

    i. Obtention des expressions des paramètres de positionnement et d'orientation demandés entre les solides liés

6. Dernière étape de la première partie, la **Création des listes des paramètres à demander à l'utilisateur** identifie les paramètres apparaissant dans les expressions à calculer afin de n'inclure qu'eux dans les paramètres des fonctions générées.

b. Vérification optionnelle des données

La deuxième partie n'a aucune incidence sur le résultat de la modélisation. Les lignes de code qu'elle contient permettent de visualiser certains éléments clés à des fins de débogage.

c. Génération des fichiers

La troisième et dernière partie traduit les expressions obtenues par la première partie afin qu'elles correspondent aux exigences de format des fichiers dans lesquels elles seront écrites, puis crée lesdits fichiers.

1. Les **Étapes préliminaires** définissent les éléments permettant aux fichiers d'être fonctionnels et récupèrent également des informations à propos de la modélisation comme les options choisies par l'utilisateur par exemple.

2. L'**Écriture du fichier contenant les expressions formelles (LaTeX)** est l'étape qui génère les fichiers contenant les expressions formelles. De petites modifications sont apportées aux expressions afin d'augmenter la lisibilité, telles que le remplacement de « theta » par « θ » par exemple.

3. L'**Écriture du ou des dictionnaires** produit un ou des fichiers associés à un ou des fichiers de calcul. Les dictionnaires permettent à l'utilisateur de connaître les données d'entrée à renseigner lors de l'exécution des fichiers de calcul ainsi que les données de sortie qu'ils fournissent.

4. Le **Remplacement des termes dont les valeurs numériques sont inconnues et la potentielle optimisation des éléments à calculer** changent grandement les expressions, au point de les rendre incompréhensibles sans les dictionnaires correspondants. Les variables sont changées en entrées de tableaux et les expressions sont fragmentées en succession d'équations minimisant le nombre d'opérations.

5. Enfin, l'**Écriture du ou des fichiers des calculs (C++)** inscrit les successions d'équations de l'étape précédente au sein de fichiers disposant de tous les éléments nécessaires à leur exécution ou à celle des fonctions qu'ils contiennent.

8. Exemple de performances

Pour donner une idée des temps de calcul des modèles dynamiques inverses, deux exemples sont présentés dans le tableau ci-dessous. Le premier est un robot RACER-5-0.80 de la marque Comau et le deuxième est le robot humanoïde ORHRO, présenté en détail dans la quatrième partie de ce document.

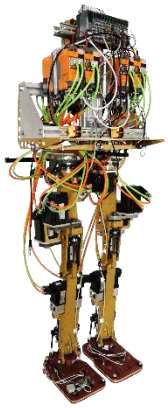
Robot modélisé	**Nombre de couples d'actionnement calculés**	**Processeur**	**Fréquence d'horloge**	**Temps de cycle**
	**6**	ARM Cortex-A9	**667 MHz**	≈ **104 µs** ≤ 157 µs
	**15**	Intel® Atom® E680T	**1.6 GHz**	≈ **68 µs** ≤ 75 µs
		Intel® Core™ i5 7300U	**2.6 GHz**	≈ **8 µs**

Tableau 2. Performances de modèles dynamiques inverses

Ces performances sont similaires à celles qu'on trouve dans la littérature [35].

La MDA est un générateur de modèles dynamiques directs et inverses d'arborescence à chaînes ouvertes. Aussi complexes que leurs descriptions/formulations le permettent, ces modèles peuvent être très complexes et donc simuler fidèlement les systèmes qu'ils représentent.

9. Proposition d'une méthode numérique

Afin de calculer le mouvement d'un mécanisme physique, il faut résoudre un système d'équations différentielles du second ordre à coefficients variables. Il n'existe pas à ce jour de moyen de résoudre formellement ces équations. Il faut donc recourir à l'usage de méthodes numériques.

Durant le développement de la MDA, nous nous sommes penchés sur la performance de ces dernières, à savoir le rapport de leurs précisions par leurs coûts calculatoires. Nous nous sommes rendus compte qu'entre les très connues méthodes d'Euler, d'une étonnante frugalité mais d'une précision toute relative, et méthodes de Runge-Kutta, bien plus précises mais aussi bien plus gourmandes, devait résider un compromis.

Nous avons imaginé une méthode que nous allons vous présenter pour deux exemples d'application élémentaires. À notre connaissance, elle n'a jamais fait l'objet d'une publication.

a. Application au pendule pesant

On notera $\theta(t) = \theta$ pour simplifier les écritures.

$$\ddot{\theta} = -\frac{g}{l} \cdot \sin(\theta)$$

Équation 5

Introduisons $\omega = \frac{d\theta}{dt}$

On a alors :

$$\begin{cases} \dfrac{d\theta}{dt} = \omega \\ \dfrac{d\omega}{dt} = -\dfrac{g}{l} \cdot \sin(\theta) \end{cases}$$

Équation 6

Si on introduit le vecteur $Y = \begin{pmatrix} \theta \\ \omega \end{pmatrix}$, ce système d'équations différentielles peut se mettre sous la forme :

$$Y' = f(Y, t)$$

Équation 7

Euler explicite

La méthode d'Euler explicite nous permet de résoudre numériquement cette équation de la manière suivante à chaque pas de temps :

$$Y_{k+1} = Y_k + h \cdot f(Y_k, t_k)$$

Équation 8

Avec :

$$Y_k = Y(t_k)$$

Équation 9

h : le pas de temps d'intégration

$Y_0 = \begin{pmatrix} \theta_0 \\ \omega_0 \end{pmatrix}$ les conditions initiales

Soit encore :

$$\begin{cases} \theta_{k+1} = \theta_k + h \cdot \omega_k \\ \omega_{k+1} = \omega_k + h \cdot \left(-\frac{g}{l} \cdot \sin(\theta_k)\right) \end{cases}$$

Équation 10

Cette méthode présente le défaut d'être non-symplectique. Ainsi, elle se comporte mal pour le problème conservatif du pendule pesant, les erreurs étant non-négligeables dès quelques oscillations libres du pendule comme le montre la figure suivante pour un pendule de longueur $l = 1m$, une accélération de la pesanteur $g = 9{,}81m/s^2$, un pas de temps $h = 0{,}001s$ et un temps de simulation de $60s$. Le pas de temps a été choisi de manière arbitraire, il s'avère qu'il représente 1/2368ème de la période propre du système.

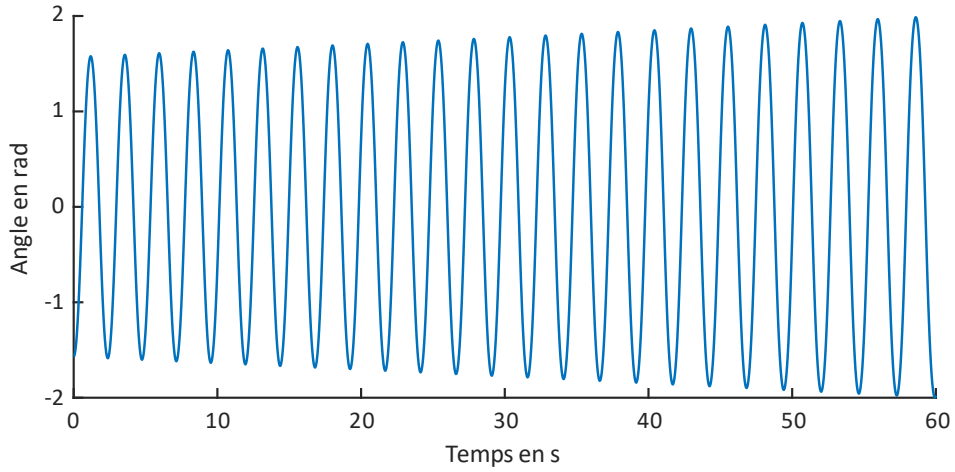


Figure 3. Oscillations du pendule pesant en utilisant la méthode d'Euler explicite

Euler semi-implicite

Une autre technique est d'utiliser la méthode d'Euler semi-implicite. Dans ce cas, on utilise la méthode d'Euler explicite pour calculer d'abord la vitesse puis la méthode d'Euler implicite pour calculer l'angle. On a alors :

$$\begin{cases} \omega_{k+1} = \omega_k + h \cdot \left(-\frac{g}{l} \cdot \sin(\theta_k)\right) \\ \theta_{k+1} = \theta_k + h \cdot \omega_{k+1} \end{cases}$$

Équation 11

Cette méthode se comporte beaucoup mieux dans le cas du pendule pesant.

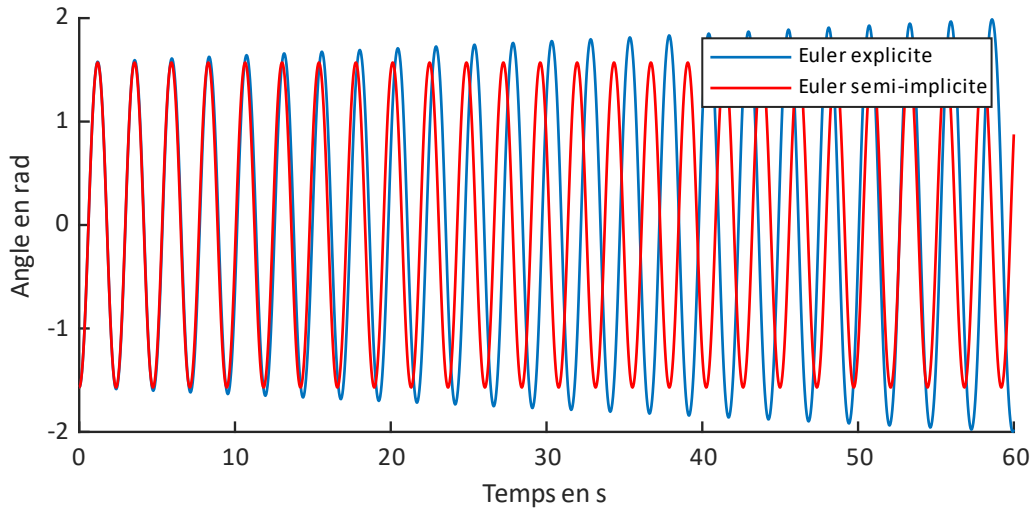


Figure 4. Oscillations du pendule pesant en utilisant les méthodes d'Euler explicite et semi-implicite

Notre méthode

Notre technique s'inspire de cette méthode semi-explicite.
Nous utilisons un ordre supérieur en dérivant une fois de plus notre équation (ce qui revient à calculer l'à-coup du pendule), ce qui nous donne :

$$\begin{cases} \frac{d\theta}{dt} = \omega \\ \frac{d\omega}{dt} = -\frac{g}{l} \cdot \sin(\theta) \end{cases}$$

Équation 12

$$\begin{cases} \frac{d\omega}{dt} = -\frac{g}{l} \cdot \sin(\theta) \\ \frac{d\dot{\omega}}{dt} = -\omega \cdot \frac{g}{l} \cdot \cos(\theta) \end{cases}$$

Équation 13

Ainsi, nous avons le nouveau système :

$$\begin{cases} Y' = f(Y,t) \\ Y'' = f'(Y,t) \end{cases}$$

Équation 14

En utilisant la méthode d'Euler explicite pour la partie $Y'' = f'(Y,t)$, on peut calculer alors :

$$Y'_{k+1} = Y'_k + h \cdot f'(Y_k, t_k)$$

Équation 15

Ou encore, en remplaçant $Y'_k = f(Y_k, t_k)$:

$$Y'_{k+1} = f(Y_k, t_k) + h \cdot f'(Y_k, t_k)$$

Équation 16

Ce qui donne :

$$\begin{cases} \cancel{\omega_{k+1} = \omega_k + h \cdot \left(-\frac{g}{l} \cdot \sin(\theta_k)\right)} \\ \dot{\omega}_{k+1} = -\frac{g}{l} \cdot \sin(\theta_k) + h \cdot \left(-\omega_k \cdot \frac{g}{l} \cdot \cos(\theta_k)\right) \end{cases}$$

Équation 17

La première équation ne sera pas utilisée, en effet, on utilisera pour la partie $Y' = f(Y,t)$, un schéma centré d'intégration (ce qui nous différencie de la méthode d'Euler semi-explicite qui utilise un schéma d'intégration implicite) :

$$Y_{k+1} = Y_k + \frac{h}{2} \cdot \left(f(Y_k, t_k) + f(Y_{k+1}, t_{k+1})\right)$$

Équation 18

On obtient alors :

$$\begin{cases} \theta_{k+1} = \theta_k + \frac{h}{2} \cdot (\omega_k + \omega_{k+1}) \\ \omega_{k+1} = \omega_k + \frac{h}{2} \cdot (\dot{\omega}_k + \dot{\omega}_{k+1}) \end{cases}$$

Équation 19

En injectant le résultat précédent pour $\dot{\omega}_{k+1}$ et en remplaçant $\dot{\omega}_k = -\frac{g}{l} \cdot \sin(\theta_k)$, on obtient :

$$\begin{cases} \dot{\omega}_{k+1} = -\frac{g}{l} \cdot \sin(\theta_k) + h \cdot \left(-\omega_k \cdot \frac{g}{l} \cdot \cos(\theta_k)\right) \\ \omega_{k+1} = \omega_k + \frac{h}{2} \cdot \left(-\frac{g}{l} \cdot \sin(\theta_k) - \frac{g}{l} \cdot \sin(\theta_k) + h \cdot \left(-\omega_k \cdot \frac{g}{l} \cdot \cos(\theta_k)\right)\right) \\ \theta_{k+1} = \theta_k + \frac{h}{2} \cdot \left(\omega_k + \omega_k + \frac{h}{2} \cdot \left(-\frac{g}{l} \cdot \sin(\theta_k) - \frac{g}{l} \cdot \sin(\theta_k) + h \cdot \left(-\omega_k \cdot \frac{g}{l} \cdot \cos(\theta_k)\right)\right)\right) \end{cases}$$

Équation 20

Soit au final :

$$\begin{cases} \dot{\omega}_{k+1} = -\frac{g}{l} \cdot \sin(\theta_k) + h \cdot \left(-\omega_k \cdot \frac{g}{l} \cdot \cos(\theta_k)\right) \\ \omega_{k+1} = \omega_k + h \cdot \left(-\frac{g}{l} \cdot \sin(\theta_k)\right) + \frac{h^2}{2}\left(-\omega_k \cdot \frac{g}{l} \cdot \cos(\theta_k)\right) \\ \theta_{k+1} = \theta_k + h \cdot \omega_k + \frac{h^2}{2} \cdot \left(-\frac{g}{l} \cdot \sin(\theta_k)\right) + \frac{h^3}{4} \cdot \left(-\omega_k \cdot \frac{g}{l} \cdot \cos(\theta_k)\right) \end{cases}$$

Équation 21

Cette méthode semble se comporter de manière similaire à la méthode d'Euler semi-explicite dans le cas du pendule pesant.

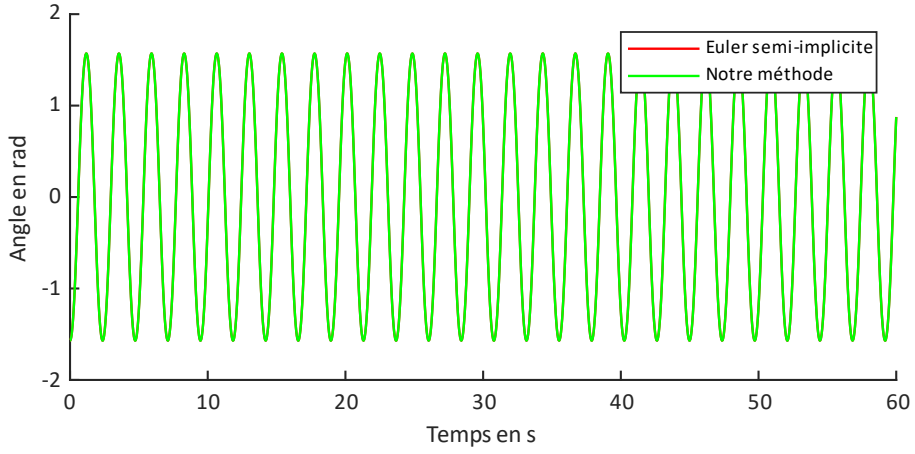


Figure 5. Oscillations du pendule pesant en utilisant la méthode d'Euler semi-implicite et notre méthode

Pour voir l'intérêt de cette méthode, il est nécessaire de comparer les résultats à ceux obtenus à l'aide des fonctions elliptiques de Jacobi, fournissant une solution exacte à ce problème du pendule pesant. Voici l'écart obtenu entre la solution exacte et les trois méthodes présentées :

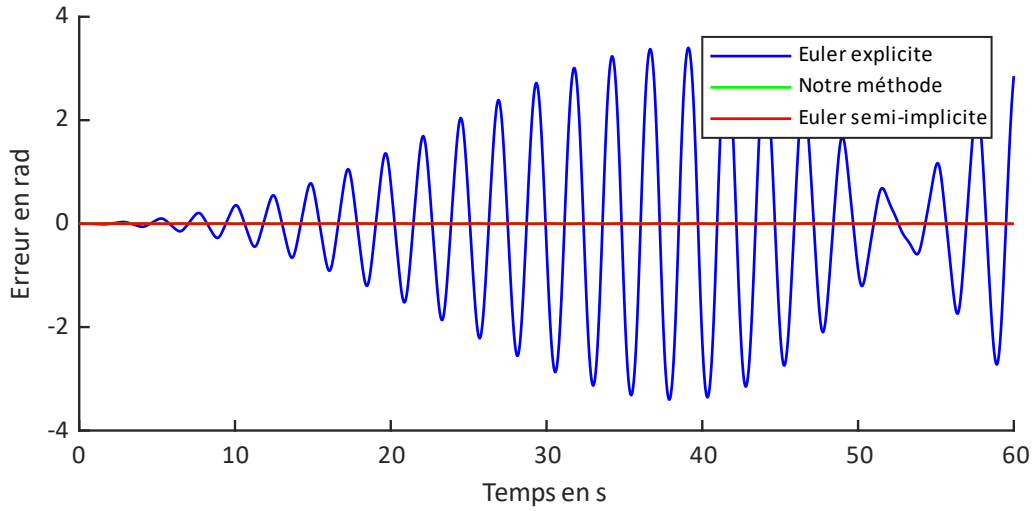


Figure 6. Erreur en position entre la solution exacte et les différentes solutions obtenues

On note que la méthode d'Euler explicite présente une erreur bien supérieure à celle des deux autres méthodes. Comparons maintenant notre méthode et celle d'Euler semi-implicite :

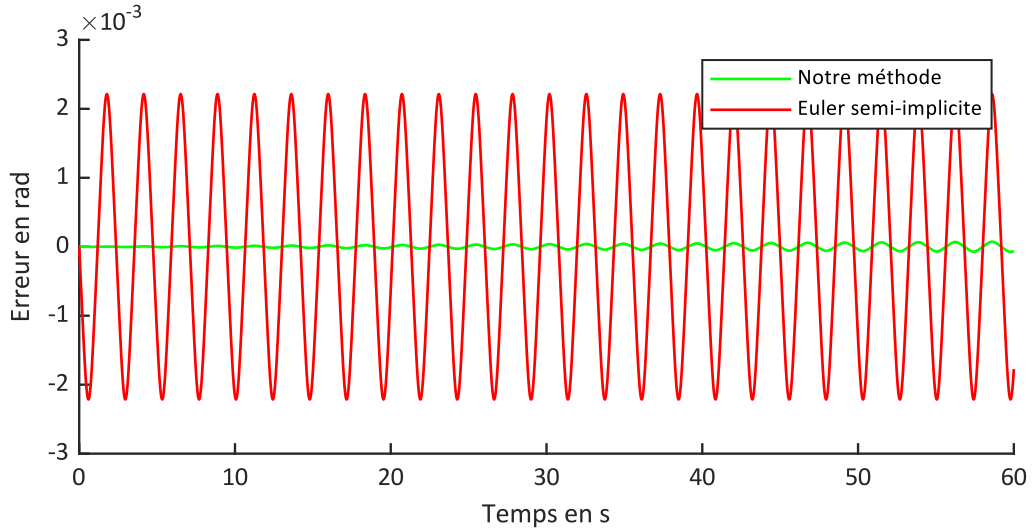


Figure 7. Erreur en position entre la solution exacte et les différentes solutions obtenues.

On observe que l'erreur obtenue est bien inférieure pour notre méthode que pour la méthode d'Euler semi-implicite. On note que l'erreur croit à chaque oscillation pour notre méthode, ce qui n'est pas le cas pour la méthode d'Euler semi-implicite. Le taux de croissance de cette erreur est néanmoins très faible pour notre méthode. Si l'on continue la simulation, l'erreur obtenue par notre méthode devient supérieure à celle de la méthode d'Euler semi-implicite aux environs de 1780s de simulation. Notre méthode est donc adaptée à des simulations sur des temps relativement longs (ici : environ 840 oscillations sont nécessaires pour arriver à un niveau d'erreur similaire entre les 2 méthodes).

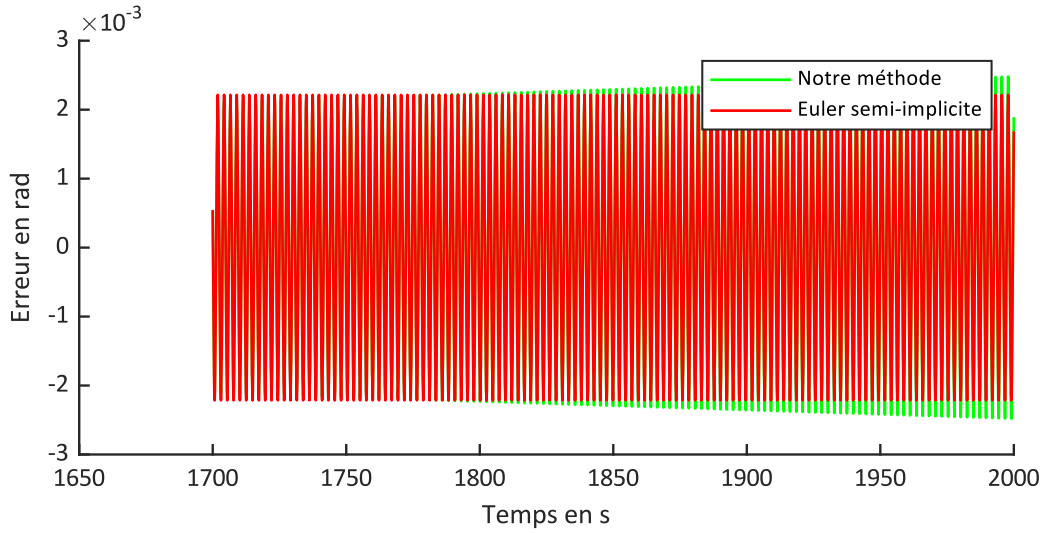


Figure 8. Erreur en position entre la solution exacte et les différentes solutions obtenues

Runge-Kutta à l'ordre 2

Comparons notre méthode à la méthode de Runge-Kutta d'ordre 2 :

$$Y_{k+1} = Y_k + h \cdot f\left(Y_k + \frac{h}{2} \cdot f(Y_k, t_k), t_k + \frac{h}{2}\right)$$

Équation 22

Calculons d'abord $Y_k + \frac{h}{2} \cdot f(Y_k, t_k)$:

$$Y_k + \frac{h}{2} \cdot f(Y_k, t_k) = \begin{pmatrix} \theta_k \\ \omega_k \end{pmatrix} + \frac{h}{2} \cdot \begin{pmatrix} \omega_k \\ -\frac{g}{l} \cdot \sin(\theta_k) \end{pmatrix}$$
$$= \begin{pmatrix} \theta_k + \frac{h}{2} \cdot \omega_k \\ \omega_k - \frac{h}{2} \cdot \frac{g}{l} \cdot \sin(\theta_k) \end{pmatrix}$$

Équation 23

On obtient au final :

$$\begin{cases} \theta_{k+1} = \theta_k + h \cdot \left(\omega_k - \frac{h}{2} \cdot \frac{g}{l} \cdot \sin(\theta_k)\right) \\ \omega_{k+1} = \omega_k + h \cdot \left(-\frac{g}{l} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k\right)\right) \end{cases}$$

Équation 24

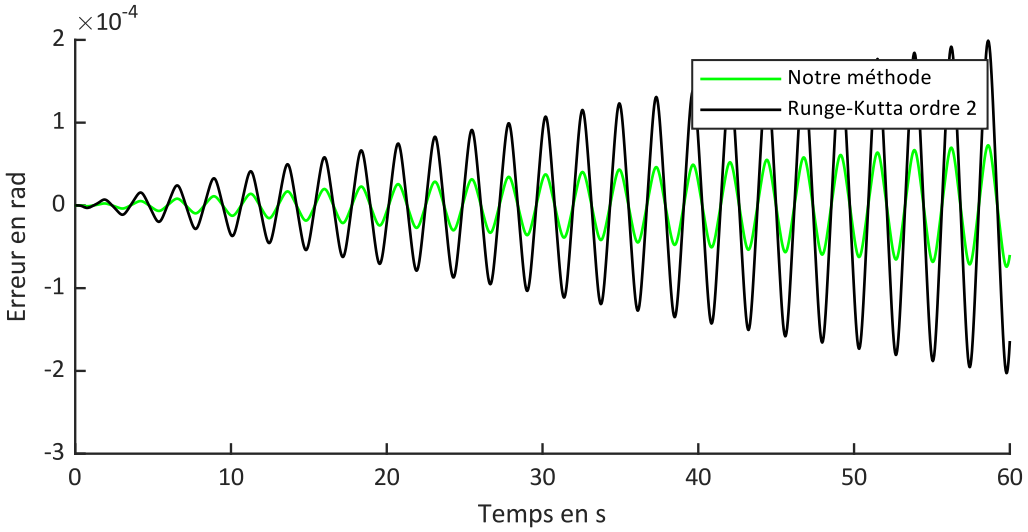


Figure 9. Erreur en position entre la solution exacte et les différentes solutions obtenues

On observe que l'erreur obtenue est bien inférieure pour notre méthode en comparaison avec la méthode de Runge-Kutta d'ordre 2. Le taux de croissance de cette erreur est plus faible pour notre méthode que pour celle de Runge-Kutta d'ordre 2.

Runge-Kutta à l'ordre 4

Comparons notre méthode à la méthode de Runge-Kutta d'ordre 4 :

$$Y_{k+1} = Y_k + \frac{h}{6} \cdot (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4)$$

Équation 25

Avec :

$$\begin{cases} k_1 = f(Y_k, t_k) \\ k_2 = f(Y_k + \frac{h}{2} \cdot k_1, t_k + \frac{h}{2}) \\ k_3 = f(Y_k + \frac{h}{2} \cdot k_2, t_k + \frac{h}{2}) \\ k_4 = f(Y_k + h \cdot k_3, t_k + h) \end{cases}$$

Équation 26

$$\boxed{k_1 = \begin{pmatrix} \omega_k \\ -\frac{g}{l} \cdot \sin(\theta_k) \end{pmatrix}}$$

Équation 27

$$Y_k + \frac{h}{2} \cdot k_1 = \begin{pmatrix} \theta_k \\ \omega_k \end{pmatrix} + \frac{h}{2} \cdot \begin{pmatrix} \omega_k \\ -\frac{g}{l} \cdot \sin(\theta_k) \end{pmatrix} = \begin{pmatrix} \theta_k + \frac{h}{2} \cdot \omega_k \\ \omega_k - \frac{h}{2} \cdot \frac{g}{l} \cdot \sin(\theta_k) \end{pmatrix}$$

Équation 28

$$\boxed{k_2 = \begin{pmatrix} \omega_k - \frac{h}{2} \cdot \frac{g}{l} \cdot \sin(\theta_k) \\ -\frac{g}{l} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k\right) \end{pmatrix}}$$

Équation 29

$$Y_k + \frac{h}{2} \cdot k_2 = \begin{pmatrix} \theta_k \\ \omega_k \end{pmatrix} + \frac{h}{2} \cdot \begin{pmatrix} \omega_k - \frac{h}{2} \cdot \frac{g}{l} \cdot \sin(\theta_k) \\ -\frac{g}{l} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k\right) \end{pmatrix} = \begin{pmatrix} \theta_k + \frac{h}{2} \cdot \omega_k - \frac{h^2}{4} \cdot \frac{g}{l} \cdot \sin(\theta_k) \\ \omega_k - \frac{g}{l} \cdot \frac{h}{2} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k\right) \end{pmatrix}$$

Équation 30

$$\boxed{k_3 = \begin{pmatrix} \omega_k - \frac{g}{l} \cdot \frac{h}{2} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k\right) \\ -\frac{g}{l} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k - \frac{h^2}{4} \cdot \frac{g}{l} \cdot \sin(\theta_k)\right) \end{pmatrix}}$$

Équation 31

$$Y_k + h \cdot k_3 = \begin{pmatrix} \theta_k \\ \omega_k \end{pmatrix} + h \cdot \begin{pmatrix} \omega_k - \frac{g}{l} \cdot \frac{h}{2} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k\right) \\ -\frac{g}{l} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k - \frac{h^2}{4} \cdot \frac{g}{l} \cdot \sin(\theta_k)\right) \end{pmatrix}$$

$$= \begin{pmatrix} \theta_k + h \cdot \omega_k - \frac{g}{l} \cdot \frac{h^2}{2} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k\right) \\ \omega_k - h \cdot \frac{g}{l} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k - \frac{h^2}{4} \cdot \frac{g}{l} \cdot \sin(\theta_k)\right) \end{pmatrix}$$

Équation 32

$$\boxed{k_4 = \begin{pmatrix} \omega_k - h \cdot \frac{g}{l} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k - \frac{h^2}{4} \cdot \frac{g}{l} \cdot \sin(\theta_k)\right) \\ -\frac{g}{l} \cdot \sin\left(\theta_k + h \cdot \omega_k - \frac{g}{l} \cdot \frac{h^2}{2} \cdot \sin\left(\theta_k + \frac{h}{2} \cdot \omega_k\right)\right) \end{pmatrix}}$$

Équation 33

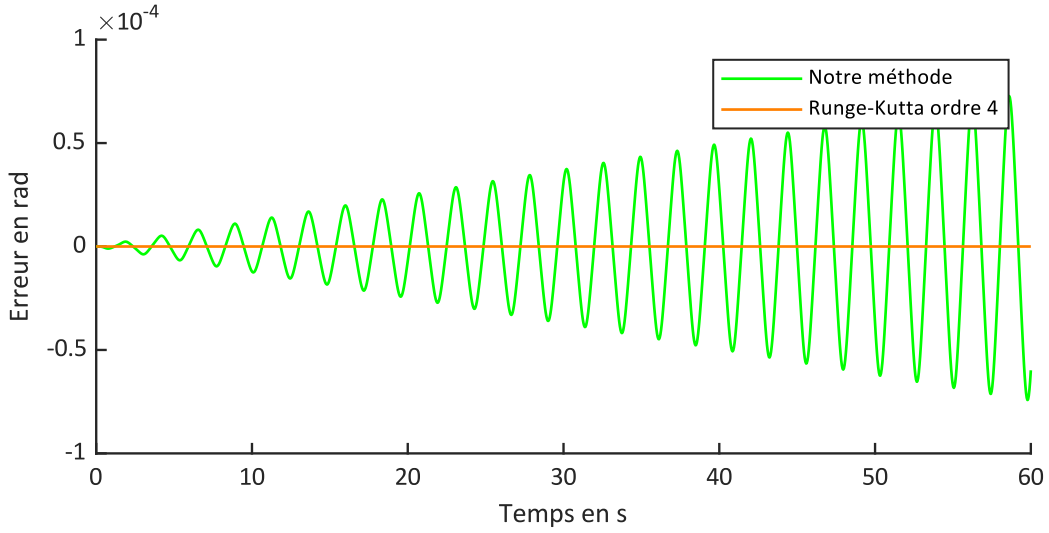


Figure 10. Erreur en position entre la solution exacte et les différentes solutions obtenues

On observe que l'erreur obtenue est bien inférieure pour la méthode de Runge-Kutta d'ordre 4 en comparaison avec notre méthode.

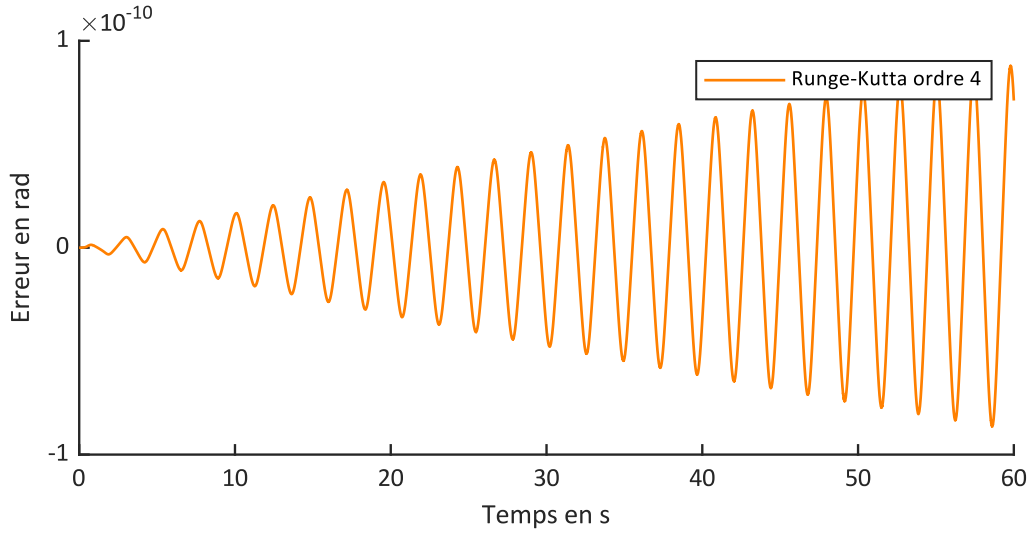


Figure 11. Erreur en position entre la solution exacte et les différentes solutions obtenues

La précision est bien meilleure pour la méthode de Runge-Kutta d'ordre 4 par rapport à notre méthode.

Cette précision se fait au prix d'une complexité calculatoire bien supérieure. Notre méthode permet d'atteindre des performances supérieures à celle de la méthode de Runge-Kutta d'ordre 2 tout en gardant une complexité calculatoire modérée par rapport à la méthode de Runge-Kutta d'ordre 4. Elle apparaît donc particulièrement pertinente dans l'optique d'un portage sur le calculateur industriel d'une machine automatisée.

Taylor à l'ordre 3

$$\begin{cases} \omega_{k+1} = \omega_k + h \cdot \left(-\frac{g}{l} \cdot \sin(\theta_k)\right) + \frac{h^2}{2} \cdot \omega_k \cdot \left(-\frac{g}{l} \cdot \cos(\theta_k)\right) \\ \theta_{k+1} = \theta_k + h \cdot \omega_k + \frac{h^2}{2} \cdot \left(-\frac{g}{l} \cdot \sin(\theta_k)\right) + \frac{h^3}{6} \cdot \omega_k \cdot \left(-\frac{g}{l} \cdot \cos(\theta_k)\right) \end{cases}$$

Équation 34

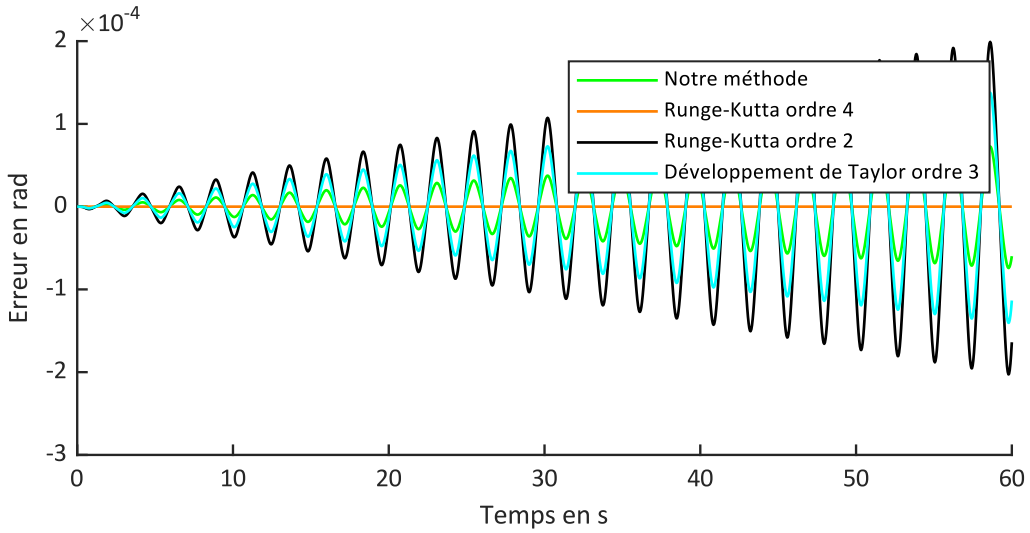


Figure 12. Erreur en position entre la solution exacte et les différentes solutions obtenues

On voit que le développement de Taylor à l'ordre 3 est très compétitif avec des résultats compris entre la méthode de Runge-Kutta d'ordre 2 et notre méthode.

Nous venons de voir le comportement des différentes méthodes pour un pas de temps fixé.
Le voici pour une durée totale de 10 secondes, pour plusieurs pas de temps différents et pour deux précisions :

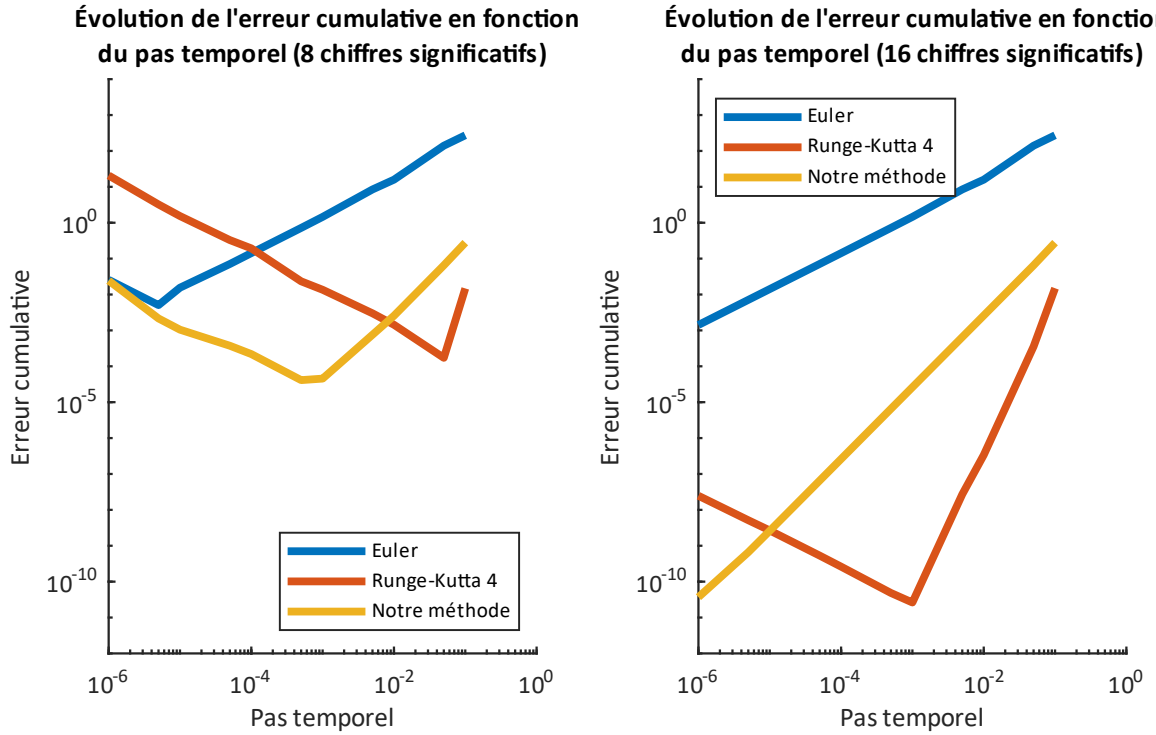


Figure 13

On remarque un très bon comportement de notre méthode pour le calcul avec 8 chiffres significatifs, Runge-Kutta, du fait de ses multiples évaluations cumule les erreurs d'arrondi. Les trois méthodes présentent un profil en « V » caractéristique. L'optimum de la méthode d'Euler nécessite un échantillonnage à 10^{-5} seconde, impliquant beaucoup de calculs. Notre méthode propose la meilleure précision à la 10^{-3} seconde.
Pour le calcul à 16 chiffres significatifs, seul Runge-Kutta présente un profil en « V », son optimum étant à la milliseconde. Comme illustré sur la figure 10, elle est plus précise que notre méthode (pour des pas de temps supérieur à 10^{-5} seconde).
Notre méthode constitue donc un intéressant compromis pour le rapport de la précision sur le coût calculatoire pour les calculs en double précision. En simple, elle surclasse Euler et Runge-Kutta à l'ordre 4.

b. Application à l'oscillateur harmonique amorti

On notera $x(t) = x$ pour simplifier les écritures.

$$\ddot{X} = -\frac{c}{m} \cdot \dot{X} - \frac{k}{m} \cdot X$$

Équation 35

Introduisons $V = \frac{dX}{dt}$

On a alors :

$$\begin{cases} \dfrac{dX}{dt} = V \\ \dfrac{dV}{dt} = -\dfrac{c}{m} \cdot V - \dfrac{k}{m} \cdot X \end{cases}$$

Équation 36

Si on introduit le vecteur $Y = \begin{pmatrix} X \\ V \end{pmatrix}$, ce système d'équations différentielles peut se mettre sous la forme :

$$Y' = f(Y, t)$$

Équation 37

Euler explicite

La méthode d'Euler explicite nous permet de résoudre numériquement cette équation de la manière suivante à chaque pas de temps :

$$Y_{k+1} = Y_k + h \cdot f(Y_k, t_k)$$

Équation 38

Avec :

$$Y_k = Y(t_k)$$

Équation 39

h : le pas de temps d'intégration

$Y_0 = \begin{pmatrix} X_0 \\ V_0 \end{pmatrix}$ les conditions initiales

Soit encore :

$$\begin{cases} X_{k+1} = X_k + h \cdot V_k \\ V_{k+1} = V_k + h \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k \right) \end{cases}$$

Équation 40

Cette fois-ci, le comportement semble cohérent avec un système masse-ressort en régime pseudopériodique comme le montre la figure suivante, obtenue pour une masse $m = 1kg$, un coefficient de frottement visqueux $c = 1\ N.s.m^{-1}$, une raideur $k = 10000N/m$, un pas de temps $h = 0{,}0000265s$ et un temps de simulation de $2s$. Le pas de temps a été choisi afin qu'il représente la même fraction de la période propre du système que celle utilisée pour le pendule (soit 1/2368ème).

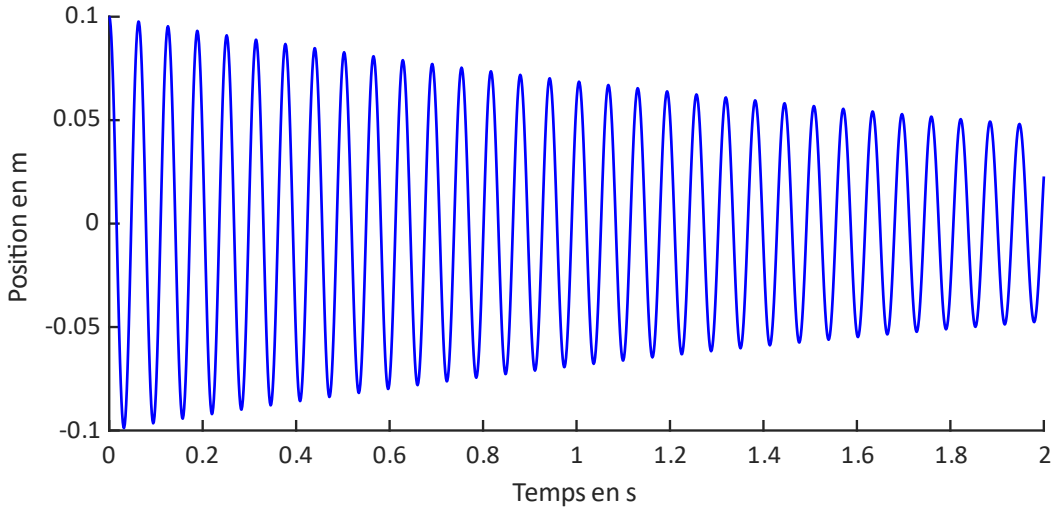


Figure 14. Oscillations de la masse en utilisant la méthode d'Euler explicite

Euler semi-implicite

Une autre technique est d'utiliser la méthode d'Euler semi-implicite. Dans ce cas, on utilise la méthode d'Euler explicite pour calculer d'abord la vitesse puis la méthode d'Euler implicite pour calculer la position. On a alors :

$$\begin{cases} V_{k+1} = V_k + h \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k \right) \\ X_{k+1} = X_k + h \cdot V_{k+1} \end{cases}$$

Équation 41

On observe un impact plus important de l'amortissement sur le système avec la méthode d'Euler semi-implicite.

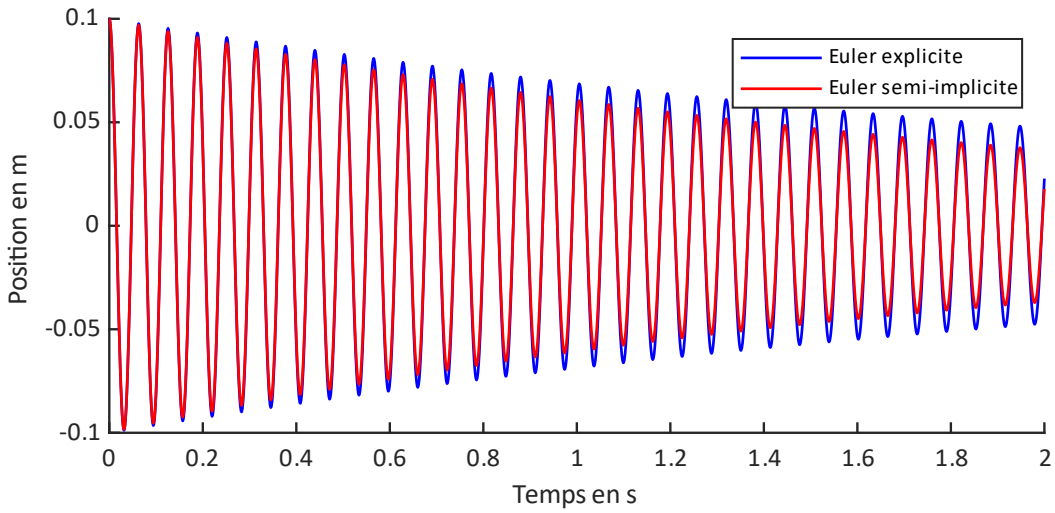


Figure 15. Oscillations de la masse en utilisant les méthodes d'Euler explicite et semi-implicite

Notre méthode

Notre technique s'inspire de cette méthode semi-explicite.
Nous utilisons un ordre supérieur en dérivant une fois de plus notre équation (ce qui revient à calculer le jerk de l'oscillateur), ce qui nous donne :

$$\begin{cases} \dfrac{dX}{dt} = V \\ \dfrac{dV}{dt} = -\dfrac{c}{m} \cdot V - \dfrac{k}{m} \cdot X \end{cases}$$

Équation 42

$$\begin{cases} \dfrac{dV}{dt} = -\dfrac{c}{m} \cdot V - \dfrac{k}{m} \cdot X \\ \dfrac{d\dot{V}}{dt} = -\dfrac{c}{m} \cdot \dot{V} - \dfrac{k}{m} \cdot V \end{cases}$$

Équation 43

Ainsi, nous avons le nouveau système :

$$\begin{cases} Y' = f(Y,t) \\ Y'' = f'(Y,t) \end{cases}$$

Équation 44

En utilisant la méthode d'Euler explicite pour la partie $Y'' = f'(Y,t)$, on peut calculer alors :

$$Y'_{k+1} = Y'_k + h \cdot f'(Y_k, t_k)$$

Équation 45

Ou encore, en remplaçant $Y'_k = f(Y_k, t_k)$:

$$Y'_{k+1} = f(Y_k, t_k) + h \cdot f'(Y_k, t_k)$$

Équation 46

Ce qui donne :

$$\begin{cases} \cancel{V_{k+1} = V_k + h \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k\right)} \\ \dot{V}_{k+1} = -\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k + h \cdot \left(-\frac{c}{m} \cdot \dot{V}_k - \frac{k}{m} \cdot V_k\right) \end{cases}$$

Équation 47

La première équation ne sera pas utilisée, en effet, on utilisera pour la partie $Y' = f(Y, t)$, un schéma centré d'intégration (ce qui nous différencie de la méthode d'Euler semi-explicite qui utilise un schéma d'intégration implicite) :

$$Y_{k+1} = Y_k + \frac{h}{2} \cdot \left(f(Y_k, t_k) + f(Y_{k+1}, t_{k+1})\right)$$

Équation 48

On obtient alors :

$$\begin{cases} X_{k+1} = X_k + \frac{h}{2} \cdot (V_k + V_{k+1}) \\ V_{k+1} = V_k + \frac{h}{2} \cdot \left(\dot{V}_k + \dot{V}_{k+1}\right) \end{cases}$$

Équation 49

En injectant le résultat précédent pour $\dot{V}_{k+1}$, on obtient :

$$\begin{cases} \dot{V}_{k+1} = -\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k + h \cdot \left(-\frac{c}{m} \cdot \dot{V}_k - \frac{k}{m} \cdot V_k\right) \\ V_{k+1} = V_k + \frac{h}{2} \cdot \left(\dot{V}_k + \dot{V}_{k+1}\right) \\ X_{k+1} = X_k + \frac{h}{2} \cdot (V_k + V_{k+1}) \end{cases}$$

Équation 50

Cette méthode semble se comporter de manière similaire à la méthode d'Euler semi-explicite dans le cas de l'oscillateur harmonique amorti.

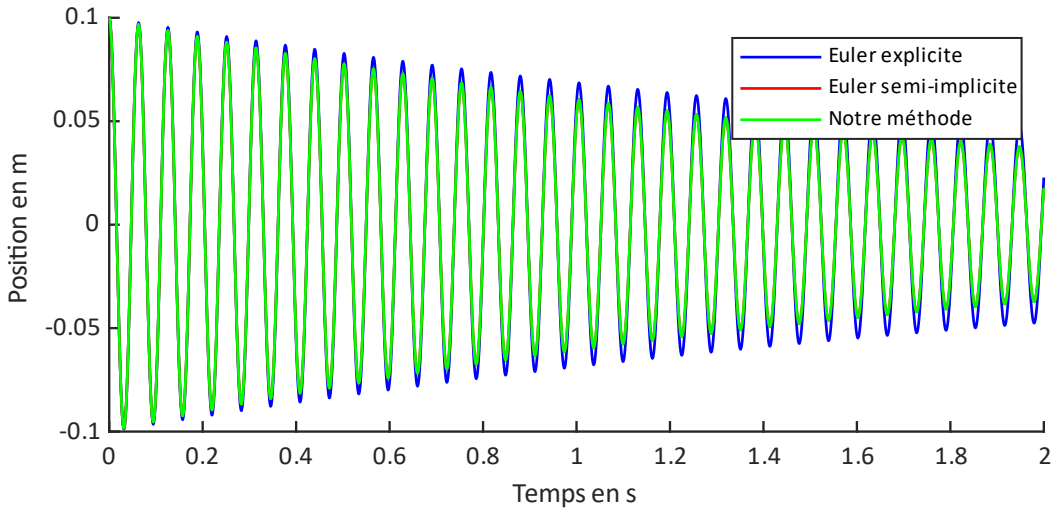


Figure 16. Oscillations de la masse en utilisant les méthodes d'Euler explicite, semi-implicite et notre méthode

Pour voir l'intérêt de cette méthode, il est nécessaire de comparer les résultats à ceux de la solution exacte à ce problème de l'oscillateur harmonique. Notre système fonctionne dans le régime pseudopériodique, la solution de l'équation différentielle du mouvement est donc de la forme :

$$e^{-u \cdot t} \cdot (A \cdot \cos(v \cdot t) + B \cdot \sin(v \cdot t))$$

Équation 51

Avec u, respectivement v, la partie réelle, respectivement imaginaire, des racines complexes (de la forme $u \mp i \cdot v$) de l'équation caractéristique :

$$m \cdot s^2 + c \cdot x + k = 0$$

Équation 52

Voici l'écart obtenu entre la solution exacte et les trois méthodes présentées :

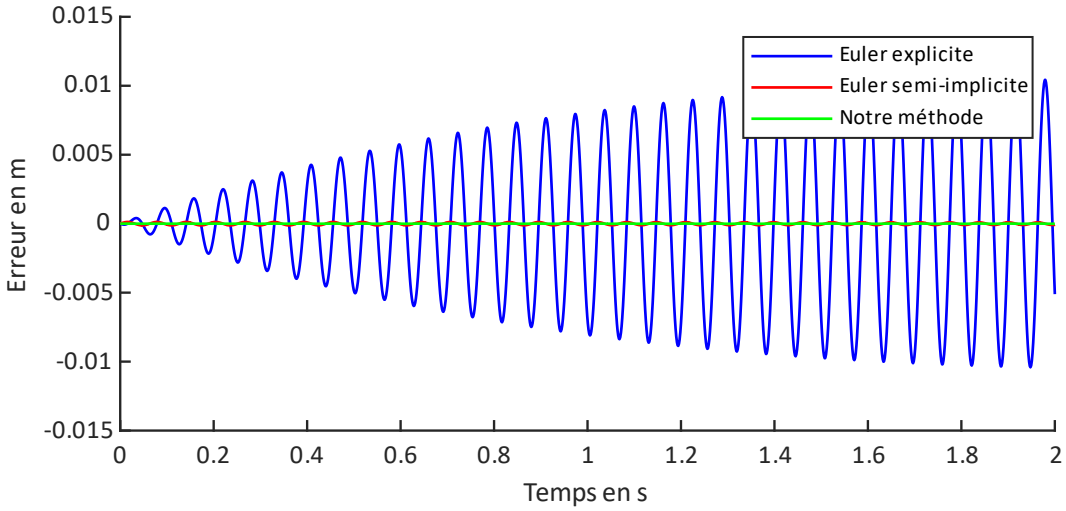


Figure 17. Erreur en position entre la solution exacte et les différentes solutions obtenues

On note que la méthode d'Euler explicite présente une erreur bien supérieure à celle des deux autres méthodes. Comparons maintenant notre méthode et celle d'Euler semi-implicite :

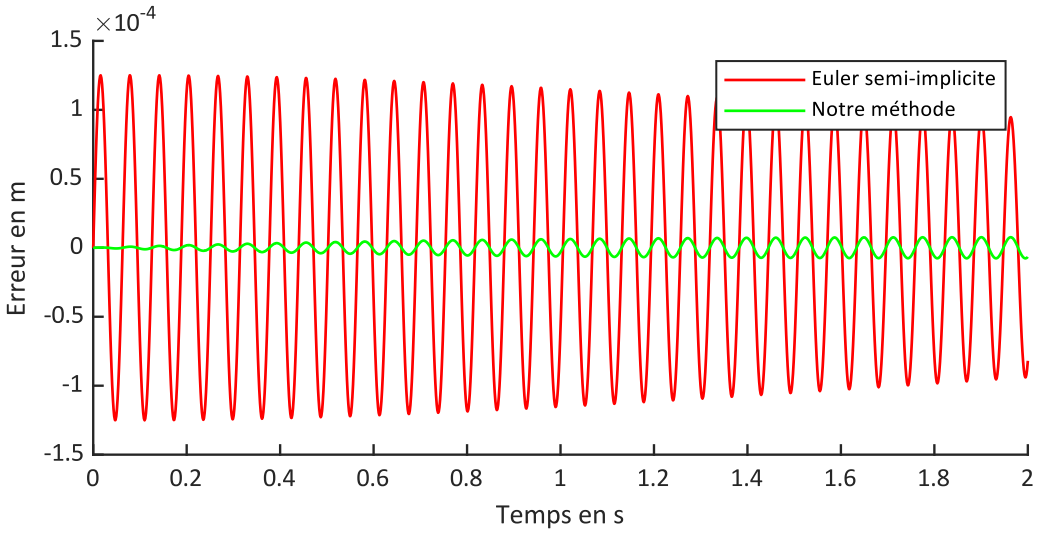


Figure 18. Erreur en position entre la solution exacte et les différentes solutions obtenues

On observe que l'erreur obtenue est bien inférieure pour notre méthode que pour la méthode d'Euler semi-implicite. On note que l'erreur décroit pour la méthode d'Euler semi-implicite ce qui ne semble pas être le cas pour notre méthode. Si l'on continue la simulation jusqu'à $20s$, l'erreur obtenue par notre méthode décroit elle aussi et reste toujours inférieure à celle obtenue à l'aide de la méthode d'Euler semi-implicite.

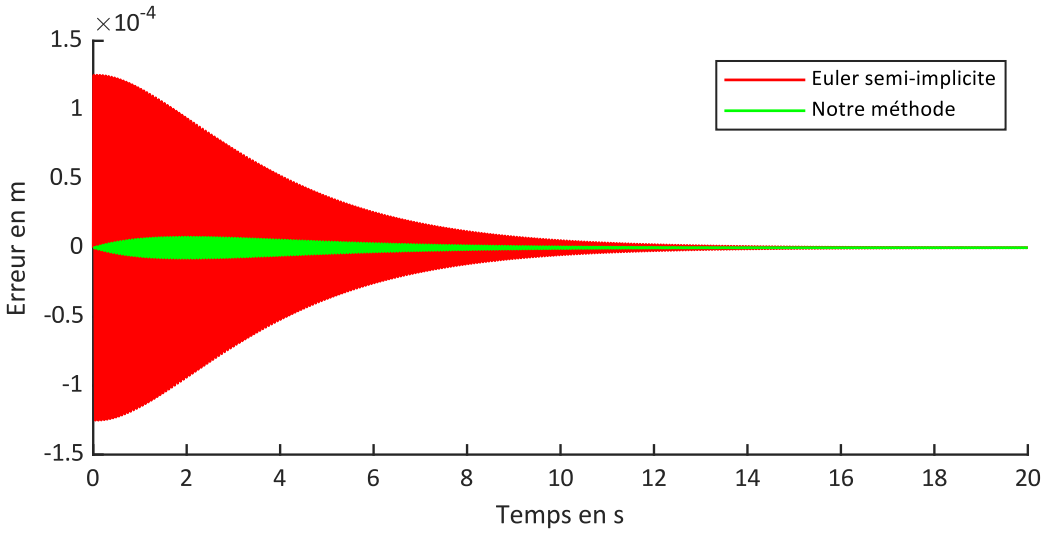


Figure 19. Erreur en position entre la solution exacte et les différentes solutions obtenues

Runge-Kutta à l'ordre 2

Comparons notre méthode à la méthode de Runge-Kutta d'ordre 2 :

$$Y_{k+1} = Y_k + h \cdot f\left(Y_k + \frac{h}{2} \cdot f(Y_k, t_k), t_k + \frac{h}{2}\right)$$

Équation 53

Calculons d'abord $Y_k + \frac{h}{2} \cdot f(Y_k, t_k)$:

$$Y_k + \frac{h}{2} \cdot f(Y_k, t_k) = \begin{pmatrix} X_k \\ V_k \end{pmatrix} + \frac{h}{2} \cdot \begin{pmatrix} V_k \\ -\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k \end{pmatrix} = \begin{pmatrix} X_k + \frac{h}{2} \cdot V_k \\ -\frac{h}{2} \cdot \frac{k}{m} \cdot X_k + V_k - \frac{h}{2} \cdot \frac{c}{m} \cdot V_k \end{pmatrix}$$

Équation 54

On obtient alors :

$$\begin{cases} X_{k+1} = X_k + h \cdot \left(-\frac{h}{2} \cdot \frac{k}{m} \cdot X_k + V_k - \frac{h}{2} \cdot \frac{c}{m} \cdot V_k\right) \\ V_{k+1} = V_k + h \cdot \left(-\frac{c}{m} \cdot \left(-\frac{h}{2} \cdot \frac{k}{m} \cdot X_k + V_k - \frac{h}{2} \cdot \frac{c}{m} \cdot V_k\right) - \frac{k}{m} \cdot \left(X_k + \frac{h}{2} \cdot V_k\right)\right) \end{cases}$$

Équation 55

Ce qui donne au final :

$$\begin{cases} X_{k+1} = X_k - \frac{h^2}{2} \cdot \frac{k}{m} \cdot X_k + h \cdot V_k - \frac{h^2}{2} \cdot \frac{c}{m} \cdot V_k \\ V_{k+1} = V_k + \frac{h^2}{2} \cdot \frac{c}{m} \cdot \frac{k}{m} \cdot X_k - h \cdot \frac{k}{m} \cdot X_k - h \cdot \frac{c}{m} \cdot V_k + \frac{h^2}{2} \cdot \frac{c^2}{m^2} \cdot V_k - \frac{h^2}{2} \cdot \frac{k}{m} \cdot V_k \end{cases}$$

Équation 56

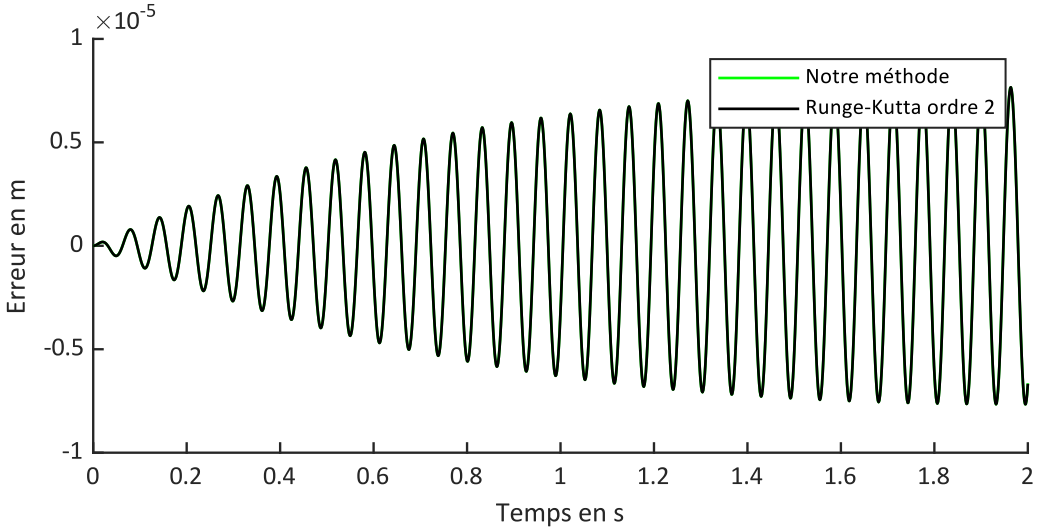


Figure 20. Erreur en position entre la solution exacte et les différentes solutions obtenues

On observe que l'erreur obtenue est quasi-identique pour notre méthode en comparaison avec la méthode de Runge-Kutta d'ordre 2. Notre méthode est très légèrement plus précise si l'on zoome sur le pic positif de la dernière oscillation par exemple.

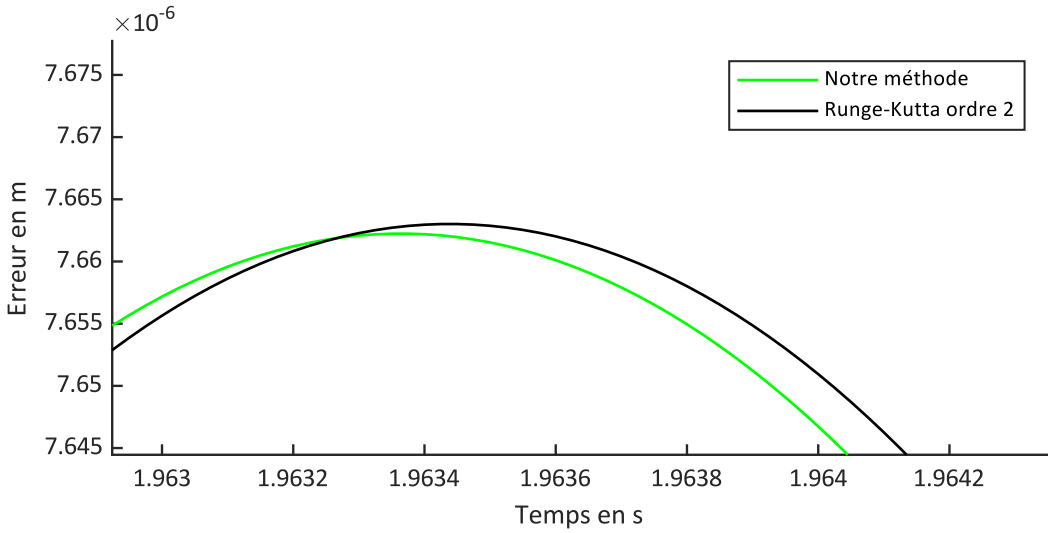


Figure 21. Erreur en position entre la solution exacte et les différentes solutions obtenues

Runge-Kutta à l'ordre 4

Comparons notre méthode à la méthode de Runge-Kutta d'ordre 4 :

$$Y_{k+1} = Y_k + \frac{h}{6} \cdot (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4)$$

Équation 57

Avec :

$$\begin{cases} k_1 = f(Y_k, t_k) \\ k_2 = f(Y_k + \frac{h}{2} \cdot k_1, t_k + \frac{h}{2}) \\ k_3 = f(Y_k + \frac{h}{2} \cdot k_2, t_k + \frac{h}{2}) \\ k_4 = f(Y_k + h \cdot k_3, t_k + h) \end{cases}$$

Équation 58

Pour simplifier les écritures, nous exprimerons k_i en fonction de k_{i-1} sans aller jusqu'à l'écriture formelle complète des k_i en fonction des paramètres.

$$\boxed{k_1 = \begin{pmatrix} V_k \\ -\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k \end{pmatrix} = \begin{pmatrix} k_{11} \\ k_{12} \end{pmatrix}}$$

Équation 59

$$Y_k + \frac{h}{2} \cdot k_1 = \begin{pmatrix} X_k \\ V_k \end{pmatrix} + \frac{h}{2} \cdot \begin{pmatrix} k_{11} \\ k_{12} \end{pmatrix} = \begin{pmatrix} X_k + \frac{h}{2} \cdot k_{11} \\ V_k + \frac{h}{2} \cdot k_{12} \end{pmatrix}$$

Équation 60

$$\boxed{k_2 = \begin{pmatrix} V_k + \frac{h}{2} \cdot k_{12} \\ -\frac{c}{m} \cdot \left(V_k + \frac{h}{2} \cdot k_{12}\right) - \frac{k}{m} \cdot \left(X_k + \frac{h}{2} \cdot k_{11}\right) \end{pmatrix} = \begin{pmatrix} k_{21} \\ k_{22} \end{pmatrix}}$$

Équation 61

$$Y_k + \frac{h}{2} \cdot k_2 = \begin{pmatrix} X_k \\ V_k \end{pmatrix} + \frac{h}{2} \cdot \begin{pmatrix} k_{21} \\ k_{22} \end{pmatrix} = \begin{pmatrix} X_k + \frac{h}{2} \cdot k_{21} \\ V_k + \frac{h}{2} \cdot k_{22} \end{pmatrix}$$

Équation 62

$$k_3 = \begin{pmatrix} V_k + \frac{h}{2} \cdot k_{22} \\ -\frac{c}{m} \cdot \left(V_k + \frac{h}{2} \cdot k_{22}\right) - \frac{k}{m} \cdot \left(X_k + \frac{h}{2} \cdot k_{21}\right) \end{pmatrix} = \begin{pmatrix} k_{31} \\ k_{32} \end{pmatrix}$$

Équation 63

$$Y_k + h \cdot k_3 = \begin{pmatrix} X_k \\ V_k \end{pmatrix} + h \cdot \begin{pmatrix} k_{31} \\ k_{32} \end{pmatrix} = \begin{pmatrix} X_k + h \cdot k_{31} \\ V_k + h \cdot k_{32} \end{pmatrix}$$

Équation 64

$$k_4 = \begin{pmatrix} V_k + h \cdot k_{32} \\ -\frac{c}{m} \cdot (V_k + h \cdot k_{32}) - \frac{k}{m} \cdot (X_k + h \cdot k_{31}) \end{pmatrix}$$

Équation 65

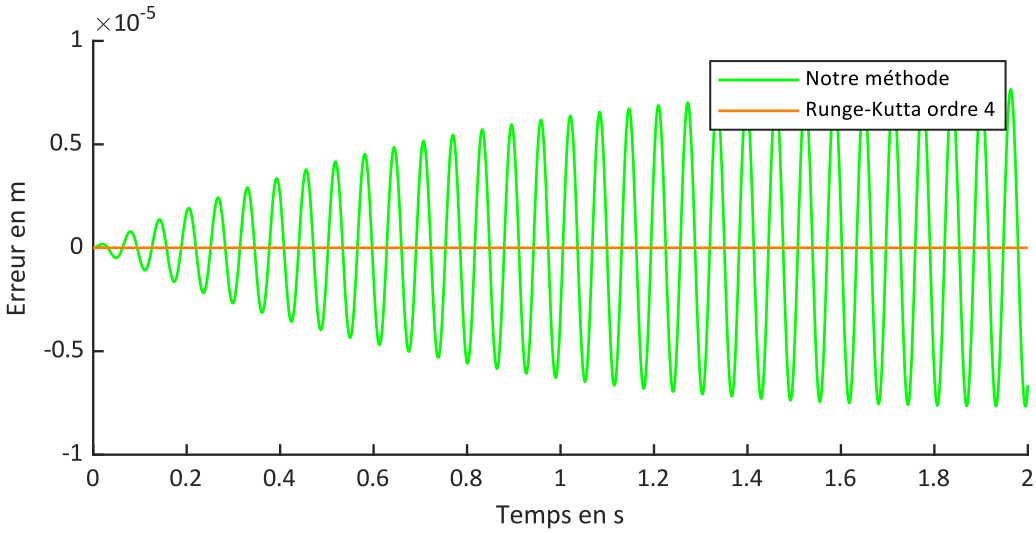


Figure 22. Erreur en position entre la solution exacte et les différentes solutions obtenues

On observe que l'erreur obtenue est bien inférieure pour la méthode de Runge-Kutta d'ordre 4 en comparaison avec notre méthode.

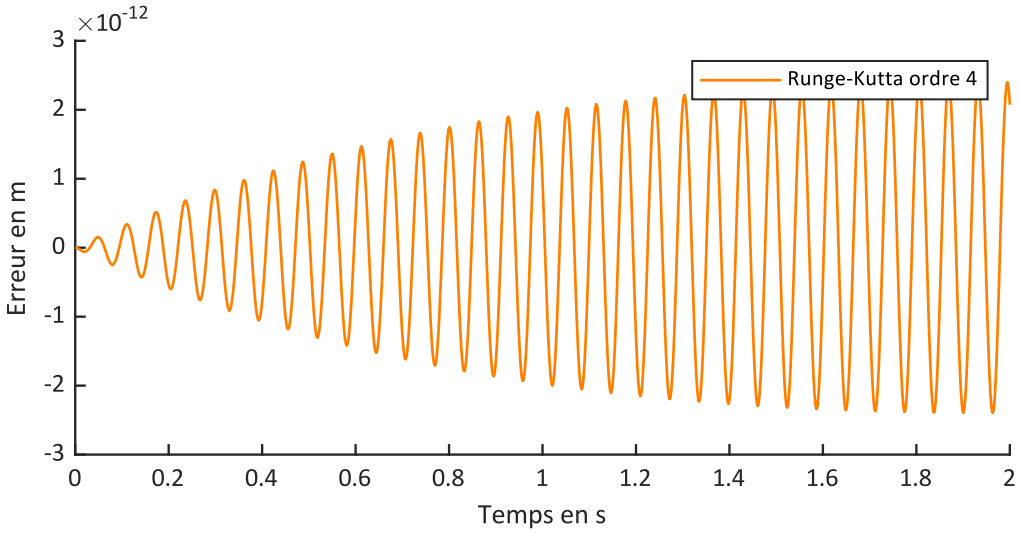


Figure 23. Erreur en position entre la solution exacte et les différentes solutions obtenues

La précision est bien meilleure pour la méthode de Runge-Kutta d'ordre 4 par rapport à notre méthode.

Cette précision se fait au prix d'une complexité calculatoire bien supérieure. Notre méthode permet d'atteindre des performances légèrement supérieures à celle de la méthode de Runge-Kutta d'ordre 2 tout en gardant une complexité calculatoire modérée par rapport à la méthode de Runge-Kutta d'ordre 4. Elle apparaît donc particulièrement pertinente dans l'optique d'un portage sur le calculateur industriel d'une machine automatisée.

Taylor à l'ordre 3

$$\begin{cases} V_{k+1} = V_k + h \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k\right) + \frac{h^2}{2} \cdot \left(-\frac{c}{m} \cdot \dot{V}_k - \frac{k}{m} \cdot V_k\right) \\ \theta_{k+1} = \theta_k + h \cdot V_k + \frac{h^2}{2} \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k\right) + \frac{h^3}{6} \cdot \left(-\frac{c}{m} \cdot \dot{V}_k - \frac{k}{m} \cdot V_k\right) \end{cases}$$

Équation 66

Or :

$$\dot{V}_k = -\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k$$

Équation 67

D'où :

$$\begin{cases} V_{k+1} = V_k + h \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k\right) + \frac{h^2}{2} \cdot \left(-\frac{c}{m} \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k\right) - \frac{k}{m} \cdot V_k\right) \\ \theta_{k+1} = \theta_k + h \cdot V_k + \frac{h^2}{2} \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k\right) + \frac{h^3}{6} \cdot \left(-\frac{c}{m} \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k\right) - \frac{k}{m} \cdot V_k\right) \end{cases}$$

Équation 68

Soit au final :

$$\begin{cases} V_{k+1} = V_k + h \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k\right) + \frac{h^2}{2} \cdot \left(\frac{c}{m} \cdot \frac{k}{m} \cdot X_k + \frac{c^2}{m^2} \cdot V_k - \frac{k}{m} \cdot V_k\right) \\ \theta_{k+1} = \theta_k + h \cdot V_k + \frac{h^2}{2} \cdot \left(-\frac{c}{m} \cdot V_k - \frac{k}{m} \cdot X_k\right) + \frac{h^3}{6} \cdot \left(\frac{c}{m} \cdot \frac{k}{m} \cdot X_k + \frac{c^2}{m^2} \cdot V_k - \frac{k}{m} \cdot V_k\right) \end{cases}$$

Équation 69

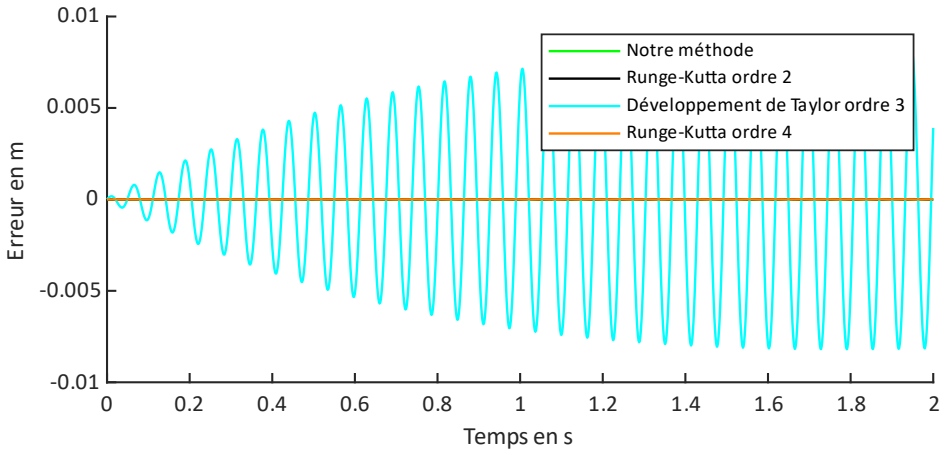


Figure 24. Erreur en position entre la solution exacte et les différentes solutions obtenues

On voit que le développement de Taylor à l'ordre 3 n'est pas compétitif pour ce problème en comparaison avec notre méthode et celles de Runge-Kutta d'ordre 2 et 4.

Nous venons de voir le comportement des différentes méthodes pour un pas de temps fixé.
Il aurait été intéressant d'étudier l'évolution de l'erreur pour le cas de l'oscillateur harmonique :

- en fonction du pas de temps utilisé,

- en fonction de la précision des nombres flottants.

Enfin, la MDA a fait l'objet de dépôt logiciel auprès de l'Agence pour la Protection des Programmes sous le numéro de dépôt IDDN .FR .001 .160032 .000 .S .P .2022 .000 .31235 en date du 21 avril 2022. Le Centre National pour la Recherche Scientifique est l'organisme chargé de sa valorisation.

Cette deuxième partie étant terminée, la partie suivante présentera le premier exemple d'application de la MDA. Il illustrera son utilisation dans des contextes d'identification de paramètres et de génération de mouvement optimisé, contextes communs à la création d'un modèle satisfaisant pour l'étude du balancement des membres supérieurs dans la marche humanoïde.

III. La figure du passement filé, exemple d'application à l'étude du mouvement humain

Dans cette troisième partie sont présentés une figure de gymnastique aux barres parallèles et asymétriques, sa modélisation via la MDA et la synthèse d'un mouvement optimisé.

Les figures de gymnastique sont des mouvements particulièrement intéressants à étudier dans le cadre des travaux présentés dans ce document : les corps des gymnastes peuvent être modélisés comme des mécanismes arborescents ouverts, composés de plusieurs segments, soumis à d'importantes accélérations et qui doivent être suffisamment complexes pour représenter fidèlement la réalité. Il se trouve en effet que le modèle initialement utilisé pour l'étude s'était avéré inadapté en raison de sa relative simplicité imposée par Simulink/SimMechanics, le logiciel utilisé pour le créer et des grands temps de calculs que nous avions à l'époque pour générer un geste optimal sur ce logiciel (une vingtaine d'heures). Comme cela sera détaillé par la suite, modéliser la figure du passement filé demande une grande flexibilité dans sa description. Les mouvements aussi bien que les efforts sont considérés, nécessitant des calculs de dynamique direct et inverse. L'existence de données d'enregistrement de mouvements permet d'identifier certains paramètres du modèle et de valider les résultats fournis par la simulation. Un geste optimisé peut alors être synthétisé afin d'illustrer la capacité de la MDA et la rapidité d'évaluation des fichiers qu'elle génère.

1. Origine et objectifs du projet BE-MORE

Bien qu'il ne soit pas nécessaire de présenter le projet BE-MORE pour analyser l'exemple d'application auquel il est associé, cela apporte un contexte permettant de comprendre l'intérêt des fonctionnalités proposées par la MDA. Les textes suivants sont issus d'un document de communication du projet :

L'apprentissage d'un geste technique en gymnastique, tel que le « passe-file » représenté ci-dessous, requiert du temps et de nombreuses répétitions. Le(la) gymnaste se retrouve très souvent en situation d'échec, surtout en début d'apprentissage ; or, ces situations peuvent s'avérer risquées, voire traumatisantes.

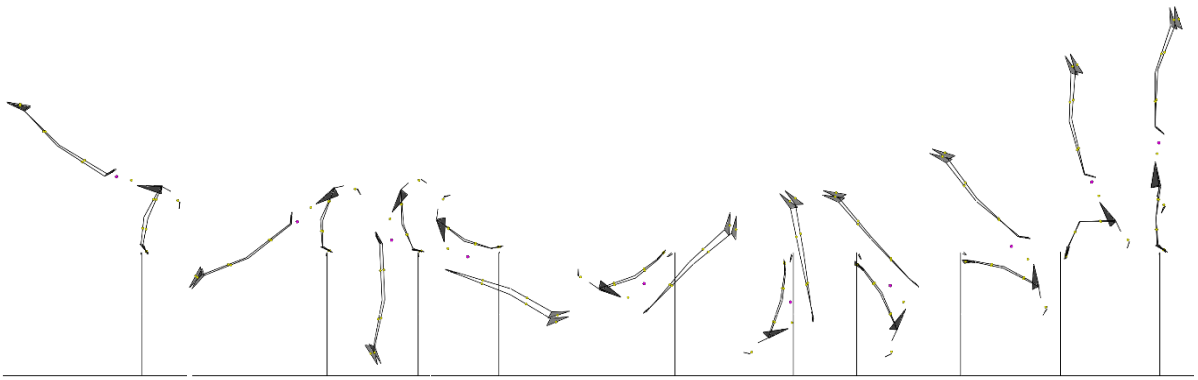


Figure 25. Décomposition du mouvement réalisé par une gymnaste lors de la figure du passement filé, enregistré optoélectroniquement

Le recours à un dispositif robotisé, schématisé ci-dessous, conçu spécifiquement pour préparer physiquement et techniquement le(la) gymnaste à réaliser la gestuelle articulaire optimale devrait permettre de réduire le nombre de tentatives en condition réelle pour réussir le geste et par conséquent réduire les risques de chutes et/ou de blessures. Un prototype du banc a été réalisé pour valider en laboratoire son efficacité sur un geste de musculation simple. Dans le cadre de ce projet, il conviendra d'adapter ce banc aux exigences du geste visé en termes de vitesse et d'effort à supporter.

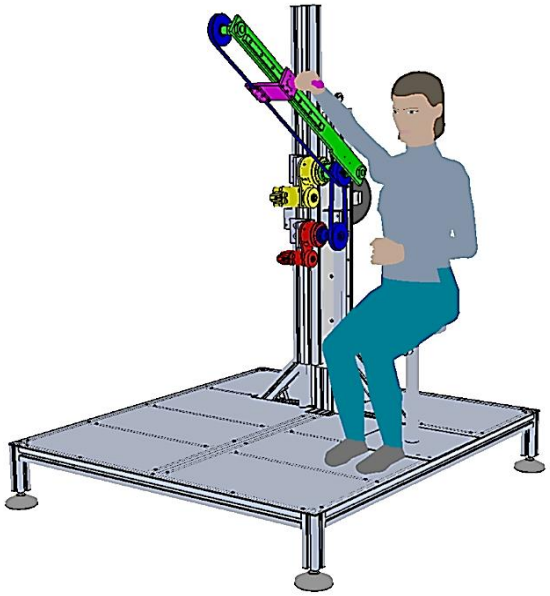


Figure 26. Image d'une des premières conceptions du banc - Brevet WO2022/084283 PCT/EP2021/078888

Afin de déterminer les vitesses et les efforts que le banc doit supporter, qui correspondent à la combinaison du mouvement et des efforts que le gymnaste devrait faire tout au long de la figure pour la réaliser de façon optimisée, il faut résoudre les équations issues de l'application du principe fondamental de la dynamique à un modèle du gymnaste. La justesse des vitesses et des efforts déterminés est directement liée à la fidélité du modèle utilisé vis-à-vis de la réalité. Or, les paramètres inertiels et géométriques des segments ne sont pas toujours simples à estimer. Bien que des tables anthropométriques existent, une part d'incertitude subsiste surtout dès lors que l'on s'intéresse à des sportifs avec des anthropométries parfois particulières. De plus, les actions mécaniques de liaison sont à calculer et dépendent à la fois des paramètres inertiels et du nombre de degrés de liberté envisagé à l'articulation. Il est donc nécessaire de valider ce modèle en l'utilisant pour calculer l'évolution en simulation d'un mouvement réalisé expérimentalement et de s'assurer que l'écart entre les données calculées et enregistrées soit suffisamment faible pour être négligé.

Un tel modèle requiert :

- une finesse permettant de tenir compte de la complexité de l'ensemble composé de la barre et du corps du gymnaste,

- une capacité de modification à la volée de ses paramètres pour une personnalisation rapide.

Comme expliqué dans la partie précédente, la MDA été conçue de sorte que les modèles qu'elle génère présentent ces caractéristiques.

Le dimensionnement du banc nécessite deux grandes étapes :

1. Identifier les valeurs numériques des paramètres méconnus ou inconnus du mécanisme modélisant le gymnaste pour que le mouvement simulé soit au plus proche du mouvement enregistré en

    a. faisant l'inventaire des données disponibles,

    b. décrivant le mouvement étudié,

    c. concevant un mécanisme permettant de modéliser le gymnaste « réel »,

    d. identifiant les valeurs des paramètres optimisant la fidélité du modèle.

2. Synthétiser un geste permettant au gymnaste modélisé de réaliser la figure du passement filé de manière optimisée en

    a. concevant un mécanisme permettant de modéliser un gymnaste respectant les critères d'évaluation de la figure,

    b. générant un geste optimisé minimisant les efforts nécessaires pour la réaliser

Le geste optimisé et les efforts qui en résultent sont ceux que le banc devra supporter afin de préparer le gymnaste physiquement et techniquement.

2. Identification des paramètres

a. Données disponibles

Les paramètres inertiels et géométriques initiaux de la gymnaste réalisant la figure du passement filé sont issus de tables anthropométriques (de Davidson et al. 2008 [47] et de de Leva 1996 [48]).

Nom du segment	**Masse (kg)**	**Moment d'inertie au centre d'inertie (kg.m²)**	**Longueur (m)**	**Position du centre d'inertie depuis la liaison du côté de la barre (m)**
Tête	3,565	0,020	0,210	0,081
Tronc	17,529	0,311	0,428	0,178
Bras	2*1,387	0,018	0,290	0,123
Avant-bras	2*1,188	0,007	0,206	0,112
Mains	2*0,396	0,001	0,075	0
Cuisses	2*4,952	0,234	0,419	0,151
Jambes	2*1,089	0,020	0,362	0,160
Pieds	2*0,990	0,005	0,175	0,035

Tableau 3. Fiche d'identité de la gymnaste

Le mouvement a été enregistré par un système optoélectronique Qualisys AB, composé de 16 caméras Oqus 7+, filmant 41 marqueurs sur la gymnaste et 12 sur la barre, cadencé à 100Hz et dont les données ont été interpolées à 1000Hz pour obtenir 1521 images.

N°	Position	N°	Position	N°	Position
1	Sommet de la tête	**16**	Triceps gauche	**31**	Tibia droit
2	Tempe droite	**17**	Épicondyle gauche	**32**	Malléole externe droite
3	Tempe gauche	**18**	Radius gauche	**33**	Malléole interne droite
4	C7	**19**	Ulna gauche	**34**	Métatarse du 5 droit
5	T10	**20**	Métacarpe du 2 gauche	**35**	Métatarse du 1 droit
6	Xiphoïde	**21**	Métacarpe du 5 gauche	**36**	Talon droit
7	Sternum	**22**	L5 droite	**37**	Quadriceps gauche
8	Acromion droit	**23**	L5 gauche	**38**	Condyle externe gauche
9	Triceps droit	**24**	Épine iliaque postéro-supérieure droite	**39**	Condyle interne gauche
10	Épicondyle droit	**25**	Épine iliaque postéro-supérieure gauche	**40**	Tibia gauche
11	Radius droit	**26**	Grand trochanter droit	**41**	Malléole externe gauche
12	Ulna droit	**27**	Grand trochanter gauche	**42**	Malléole interne gauche
13	Métacarpe du 2 droit	**28**	Quadriceps droit	**43**	Métatarse du 5 gauche
14	Métacarpe du 5 droit	**29**	Condyle externe droit	**44**	Métatarse du 1 gauche
15	Acromion gauche	**30**	Condyle interne droit	**45**	Talon gauche

Tableau 4. Tableau décrivant la position des marqueurs par numéro sur la gymnaste (en soulignés les marqueurs techniques retirés lors de l'enregistrement)

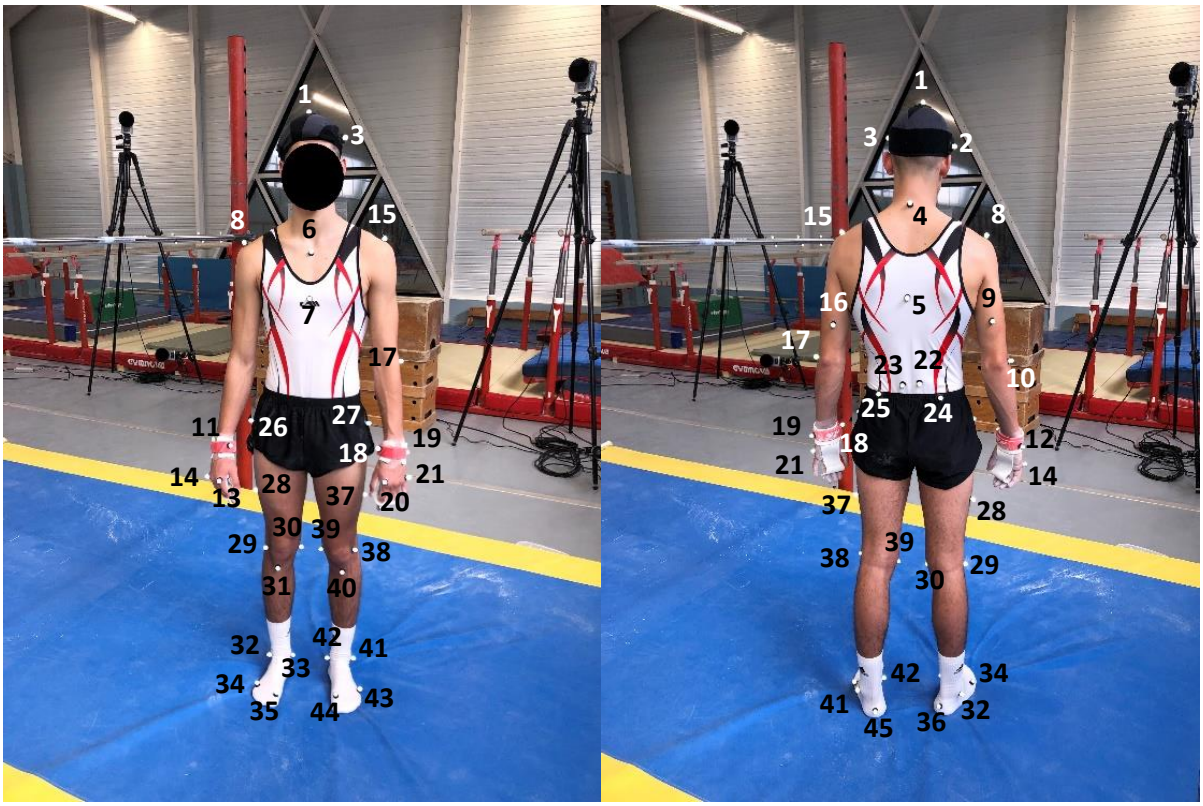


Figure 27. Photographies d'un gymnaste équipé de l'ensemble des marqueurs

La gymnaste dont le mouvement a été enregistré ne parvenait pas à garder ses membres supérieurs tendus sur l'intégralité du mouvement lors de l'enregistrement.

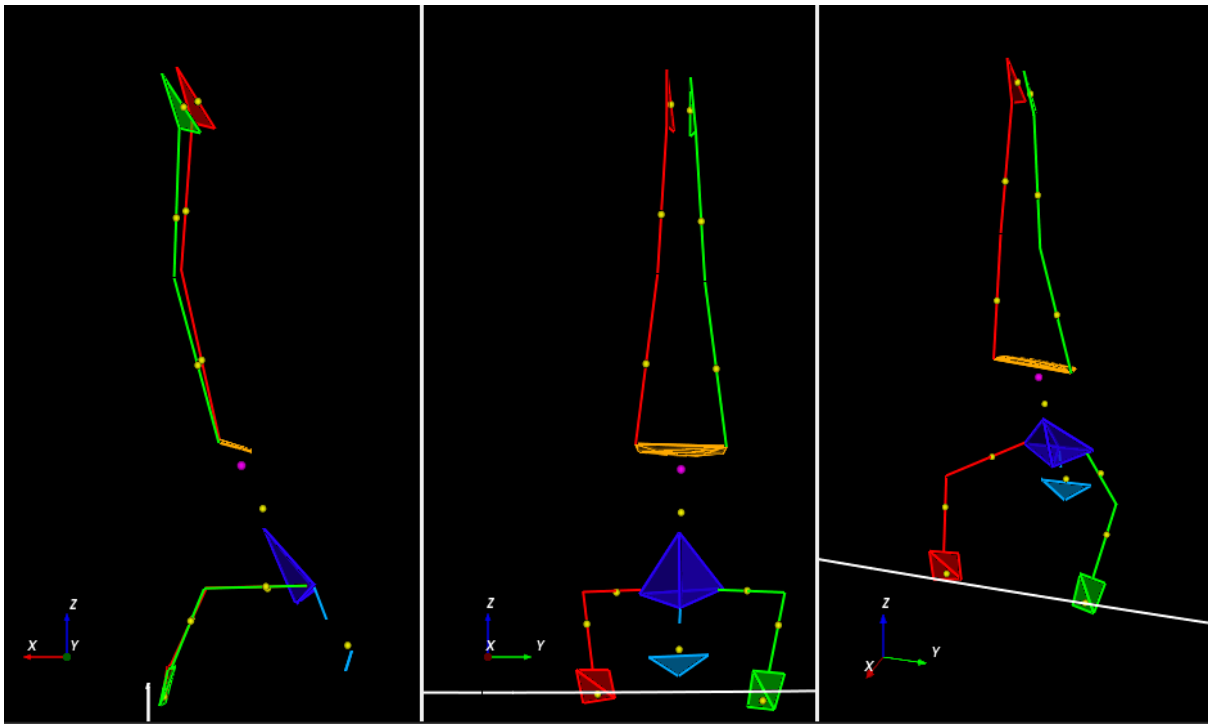


Figure 28. Capture d'écran illustrant les mouvements hors plan d'étude des membres supérieurs

Il a donc fallu intégrer au mécanisme la variation de la longueur apparente de ces segments dans le plan sagittal. Les longueurs des bras et des avant-bras, dans cette étude, dépendent donc du temps.

b. Description du mouvement étudié

La figure du passement filé débute par un appui tendu renversé. Tout au long de la figure, le gymnaste doit garder ses jambes dans l'alignement de son tronc ; seul le mouvement des membres supérieurs, tendus, par rapport au reste du corps est autorisé. Le gymnaste commence par laisser son corps basculer sous l'effet de la pesanteur tout en le ramenant vers ses bras. Lorsque son corps s'approche de la barre, le gymnaste doit tourner autour d'elle en restant au plus proche. Il doit ensuite revenir en position d'appui tendu renversé une fois passé de l'autre côté de la barre.

Il s'agit d'un mouvement majoritairement plan où la flexion/extension des bras est l'unique modification de posture que le gymnaste peut effectuer.

Obtenir la note maximale, c'est-à-dire respecter tous les critères d'évaluation lors de la réalisation de la figure, est extrêmement difficile. Une flexion importante au niveau des hanches est souvent observée par exemple (comme c'est le cas sur la décomposition du mouvement représenté sur la Figure 25). Il est nécessaire que le modèle incorpore ces imperfections pour être fidèle à la réalité.

c. Modélisation du gymnaste « réel » sous forme de mécanisme

De nombreuses versions de ce mécanisme ont été élaborées, chaque itération permettant d'obtenir un mouvement simulé plus proche de celui enregistré. La version montrée ici, inspirée des travaux de Hiley et Yeadon en 2003 [49], est composée de dix solides ayant chacun leurs propriétés inertielles (telles que la masse, la distribution de cette masse sur leur volume, la position du centre d'inertie…) et géométriques (telles que la position des points de liaisons, longueurs des solides…).

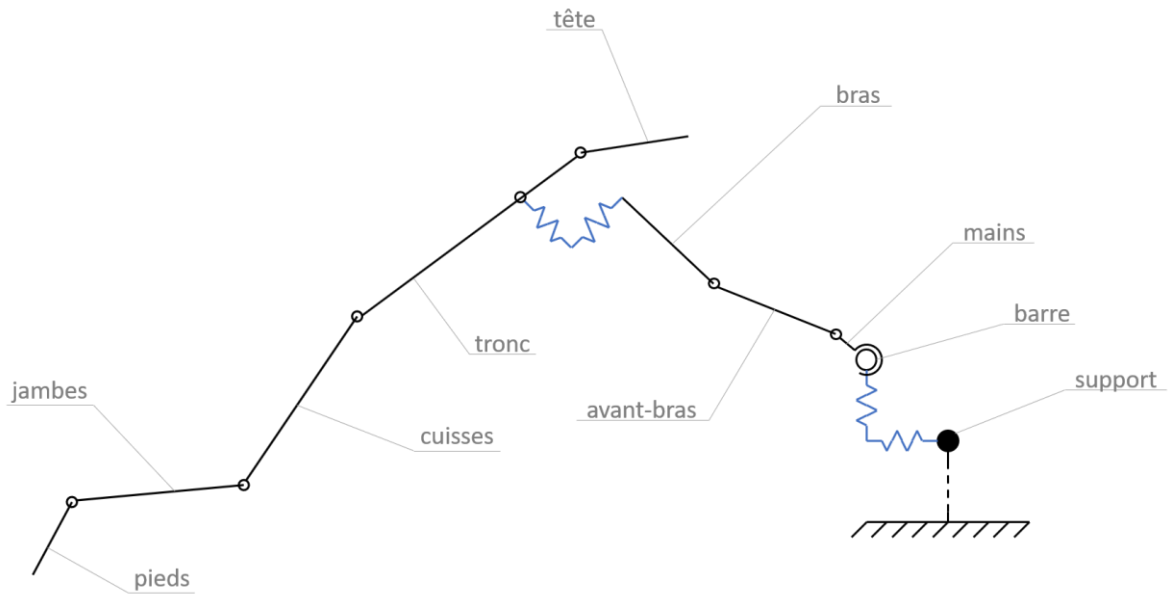


Figure 29. Mécanisme modélisant un gymnaste à la barre

Ce mécanisme prend en compte les rotations relatives entre les segments du gymnaste et autour de la barre. La déformation de la barre et le déplacement du bras par rapport au tronc sont représentés par des ressorts. Cela représente quatorze degrés de mobilité en tout dont neuf sont pilotés par le gymnaste et cinq sont déterminés par les lois de la dynamique.

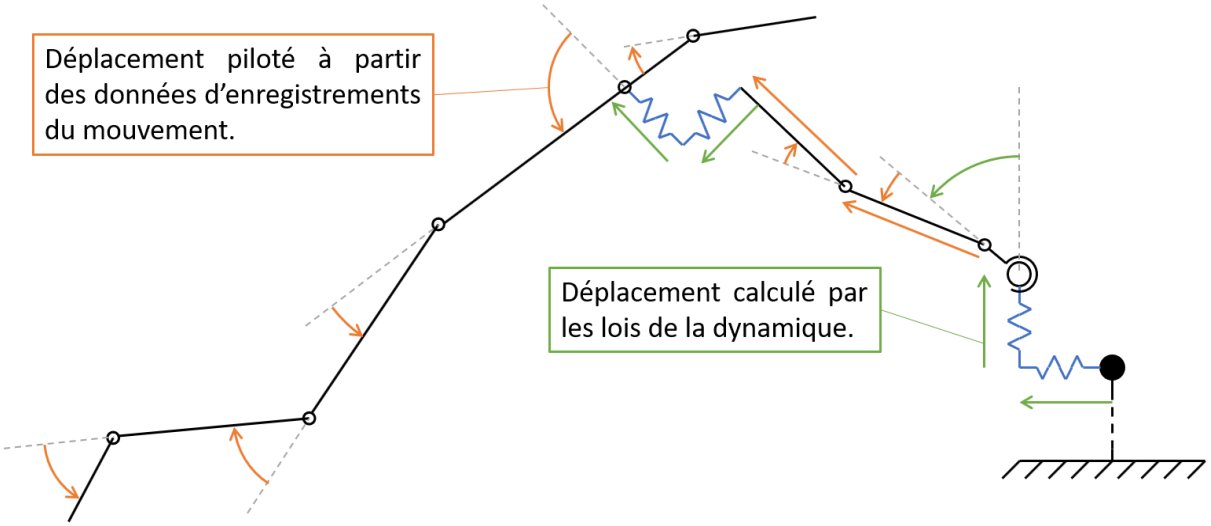


Figure 30. Illustration des mobilités du mécanisme précédent

Afin de pouvoir adapter le modèle, douze paramètres ont été retenus aux motifs que leurs valeurs numériques sont méconnues ou inconnues :

1. La répartition de la masse entre le tronc et les cuisses,
2. la position du centre d'inertie du tronc,
3. la position du centre d'inertie des cuisses pour ajuster les valeurs données par les tables anthropométriques pour les deux segments les plus lourds du corps humain.
4. La partie « en mouvement » de la masse de la barre.
5. La raideur « horizontale » de la barre.
6. L'amortissement « horizontal » de la barre.
7. La raideur « verticale » de la barre.
8. L'amortissement « vertical » de la barre.
9. La raideur « antéro-postérieure » de l'épaule.
10. L'amortissement « antéro-postérieur » de l'épaule.
11. La raideur « proximo-distale » de l'épaule.
12. L'amortissement « proximo-distal » de l'épaule.

Ce sont ces paramètres qui seront les variables d'optimisation de la prochaine étape. Afin de rendre le processus d'optimisation insensible à l'amplitude de leurs variations, elles seront normalisées.

Le tableau suivant nomme les deux fichiers générés par la MDA qui ont été utilisés pour l'identification des paramètres du modèle et donne quelques détails à leur sujet. Les fichiers utilisés pour cette étude sont disponibles en annexe.

Nom du fichier	Grandeurs calculées	Usage
GB12DDLLMSVRD_ GS1PBNDRD_ MD_DPS_CMJB_ PPO_Libres_ MATLAB.cpp	Positions et vitesses angulaires aux liaisons non pilotées de l'arborescence	Calcul du mouvement complet de l'arborescence en fonction des déplacements pilotés
GB12DDLLMSVRD_ GS1PBNDRD_ MD_APS_ CVD_PCLSetCI_R0_ MATLAB.cpp	Composantes des vecteurs position dans l'espace de points géométriques d'intérêt de l'arborescence	Affichage de la configuration de l'arborescence tout au long de la simulation

Tableau 5. Tableau résumant les fonctions utilisées pour l'identification des paramètres du modèle

d. Processus d'optimisation

Pour guider l'identification, la volonté que le mouvement simulé par le modèle soit au plus proche du mouvement enregistré a été traduite par la minimisation du critère f_1 :

$$f_1 = \sum_{i=1}^{1521} \left(\sum_{k=1}^{7} \left(\left(\left\| \overrightarrow{O_0 L_{k,EM}}(i) - \overrightarrow{O_0 L_{k,S}}(i) \right\| \right)^2 \right) \right)$$

Équation 70

où

- O_0 est l'origine du repère associé au support de la barre, considéré immobile dans le repère associé au système d'enregistrement du mouvement.
- *i* est le numéro d'un instant du mouvement.
- *k* est le numéro du marqueur du centre *L* d'une liaison du mécanisme modélisé :
    1. entre la barre et la main,
    2. entre la main et l'avant-bras,
    3. entre l'avant-bras et le bras,
    4. entre le tronc et la tête,
    5. entre le tronc et les cuisses,
    6. entre les cuisses et les jambes,
    7. entre les jambes et les pieds.
- *EM* signifie que le marqueur a été obtenu par enregistrement du mouvement.
- *S* signifie que le marqueur a été obtenu par simulation.

En minimisant f_1, l'algorithme d'optimisation diminue les différences entre les positions des marqueurs du gymnaste et celles des marqueurs de son modèle sur l'entièreté du mouvement en modifiant les valeurs des variables d'optimisation du modèle afin d'ajuster son comportement. La liaison entre les bras et le tronc n'a pas été retenue pour mesurer ces différences en raison de la difficulté d'obtenir la position de cette liaison dans les enregistrements de mouvements.

Le choix de l'algorithme pour l'identification s'est porté sur le Grey Wolf Optimizer (Mirjalili et al. 2014 [50]). Il s'agit d'une métaheuristique de recherche globale inspirée par la nature, imitant la méthode de chasse des loups gris et la hiérarchie sociale qui régit cette dernière. Basée sur l'emploi d'une population, elle appartient à la catégorie des stratégies utilisant l'intelligence collective pour explorer des vastes espaces de recherche, générés par exemple par un grand nombre de variables d'optimisation tel que c'est le cas dans cet exemple d'application. Voici les paramètres d'optimisation :

- Nombre de loups dans la meute : 200
- Nombre d'itérations : 50
- Bornes inférieures et supérieures pour les valeurs des paramètres du modèle à identifier :

Paramètre	**Valeur de la borne inférieure**	**Valeur de la borne supérieure**
Masse du tronc (kg)	5,259	29,800
Position du centre d'inertie du tronc (m)	0,049	0,277
Position du centre d'inertie des cuisses (m)	0,046	0,259
Masse en mouvement de la barre (kg)	0,1	1
Raideur horizontale de la barre (kg/s²)	5 000	50 000
Amortissement horizontal de la barre (kg/s)	10	20 000
Raideur verticale de la barre (kg/s²)	5 000	50 000
Amortissement vertical de la barre (kg/s)	10	20 000
Raideur antéro-postérieure de l'épaule (kg/s²)	15 000	50 000
Amortissement antéro-postérieur de l'épaule (kg/s)	10	50 000
Raideur proximo-distale de l'épaule (kg/s²)	15 000	50 000
Amortissement proximo-distal de l'épaule (kg/s)	10	50 000

Tableau 6. Valeurs des bornes inférieure et supérieure

Les écarts entre ces bornes ont volontairement été choisis très larges afin que l'identification ne soit pas limitée et puisse sélectionner les valeurs les plus appropriées.

Les conditions initiales sont définies pour correspondre à celle de l'enregistrement du mouvement. Ainsi les valeurs numériques des déplacements régis par les lois de la dynamique sont identiques au premier instant pour le mouvement réel et le mouvement simulé. À partir de là et à chaque itération, tous les loups réalisent chacun un cycle de trois étapes pour tout le mouvement simulé :

1. Le mouvement du modèle est calculé en fonction des valeurs prises par les paramètres à identifier lors de l'itération.

2. Les positions des sept marqueurs du modèle sont obtenues grâce aux déplacements calculés à l'étape précédente et aux déplacements pilotés.

3. Le cumul des carrés des différences de positions entre le mouvement enregistré et le mouvement simulé est calculé puis comparée à celles obtenues par d'autres loups pour définir les modifications à opérer sur les paramètres à identifier du modèle pour la prochaine itération.

La meute étant constituée de deux cents loups chassant pendant cinquante itérations, ce cycle est réalisé dix mille fois.

e. Résultats

Durée de l'optimisation : 39,77 minutes.
Durée moyenne d'un cycle : 0,237 seconde.

Paramètre	Valeur optimale	Position relative aux bornes
Masse du tronc (kg)	12,277	
Position du centre d'inertie du tronc (m)	0,194	
Position du centre d'inertie des cuisses (m)	0,124	
Masse en mouvement de la barre (kg)	0,675	
Raideur horizontale de la barre (kg/s²)	15 326	
Amortissement horizontal de la barre (kg/s)	910	
Raideur verticale de la barre (kg/s²)	31 983	
Amortissement vertical de la barre (kg/s)	593	
Raideur antéro-postérieure de l'épaule (kg/s²)	24 496	
Amortissement antéro-postérieur de l'épaule (kg/s)	3 956	
Raideur proximo-distale de l'épaule (kg/s²)	20 533	
Amortissement proximo-distal de l'épaule (kg/s)	1 057	

Tableau 7. Valeurs optimales des paramètres à identifier

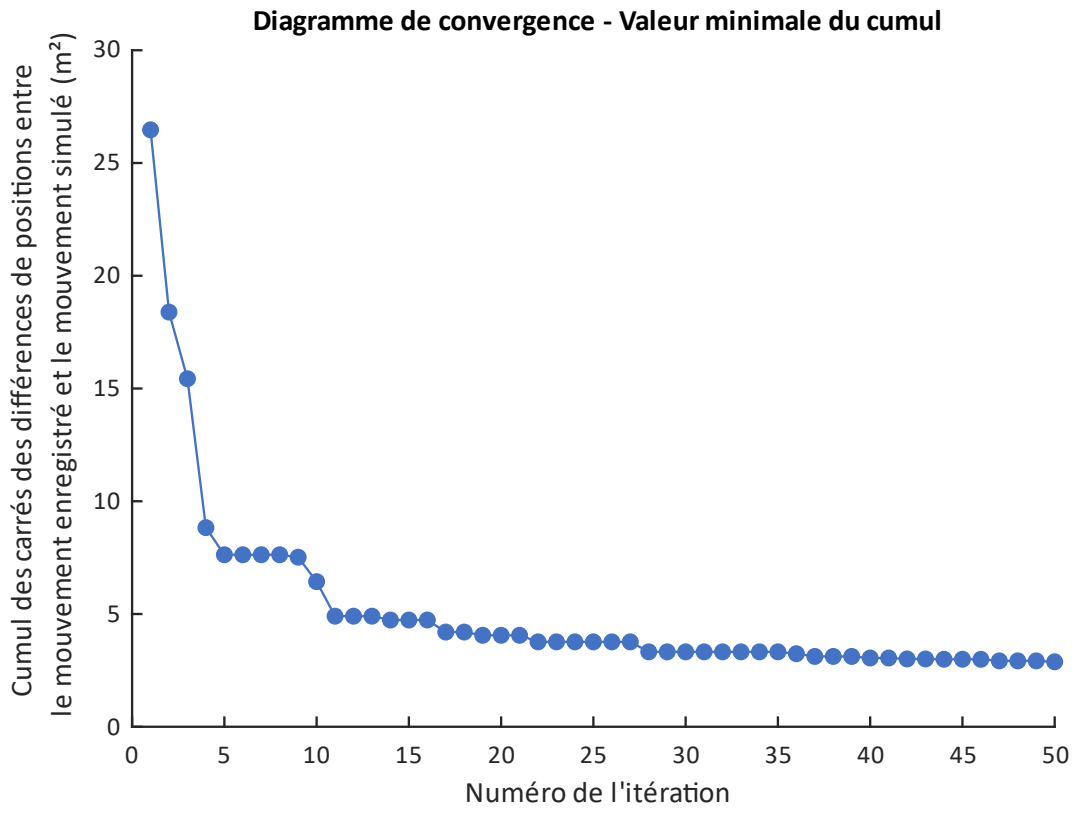


Figure 31

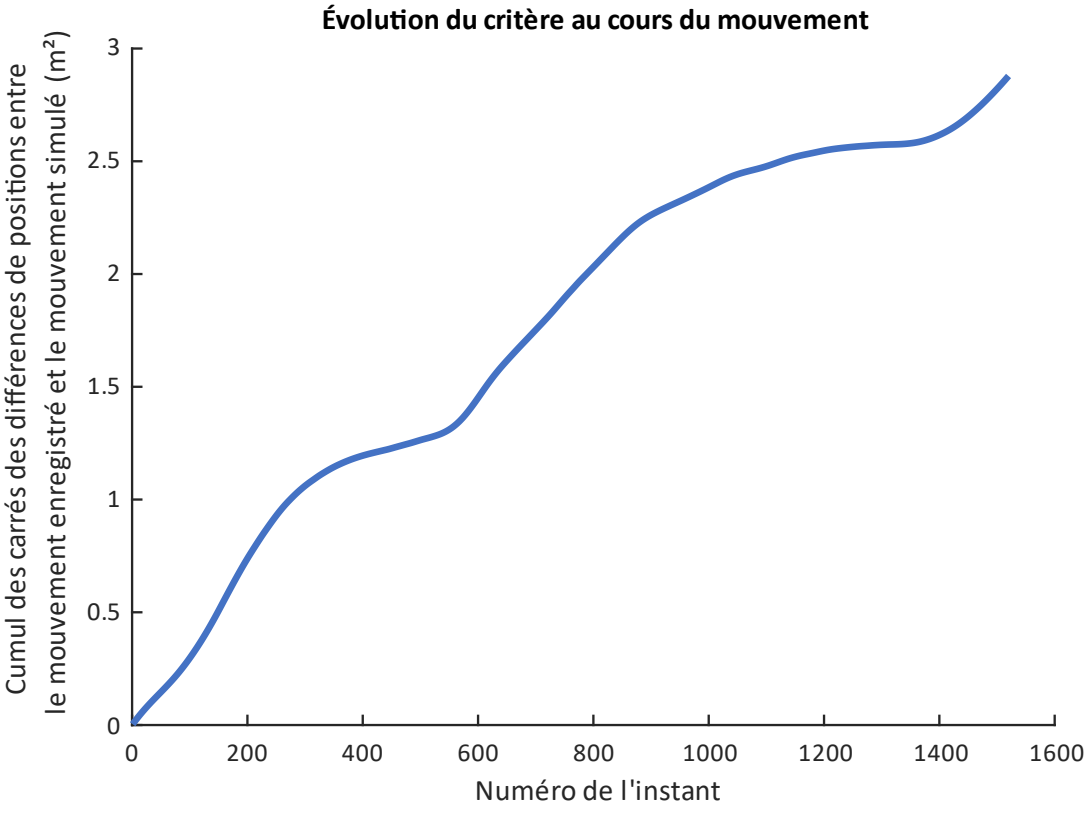


Figure 32

f. Discussion

Le cumul total des carrés des différences de positions entre le mouvement enregistré et le mouvement simulé vaut 2,87 mètres carrés. Cela signifie que chaque marqueur simulé suivi est en moyenne à 0,016 mètre, soit 1,6 centimètres, du marqueur enregistré correspondant à chaque instant. Cette valeur ainsi que celles de la raideur horizontale de la barre, de la raideur et de l'amortissement antéro-postérieurs de l'épaule, sont de l'ordre de grandeur de celles déterminées par Hiley et Yeadon en 2007 [51] pour la figure du grand tour arrière.

Les modifications apportées à la masse et à la position du centre d'inertie du tronc ainsi qu'à la position du centre d'inertie des cuisses peuvent être dues au fait qu'un seul individu ait été simulé. Cet individu, de sexe féminin, est par ailleurs singulier puisqu'athlétique et relativement jeune (14 ans). L'écart entre les propriétés mécaniques horizontale et verticale de la barre est peut-être lié aux solutions techniques de fixation de la barre à son support. Enfin, les différences entre les propriétés mécaniques antéro-postérieure et proximo-distale de l'épaule sont suffisamment faibles pour qu'un ressort amortisseur équivalent ait des propriétés similaires (tel que modélisé dans [51]). Il serait également possible de continuer la complexification du modèle, discuté en ouverture à la fin de cette partie.

3. Synthèse du geste optimisé

a. Modélisation du gymnaste « exemplaire » sous forme de mécanisme

Une fois les valeurs numériques des variables d'optimisation identifiées, un deuxième modèle, intégrant nativement ces valeurs, est utilisé pour déterminer un geste optimisé. Comme ce geste doit respecter les critères d'évaluation « artistique », le mécanisme représentant le gymnaste est beaucoup plus simple :

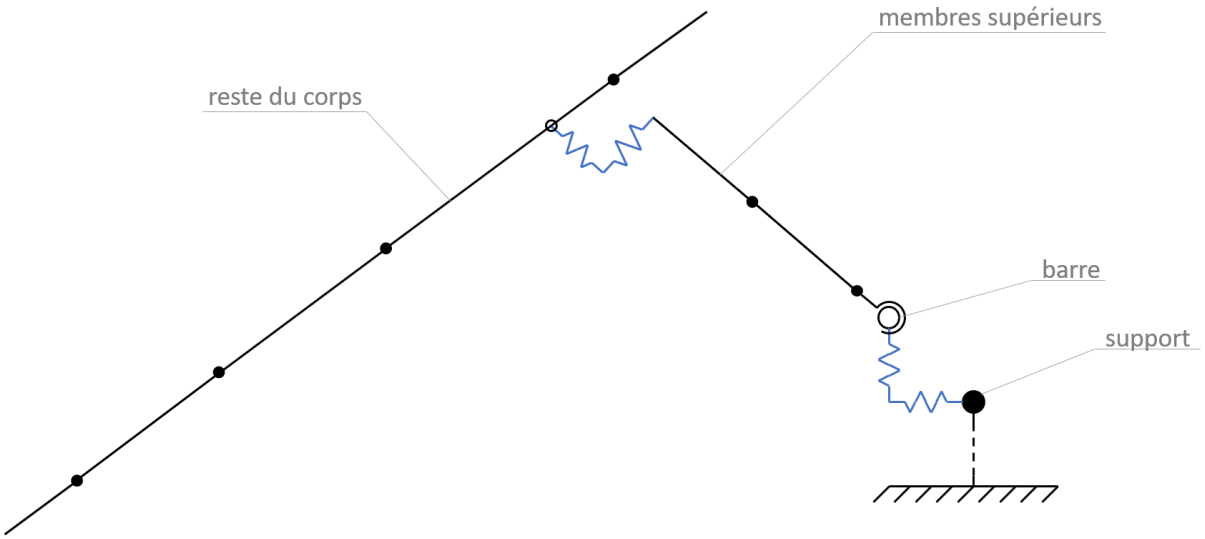


Figure 33. Mécanisme modélisant un gymnaste « exemplaire » à la barre

Composé de trois solides mobiles, ce mécanisme ne dispose que d'une seule variable que le gymnaste peut contrôler : l'angle de flexion/extension au niveau des épaules.

Pour générer un mouvement paramétrable de flexion aux épaules, deux polynômes d'ordre huit raccordés ont été utilisés (Williams 2013 [52]) : un pour la phase de flexion et un pour la phase d'extension.

$$\phi(0 \to t_{FlexionMax}) = a_8.t^8 + a_7.t^7 + a_6.t^6 + a_5.t^5 + a_4.t^4 + a_3.t^3 + a_2.t^2 + a_1.t + a_0$$

Équation 71

$$\phi(t_{FlexionMax} \to t_f) = b_8.t^8 + b_7.t^7 + b_6.t^6 + b_5.t^5 + b_4.t^4 + b_3.t^3 + b_2.t^2 + b_1.t + b_0$$

Équation 72

où $t_{FlexionMax}$ est le temps écoulé lors de la flexion maximale et t_f la durée totale du mouvement.

Chaque polynôme contient neuf inconnues. Afin de les déterminer, les valeurs aux bornes de l'intervalle de chaque polynôme et de ses dérivées première, deuxième, et troisième, sont choisies. Pour cette étude, mécanique, cela signifie imposer l'angle, la vitesse angulaire, l'accélération angulaire et l'à-coup angulaire de flexion aux épaules. Le procédé est le même pour les deux polynômes ainsi seul le cas du premier sera présenté.

$$\begin{cases} \phi(0) = 0 \\ \dot{\phi}(0) = 0 \\ \ddot{\phi}(0) = 0 \\ \dddot{\phi}(0) = 0 \end{cases} \quad et \quad \begin{cases} \phi(t_{FlexionMax}) = \phi_{Max} \\ \dot{\phi}(t_{FlexionMax}) = 0 \\ \ddot{\phi}(t_{FlexionMax}) = 0 \\ \dddot{\phi}(t_{FlexionMax}) = 0 \end{cases}$$

Équation 73

où ϕ_{Max} est l'angle de flexion maximal avec

$$\dot{\phi}(0 \to t_{FlexionMax}) = 8.a_8.t^7 + 7.a_7.t^6 + 6.a_6.t^5 + 5.a_5.t^4 + 4.a_4.t^3 + 3.a_3.t^2 + 2.a_2.t + a_1$$

Équation 74

$$\ddot{\phi}(0 \to t_{FlexionMax}) = 56.a_8.t^6 + 42.a_7.t^5 + 30.a_6.t^4 + 20.a_5.t^3 + 12.a_4.t^2 + 6.a_3.t + 2.a_2$$

Équation 75

$$\dddot{\phi}(0 \to t_{FlexionMax}) = 336.a_8.t^5 + 210.a_7.t^4 + 120.a_6.t^3 + 60.a_5.t^2 + 24.a_4.t + 6.a_3$$

Équation 76

Les conditions pour $t = 0$ permettent déjà de déterminer que a_0, a_1, a_2 et a_3 sont nuls en remplaçant t par 0 dans les expressions de ϕ et de ses dérivées. Pour obtenir les valeurs des cinq coefficients restants, il ne reste que quatre équations (celles pour $t = t_{FlexionMax}$). Il faut donc en ajouter une cinquième, dans le cas présent :

$$\phi\left(\frac{t_{FlexionMax}}{2}\right) = \frac{\phi_{Max}}{2}$$

Équation 77

La résolution de ce système linéaire de cinq équations permet d'exprimer les coefficients a_4, a_5, a_6, a_7 et a_8 en fonction de $t_{FlexionMax}$ et ϕ_{Max} pour le premier polynôme et ceux du deuxième polynôme en fonction de $t_{FlexionMax}$, ϕ_{Max} et t_f. Ainsi, le geste optimisé peut être obtenu en modifiant les valeurs numériques de ces trois variables qui seront donc les variables d'optimisation du processus de synthèse de mouvement nécessaire à l'obtention des données du geste à apprendre.

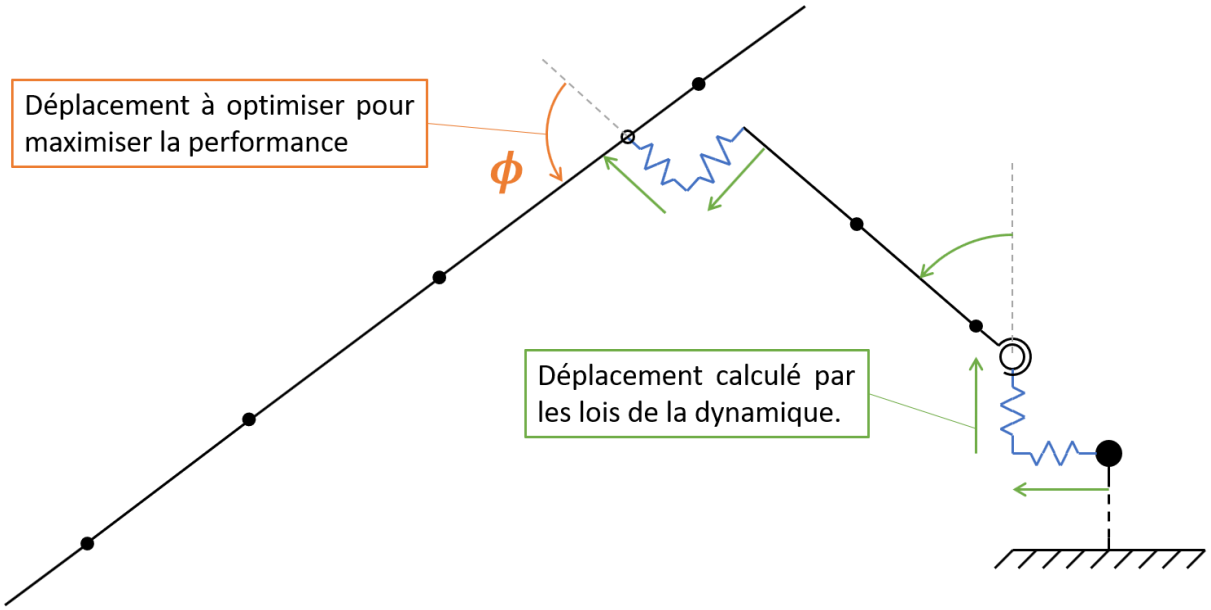


Figure 34. Illustration des mobilités du mécanisme précédent

Le tableau suivant nomme les deux fichiers générés par la MDA qui ont été utilisés pour la génération du geste optimisé et donne quelques détails à leur sujet. Les fonctions contenues dans ces fichiers, disponibles en annexe, servent des objectifs similaires à celles employées précédemment.

Nom du fichier	Grandeurs calculées	Usage
GB12DDLLMSVRD_ GS1R6DDL_ MD_CPS_CMJB_ PPO_Libres_Couple_Epaule_ MATLAB.cpp	Positions et vitesses angulaires aux liaisons non pilotées de l'arborescence et couple de flexion/extension aux épaules	Calcul du mouvement complet de l'arborescence en fonction des déplacements pilotés et de l'effort aux épaules fourni par le gymnaste
GB12DDLLMSVRD_ GS1R6DDL_ MD_APS_ CVD_PCLSetCI_R0_ MATLAB.cpp	Composantes des vecteurs position dans l'espace de points géométriques d'intérêt de l'arborescence	Affichage de la configuration de l'arborescence tout au long de la simulation

Tableau 8. Tableau résumant les fonctions utilisées pour la génération du geste optimisé

b. Processus d'optimisation

Puisque le banc se concentre sur la musculation de l'articulation de l'épaule et que l'un des objectifs du projet BE-MORE est de faciliter la réussite de la figure du passement filé, le critère à minimiser choisi pour cette deuxième optimisation est le suivant :

$$f_2 = \int_0^{t_f} |C(t)|.\, dt$$

Équation 78

où $|C(t)|$ est la valeur absolue du couple de flexion aux épaules lorsqu'un temps *t* s'est écoulé.
En minimisant f_2, l'algorithme d'optimisation diminue le couple aux épaules que la gymnaste doit fournir sur l'entièreté du mouvement en modifiant les valeurs des variables d'optimisation du modèle afin d'ajuster son comportement.

Le choix de l'algorithme pour la génération du geste s'est porté sur « fmincon » de l'*Optimization Toolbox™* développée par MathWorks® pour son logiciel MATLAB® [53]. « fmincon » est un solutionneur recherchant le minimum d'une fonction non-linéaire contrainte à plusieurs variables. C'est la possibilité de spécifier simplement des contraintes qui a motivé la sélection de ce solutionneur, dont voici le détail :

Contraintes de boîte		
Paramètre	**Valeur de la borne inférieure**	**Valeur de la borne supérieure**
Durée totale du mouvement t_f (s)	2,000	4,000
Temps écoulé lors de la flexion maximale $t_{FlexionMax}$ (s)	0,500	3,500
Angle de flexion maximal ϕ_{Max} (°)	120	160
Contraintes d'inégalité linéaire		
$t_f - t_{FlexionMax} \geq 0{,}5s$		
Contraintes d'inégalité non-linéaire		
$360° \leq \phi(t_f) \leq 362°$		

Tableau 9. Expressions des contraintes d'optimisation

Les options utilisées lors de l'invocation du solutionneur sont les suivantes :

```
% Options Fmincon
    options=optimoptions(@fmincon,...
          'Algorithm','sqp',...
          'Display','iter-detailed',...
          'MaxFunEvals',10000,...
          'MaxIter',50000,...
          'TolFun',1e-1,...
          'TolX',1e-9,...
          'TolCon',1e-6,...
          'OutputFcn','',...
          'PlotFcn','optimplotfvalconstr',...
          'GradObj','off');
```

Les conditions initiales sont définies pour correspondre à celle de l'appui tendu renversé, en équilibre instable. À partir de là et à chaque évaluation, « fmincon » effectue un cycle de deux étapes pour tout le mouvement simulé :

1. Le mouvement du modèle et le couple de flexion aux épaules est calculé en fonction des valeurs prises par les paramètres de génération du geste lors de l'itération.

2. L'intégrale de la valeur absolue du couple de flexion aux épaules est calculée puis comparée à celles obtenues par les autres évaluations de l'itération pour définir les modifications à opérer sur les paramètres de génération du geste pour la prochaine itération.

c. Résultats

Durée de l'optimisation : 23,82 secondes.
Nombre d'itération : 21.
Nombre d'évaluation de la fonction coût : 354.
Durée moyenne d'une évaluation : 0,067 seconde.

Paramètre	**Valeur optimale**	**Position relative aux bornes**
Durée totale du mouvement t_f (s)	2,405	
Temps écoulé lors de la flexion maximale $t_{FlexionMax}$ (s)	1,468	
Angle de flexion maximal ϕ_{Max} (°)	160	

Tableau 10. Valeurs optimales des paramètres de la synthèse du geste optimisé

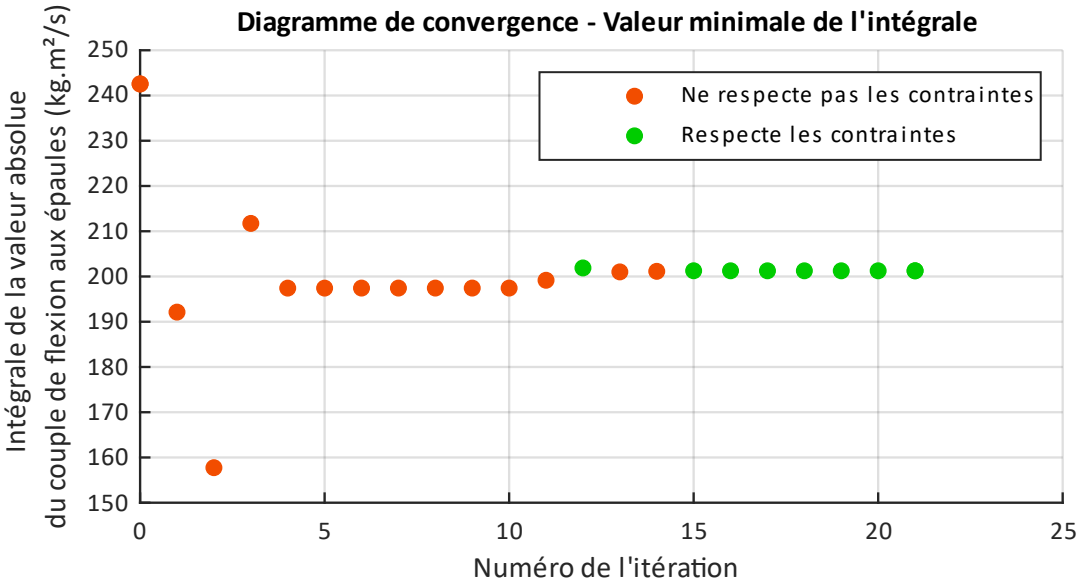


Figure 35

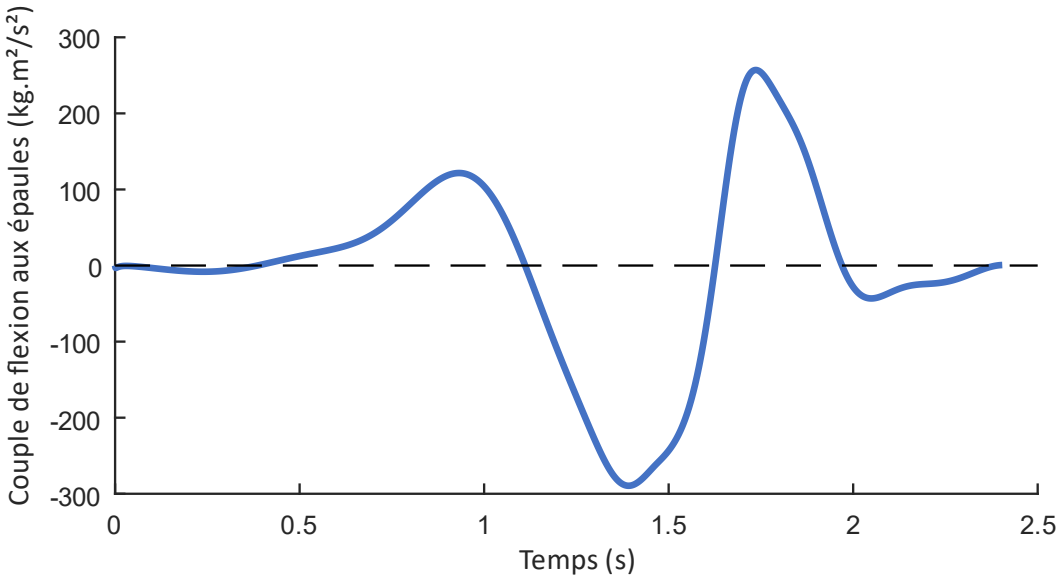


Figure 36

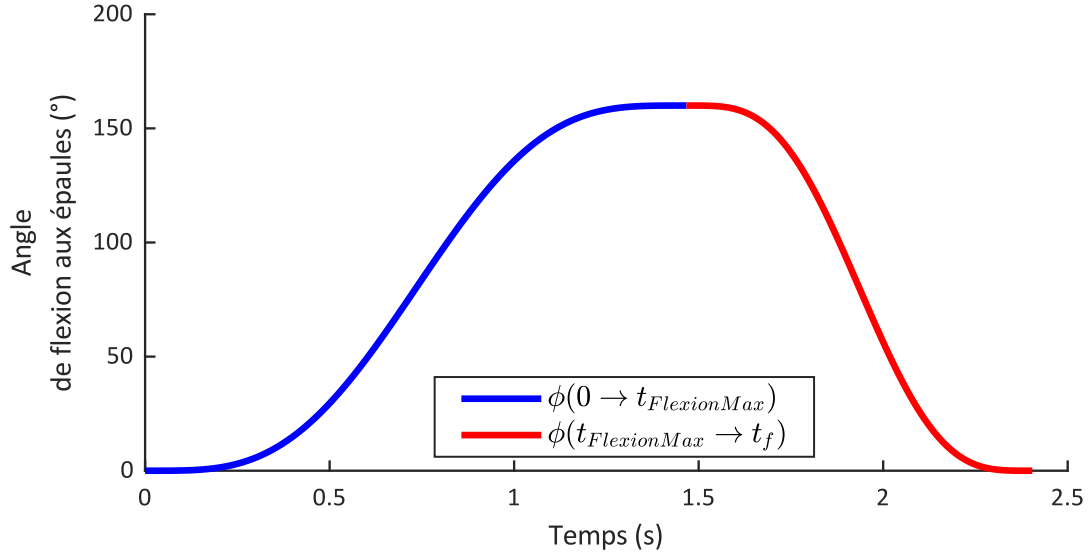


Figure 37

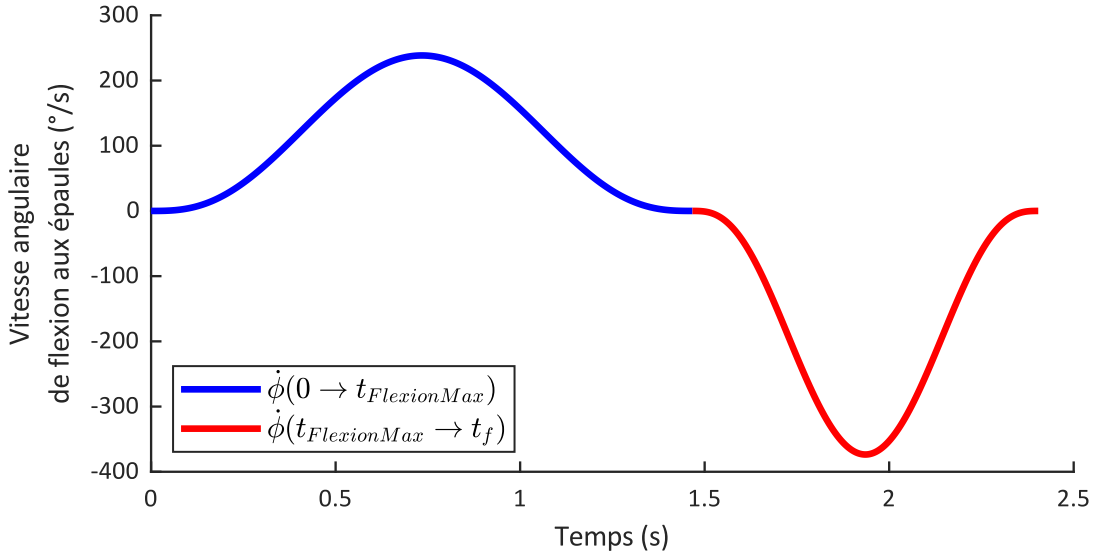


Figure 38

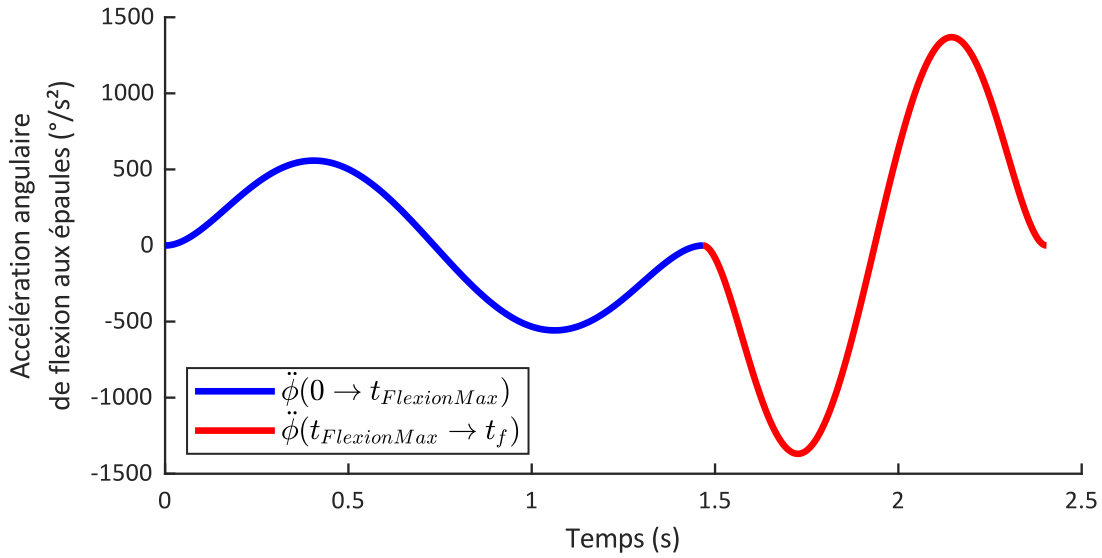


Figure 39

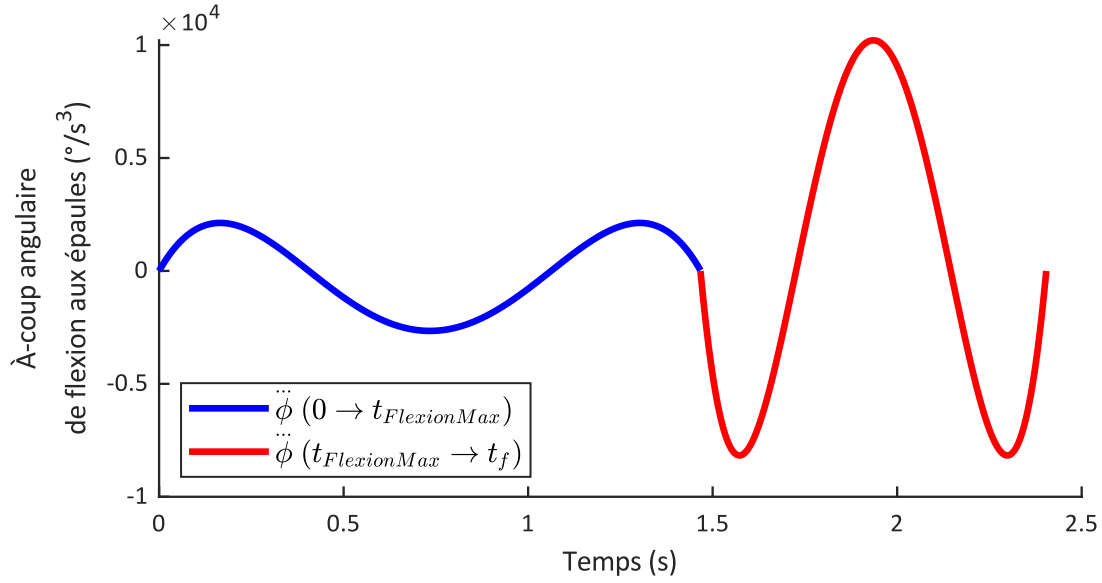


Figure 40

d. Discussion

La valeur du couple calculé par l'optimisation est celui que doivent fournir les deux épaules, qu'il faut donc diviser par deux pour obtenir les efforts que le gymnaste doit développer à chaque épaule. Une comparaison avec ceux exercés par la gymnaste lors de la figure enregistrée aurait permis de quantifier la diminution de l'effort nécessaire. Hélas, le mouvement enregistré ne respectant pas les critères d'évaluation, l'interprétation de cette comparaison aurait été faussée. La littérature sur la figure de passement filé est par ailleurs pauvre en données quantitatives sur les efforts (Petković et al. 2018 [54]).

Les graphiques de l'angle, la vitesse angulaire, l'accélération angulaire et l'à-coup angulaire de flexion aux épaules illustrent le geste généré par l'optimisation. Il est intéressant de noter la dérivabilité de l'accélération angulaire, témoignant du réalisme du mouvement.

En cumulant les temps d'identification et de synthèse, 40 minutes suffisent pour obtenir les données nécessaires au pilotage du banc. Si des données provenant de tables anthropomorphiques sont utilisées, cette durée est réduite à une trentaine de secondes. Ces temps rendent possible l'intégration du procédé tout entier lors d'une séance d'entraînement d'un gymnaste. La réduction du temps de synthèse du geste est particulièrement impressionnante par rapport à la vingtaine d'heures que nous avions précédemment avec Matlab Simulink/SimMechanics ce qui prouve l'efficacité de la MDA.

4. Perspective de développement

Bien que le mouvement simulé soit proche du mouvement réel, le mécanisme utilisé pour modéliser le gymnaste présente une faiblesse au niveau de l'épaule. Celle-ci provient des difficultés à obtenir la position de l'articulation dans l'espace. C'est un point mobile vis-à-vis du tronc qu'aucun ensemble de marqueurs ne permet d'obtenir directement. Qualifier et quantifier cette position est délicat et constitue un sujet de recherche en biomécanique.

La méthode utilisée dans cette étude est de faire réaliser des mouvements dits fonctionnels aux gymnastes et de déterminer les positions de deux points fixes, un pour le tronc et un pour les bras, permettant de décrire le mouvement d'un segment par rapport à l'autre comme une rotation en minimisant l'erreur (Lura et al. 2012 [55]). C'est cette méthode qui a permis de définir les points d'attache des ressorts de l'épaule. Son principal inconvénient est justement qu'elle minimise les mouvements de translation entre le tronc et les bras, mouvements que le modèle est justement sensé permettre.

Une autre méthode serait de faire réaliser des mouvements de flexion/extension aux gymnastes en calculant les positions successives des axes instantanés de rotation (AIR) des bras par rapport au tronc. Il faudrait ensuite déterminer une interpolation permettant de retrouver la position des AIR dans le repère du tronc en fonction de l'angle de flexion/extension. L'articulation de l'épaule, appartenant à l'AIR, sera donc mobile dans le repère du tronc. La position des coudes provenant de l'enregistrement du mouvement, cette méthode induit une variation de longueur des bras. Un ressort dans le prolongement des bras permettra de modéliser cette variation tout en conservant l'hypothèse des solides rigides.

La MDA a donc permis de simuler des mécanismes dont la modélisation exige une importante flexibilité, aussi bien dans leurs descriptions que dans leurs fonctionnements, et ce sans nécessiter de long temps de calcul.

Cette troisième partie étant terminée, la partie suivante présentera le second exemple d'application de la MDA. Il illustrera son utilisation dans des contextes de modélisation de mécanismes aux

paramètres inconnus et de calcul des grandeurs nécessaires à la qualification et à la quantification de la marche.

IV. La bipédie d'ORHRO, exemple d'application à la robotique

Dans cette quatrième partie sont présentés la marche d'un robot humanoïde, sa modélisation via la MDA et le dimensionnement de membres supérieurs optimisés.

En plus des raisons mentionnées précédemment, la MDA fut développée dans l'optique d'assister le développement d'Open Robotics Humanoid RObot (ORHRO). Le schéma suivant illustre la façon dont elle s'intègre dans ce dernier :

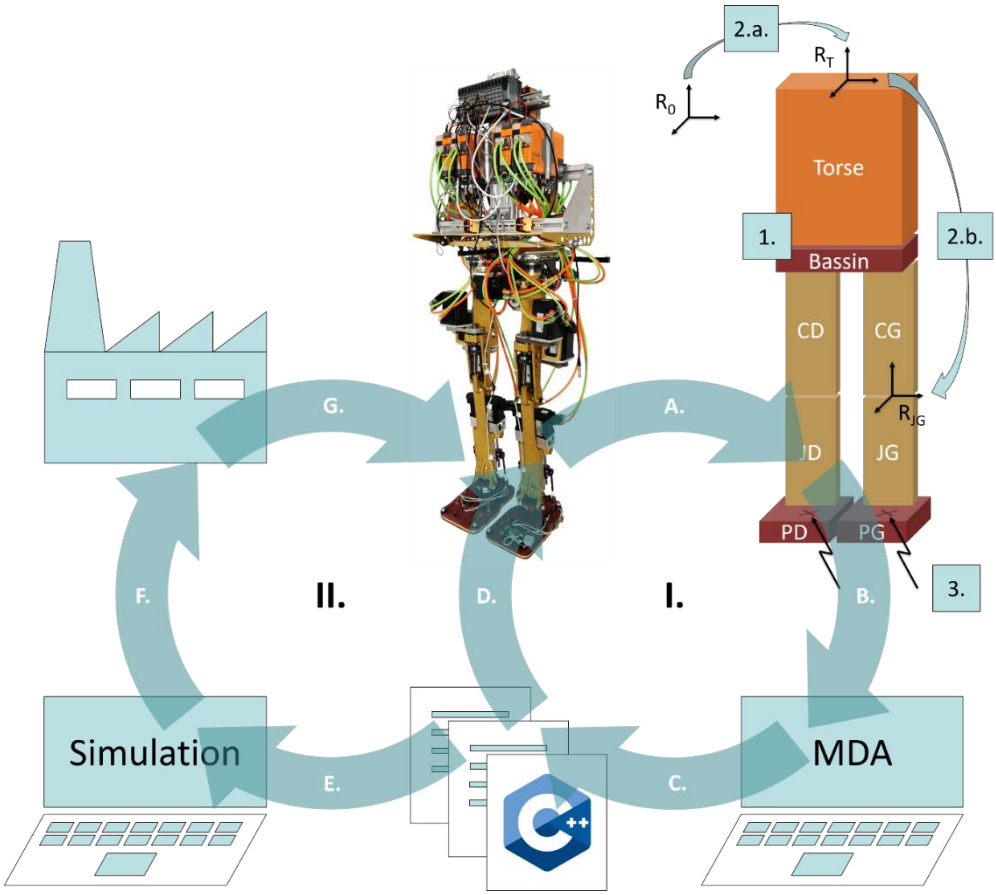


Figure 41. Schéma de l'utilisation de la MDA dans le cas du développement d'ORHRO

Il est composé de deux boucles, A-B-C-D et A-B-C-E-F-G, qui peuvent s'appliquer au développement de nombreux systèmes mécatroniques et dont le détail est donné ci-après.

I. Modélisation et commande

A. Description du système étudié sous forme d'arborescence.

1. Création des segments en renseignant les masses, matrices d'inertie, centres d'inertie, etc.

2. Positionnement et orientation…

a. d'au moins un segment par rapport au milieu extérieur,

b. des segments les uns par rapport aux autres.

3. Renseignement des potentielles actions mécaniques extérieures ponctuelles.

B. Écriture du fichier contenant la description précédente et utilisé en entrée par la MDA. Cette description peut inclure des modifications et ajouts (cf. « II. E. »).

C. Exécution de la MDA et obtention des fichiers de calcul.

D. Utilisation des fichiers pour la commande.

II. Recherche et développement

E. Utilisation des fichiers en simulation afin d'étudier les conséquences de l'ajout de nouveaux solides au système initial et d'en déterminer les paramètres de conception optimaux.

F. Conception mécatronique et fabrication des nouveaux solides.

G. Intégration des nouveaux solides et essais sur le nouveau système dont la commande est déjà prise en charge (cf. « I. B. » et « I. D. »). La comparaison entre la simulation et l'expérience peut alors commencer.

C'est l'étape « II. E. » qui est présentée ici. Comme mentionné dans la première partie, des questions subsistent sur les mécanismes générant le balancement des bras dans la marche humaine. Cela est dû à la complexité de l'identification des paramètres nécessaires à la modélisation du corps humain et l'étude de la bipédie d'un robot humanoïde, plus simple, pourrait y apporter des éléments de réponse. Par ailleurs, une optimisation dimensionnelle des bras des robots humanoïdes permettrait d'améliorer la performance dynamique de leur marche et constituerait une aide à la conception.

1. Présentation de l'existant

Les travaux exposés ici se basent sur les moyens et l'expérience de l'équipe Robotique Biomécanique Sport Santé (RoBioSS). Cette dernière dispose d'ORHRO (présenté dans l'article de Eon et al. [56], visible sur le schéma ci-dessus), un robot humanoïde mesurant 1,67 mètres et pesant 76 kilogrammes. Capable de marcher sans bras, ORHRO constituait un excellent support d'étude : tous les codes permettant de générer des trajectoires de marches stables, à différentes allures, étaient disponibles et validés expérimentalement pour évaluer les conséquences de l'ajout de membres supérieurs et de leur balancement.

Le générateur de trajectoire utilisé a été développé lors de la thèse de Eon en 2009 [57], ce dernier s'inspire des travaux de Kajita et al. (2001 [58]). Les données d'entrée sont les paramètres de marche suivant :

- durée d'un temps de cycle de marche,
- répartition double-appui/simple-appui,
- longueur du pas,
- hauteur de levée de pied,
- flexion initiale aux genoux imposant celles aux hanches et aux chevilles,
- localisation du Point d'Annulation du Moment de Basculement (PAMB, défini par la suite) dans le pied d'appui.

Les trajectoires de références du PAMB à suivre pour le robot ainsi que les lois de levée de pied sont calculées automatiquement à partir de ces paramètres en utilisant des fonctions polynomiales par morceaux d'ordre cinq avec raccordement aux ordres zéro, un et deux. L'ensemble de la dynamique du robot est alors approché par une masse ponctuelle centrée au centre d'inertie de la chaîne polyarticulée (ce modèle est connu sous le nom de Linear Inverted Pendulum Mode ou LIPM).

Sous l'hypothèse d'altitude constante du centre d'inertie et pour une marche sur un sol plat horizontal, on obtient deux équations (une pour l'axe antéro-postérieur, une pour l'axe transversal), de forme identique, permettant de lier la position du PAMB à la position du centre d'inertie et à sa dérivée seconde.

Ce modèle simpliste permet via un modèle de commande prédictive de calculer la trajectoire du centre d'inertie du LIPM pour que son PAMB suive une référence spécifiée. L'obtention de la trajectoire de ce centre d'inertie permet alors, via la résolution d'un modèle géométrique inverse de la chaîne polyarticulée, de calculer le mouvement du robot pour placer correctement son centre d'inertie. Il est évident que la dynamique du système multicorps complet n'est pas conforme à celle du LIPM et que des écarts de modèles existent.

Ainsi une phase de correction est nécessaire pour adapter la trajectoire de référence du PAMB à suivre afin de réduire ces écarts. Ce correcteur utilise l'erreur de localisation entre le PAMB du robot complet et celui du LIPM. Au global, le générateur fonctionne selon les étapes suivantes :

1. Génération des trajectoires de références (pieds, PAMB),
2. Modèle de commande prédictive (obtention du centre d'inertie du LIPM),
3. Résolution du MGI pour l'ensemble de la trajectoire (obtention du mouvement),
4. Calculs des écarts de modèles pour le PAMB LIPM/robot complet (correction de l'écart),
5. Génération de la référence corrigée du PAMB,
6. Modèle de commande prédictive (obtention du centre d'inertie du LIPM),
7. Résolution du MGI pour l'ensemble de la trajectoire (obtention du mouvement),
8. Calculs des écarts de modèles pour le PAMB LIPM/robot complet,
9. Validation de la trajectoire (chute, frottement, actionneur).

Cette méthode a été testée avec succès sur le robot ORHRO avec les paramètres suivants et a été filmée [59] :

Paramètre	**Valeur**
Temps de cycle (s)	5
Répartition du double-appui (% du cycle)	2 x 25%
Répartition du simple-appui (% du cycle)	2 x 25%
Longueur du pas (m)	0,35
Hauteur de levée de pied (m)	0,05
Flexion initiale aux genoux (°)	20
Position médio-latérale du PAMB par rapport au bord intérieur du pied d'appui (m)	0,09
Position antéro-postérieure du PAMB par rapport au cardan de cheville du pied d'appui (m)	-0/+0

Tableau 11. Valeurs des paramètres de marche de la vidéo

2. Ce qui est mesuré

Deux conditions de stabilité, utilisés dans de précédents travaux de l'équipe RoBioSS [57], ont été retenues pour cet exemple d'application :

- Condition de non-glissement via la vérification que la valeur maximale nécessaire du coefficient de frottement entre les pieds du robot et le sol pour une marche donnée est inférieure au coefficient de frottement réel.

- Condition de non-basculement via la vérification de l'appartenance du point d'annulation du moment de basculement au polygone de sustentation.

Ces deux conditions sont liées à l'action mécanique exercée par le sol sur les pieds. Décrite sous la forme de torseur, cette dernière peut être écrite ainsi :

$$T_{Sol \to ORHRO} = \begin{Bmatrix} X & L \\ Y & M \\ Z & N \end{Bmatrix}_{O_0, R_0}$$

Équation 79

où O_0 est l'origine du repère R_0 associé au milieu extérieur à ORHRO et l'axe $\vec{x}_0$ est normal au sol.

La minimisation du coefficient de frottement maximal nécessaire a servi de critère à l'optimisation des paramètres inertiels et géométriques des membres supérieurs. Afin de considérer une valeur sans unité et dont l'ordre de grandeur corresponde à un coefficient de frottement, l'hypothèse simplificatrice a été faite que le couple de résistance au pivotement entre les pieds d'ORHRO et le sol puisse être formulée de manière analogue à celui d'un embrayage où les semelles des pieds joueraient le rôle du disque de friction. En effet, le principe est similaire car il s'agit de pouvoir transmettre des efforts en rotation par adhérence donc sans glissement. Ce couple s'écrit ainsi :

$$L = \frac{2}{3} \times r \times \mu \times X$$

Équation 80

où

- *r* est le rayon d'un disque d'une surface égale à celle d'une semelle,
- μ est le coefficient de frottement entre les pieds et le sol.

Ce qui permet de calculer μ en reformulant :

$$\mu = \frac{3 \times L}{2 \times r \times X}$$

Équation 81

Le transport du torseur précédent au point d'annulation du moment de basculement (PAMB), de coordonnées $(x_{PAMB}, y_{PAMB}, z_{PAMB})$, donne l'expression suivante :

$$T_{Sol \to ORHRO} = \begin{Bmatrix} X & L & -x_{PAMB} & X \\ Y & M + -y_{PAMB} \wedge Y \\ Z & N & -z_{PAMB} & Z \end{Bmatrix}_{PAMB,R_0}$$

Équation 82

Or, le point d'annulation du moment de basculement est, par définition, le point d'expression du torseur précédent où son moment est selon $\vec{x}_0$ uniquement, les moments de basculement étant nuls. De fait,

$$\begin{cases} M - z_{PAMB} \times X + x_{PAMB} \times Z = 0 \\ N - x_{PAMB} \times Y + y_{PAMB} \times X = 0 \end{cases}$$

Équation 83

ORHRO marchant sur un sol horizontal et le PAMB appartenant à ce dernier (Sardain et al. 2004 [60]), x_{PAMB} est une constante dont la valeur est l'altitude du sol dans R_0. Il est donc possible de calculer les deux dernières coordonnées du PAMB avec le système d'équation suivant :

$$\begin{cases} y_{PAMB} = \dfrac{-N + x_{PAMB} \times Y}{X} \\ z_{PAMB} = \dfrac{M + x_{PAMB} \times Z}{X} \end{cases}$$

Équation 84

La position du PAMB calculée, il est alors possible de vérifier qu'il ne sorte pas du polygone de sustentation et donc que l'ajout des bras optimisés et leurs balancements ne font pas basculer ORHRO.

3. Modélisation du robot actuel et des membres supérieurs virtuels

Pour des raisons de flexibilité et d'efficacité, ORHRO a été décomposé en trois arborescences : une pour le tronc et les membres inférieurs, une pour le membre supérieur gauche et une pour le membre supérieur droit. Le tronc et les membres inférieurs existant déjà, seule l'architecture mécanique des membres supérieurs restait à définir. Pour une première modélisation, celle d'un double pendule permettait de reproduire les mouvements des membres supérieurs les plus amples dans la marche de l'être humain tout en ne nécessitant qu'un faible nombre de variables.

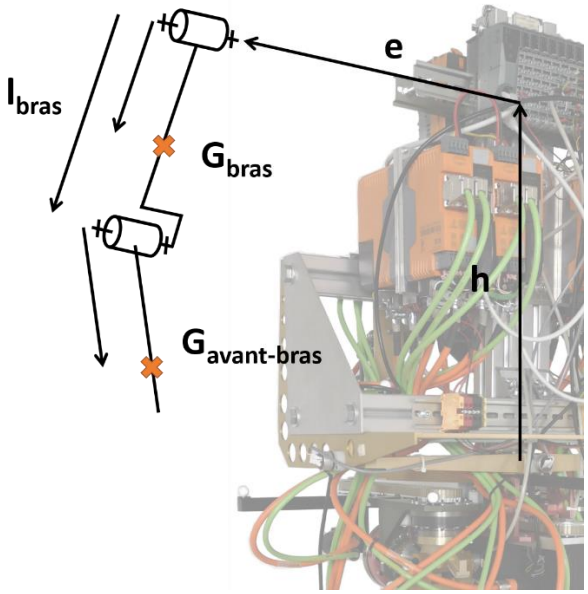


Figure 42. Schéma représentant le membre supérieur droit et ses principaux paramètres dimensionnants

Les paramètres inertiels et géométriques qui seront optimisés sont les suivants :

- Hauteur des épaules.
- Écartement des épaules.
- Masse du bras.
- Position du centre d'inertie du bras.
- Longueur du bras.
- Masse de l'avant-bras.
- Position du centre d'inertie de l'avant-bras.

Les moments et produits d'inertie ont été annulés dans cet exemple afin de diminuer le nombre de paramètres à optimiser, tout comme le fait que les centres d'inertie des membres supérieurs soient positionnés longitudinalement a permis d'exprimer leur position à l'aide d'une seule coordonnée. Ils pourraient néanmoins être intégrés au processus sans difficultés supplémentaires, étant renseignés de la même manière à la MDA que les autres paramètres.

Chacune de ces arborescences a permis de générer plusieurs modèles aux fonctions diverses que le tableau suivant résume. Seul le membre supérieur gauche est détaillé, les fichiers pour le membre supérieur droit étant similaires. Les fichiers utilisés pour cette étude sont disponibles en annexe.

Arborescence	Nom du fichier	Grandeurs calculées	Usage
Tronc et membres inférieurs	ORHROV1p5TetMI_ PBSR2DDL_ MD_APS_ TAMGAMPI_ MATLAB.cpp	Composantes du torseur de l'action mécanique exercée par le sol sur l'arborescence	Calcul du coefficient de frottement maximal nécessaire et de la position du PAMB
	ORHROV1p5TetMI_ PBSR2DDL_ MD_APS_ CVD_PLSetI_R0_ MATLAB.cpp	Composantes des vecteurs position dans l'espace de points géométriques d'intérêt de l'arborescence	Affichage de la configuration de l'arborescence tout au long des simulations
Membre supérieur gauche	ORHROV1p5MSG_ PBSR2DDL_ MD_DPS_CMJB_ MouvementBrasGauche_ MATLAB.cpp	Positions et vitesses angulaires aux articulations de l'épaule et du coude dans le cas « bras ballants »	Calcul du mouvement du membre supérieur dans le cas « bras ballants »
	ORHROV1p5MSG_ PBSR2DDL_ MD_DPS_CE_ AccArticulationsBrasGauche_ MATLAB.cpp	Accélération angulaire aux articulations de l'épaule et du coude	Calcul des accélérations nécessaires à la détermination des actions mécaniques à l'épaule
	ORHROV1p5MSG_ PBSR2DDL_ MD_IPS_ TAML_EpauleGauche_ MATLAB.cpp	Composantes du torseur des actions mécaniques de liaison exercées par le membre supérieur sur le tronc	Connexion mécanique du membre supérieur au tronc
	ORHROV1p5MSG_ PBSR2DDL_ MD_APS_ CVD_PLSetI_R0_ MATLAB.cpp	Composantes des vecteurs position dans l'espace de points géométriques d'intérêt de l'arborescence	Affichage de la configuration de l'arborescence tout au long des simulations

Tableau 12. Tableau résumant les fonctions utilisées dans cette étude

4. Cas étudiés

Pour montrer l'impact des membres supérieurs dans la bipédie d'ORHRO, quatre types de balancement des bras ont été testés pour trois vitesses de marche.

Il existe une infinité de mouvements que les membres supérieurs peuvent réaliser au cours de la marche. Les quatre étudiés ici sont parmi les plus simples à mettre en œuvre afin de conserver les paramètres inertiels et géométriques des membres supérieurs en tant que principales variables d'optimisation :

1. Passif avec l'articulation du coude bloquée.
2. Passif avec l'articulation du coude libre.
3. Actif commandé en proportion de l'angle de flexion de la hanche.
4. Actif commandé en proportion de l'angle de lacet du tronc.

La vitesse de la marche influençant l'amplitude du balancement des bras chez l'humain ([6] et [21]), son impact doit donc être évalué pour les robots humanoïdes. Les marches sont toutes composées d'un pas, de deux foulées puis d'un pas. La fréquence d'échantillonnage a été fixée à 200 Hertz. Le tableau suivant regroupe les valeurs des paramètres de génération de marche ayant été utilisés pour les obtenir.

Durée du mouvement entier simulé (s)	12,22	10,10	9,04
Vitesse de marche (m/s)	0,1	0,2	0,3
Paramètre	**Valeur**		
Temps de cycle (s)	3,5	2,5	2
Répartition du double-appui (% du cycle)	2 x 12%	2 x 12%	2 x 12%
Répartition du simple-appui (% du cycle)	2 x 38%	2 x 38%	2 x 38%
Longueur du pas (m)	0,35	0,5	0,6
Hauteur de levée de pied (m)	0,03	0,03	0,03
Flexion initiale aux genoux (°)	20	20	20
Position médio-latérale du PAMB par rapport au bord intérieur du pied d'appui (m)	0,045	0,045	0,045
Position antéro-postérieure du PAMB par rapport au cardan de cheville du pied d'appui (m)	-0,02/+0,04	-0,05/+0,08	-0,07/+0,012

Tableau 13. Valeurs des paramètres des marches utilisées durant le processus d'optimisation

5. Description du processus d'optimisation

Comme pour l'exemple d'application de la figure du passement filé et pour les mêmes raisons, l'algorithme d'optimisation utilisé ici est le Grey Wolf Optimizer (Mirjalili et al. 2014 [50]). Les paramètres d'optimisation sont les suivants :

- Nombre de loups dans la meute : 1000
- Nombre d'itérations : 50
- Bornes inférieures et supérieures pour les valeurs des paramètres inertiels et géométriques des membres supérieurs :

Paramètre	**Valeur de la borne inférieure**	**Valeur de la borne supérieure**
Hauteur des épaules (m)	0,4	1
Écartement des épaules (m)	0,1	0,5
Masse du bras (kg)	0,1	10
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	10	90
Longueur du bras (m)	0,05	0,6
Masse de l'avant-bras (kg)	0,1	10
Position du centre d'inertie de l'avant-bras (m)	0,02	0,4

Tableau 14. Valeurs des bornes inférieures et supérieures

Dans les cas de balancement actif des membres supérieurs, deux paramètres d'optimisation supplémentaires s'ajoutent :

Paramètre	**Valeur de la borne inférieure**	**Valeur de la borne supérieure**
Facteur de proportion pour la flexion à l'épaule	-10	10
Facteur de proportion pour la flexion au coude	-10	10

Tableau 15. Valeurs des bornes inférieures et supérieures

Les valeurs de ces bornes ont été choisies de sorte que l'optimisation ne puisse pas converger vers des solutions impossibles à fabriquer ou inutilisables par le robot.

La trajectoire et l'orientation dans l'espace du solide correspondant au tronc d'ORHRO est obtenue à partir des codes de génération de marche. Il s'agit du solide sur lequel les membres supérieurs seront montés, aussi bien en simulation que dans la réalité lorsqu'ils auront été fabriqués. Les membres supérieurs sont alignés le long du tronc au premier instant dans une position d'équilibre stable. À partir de là et à chaque itération, tous les loups réalisent chacun un cycle de quatre étapes pour tout le mouvement simulé :

1. Les mouvements des membres supérieurs sont calculés en fonction des valeurs prises par les paramètres inertiels et géométriques lors de l'itération.

2. Les actions mécaniques exercées par les membres supérieurs sur le tronc sont calculées connaissant leurs mouvements.

3. L'action mécanique exercée par le sol est calculée en soustrayant celles obtenues à l'étape précédente à l'action mécanique globale exercée sur ORHRO.

4. Le coefficient de frottement nécessaire entre les pieds et le sol est calculé à partir de l'action mécanique obtenue à l'étape précédente. Sa valeur maximale est ensuite comparée à celles obtenues par d'autres loups pour définir les modifications à opérer sur les paramètres inertiels et géométriques des membres supérieurs pour la prochaine itération.

La meute étant constituée de mille loups chassant pendant cinquante itérations, ce cycle est réalisé cinquante mille fois pour chaque cas d'étude.

Cet exemple, comme le précédent, sollicite la capacité de la MDA à modéliser efficacement des mécanismes dont les valeurs numériques des paramètres sont inconnues. Deux fonctionnalités supplémentaires sont cependant utilisées ici :

1. Le paramètre de positionnement du centre d'inertie du bras a été renseigné en pourcentage de la longueur du bras. Ainsi, le centre d'inertie du bras est obligatoirement situé entre l'épaule et le coude. Cela illustre la possibilité de créer des relations entre paramètres, intrinsèques au modèle, qu'il n'est donc pas nécessaire de contraindre lors de l'optimisation.

2. La MDA écrivant symboliquement les équations de la dynamique, il est possible de lui faire exprimer de nombreuses grandeurs différentes qui leur sont liées et que les générateurs de modèles existants ne permettent pas de calculer nativement, comme dans cet exemple avec l'action mécanique globale appliquée à l'arborescence.

6. Résultats

Le code couleur suivant est valable pour toutes les courbes des graphiques à venir :

- Vert : ORHRO sans membres supérieurs.
- Bleu : ORHRO avec membres supérieurs passifs.
- Rouge : ORHRO avec membres supérieurs actifs.
- Noir pointillé : Position de référence du PAMB.

Le graphique de la position du PAMB dans R_0 affiche également la position des pieds d'ORHRO au cours de la marche avec en orange le pied gauche et en mauve le pied droit.

Des vidéos des membres supérieurs générés et leurs mouvements sont jointes à ce document. Leurs noms sont les codes des cas qu'elles illustrent.

La pagination du document permet de visualiser tous les résultats d'un cas sur deux pages.

a. Passif avec l'articulation du coude bloquée

Vitesse de marche de 0,1 m/s

Code du cas : PCB01.

Durée de l'optimisation : 10,28 heures.
Durée moyenne d'un cycle : 0,740 seconde.

Paramètre	Valeur optimale PCB01	Position relative aux bornes
Hauteur des épaules (m)	0,614	
Écartement des épaules (m)	0,500	
Masse du bras (kg)	0,498	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	28,4	
Longueur du bras (m)	0,520	
Masse de l'avant-bras (kg)	3,131	
Position du centre d'inertie de l'avant-bras (m)	0,026	

Tableau 16. Valeurs optimales des paramètres d'optimisation

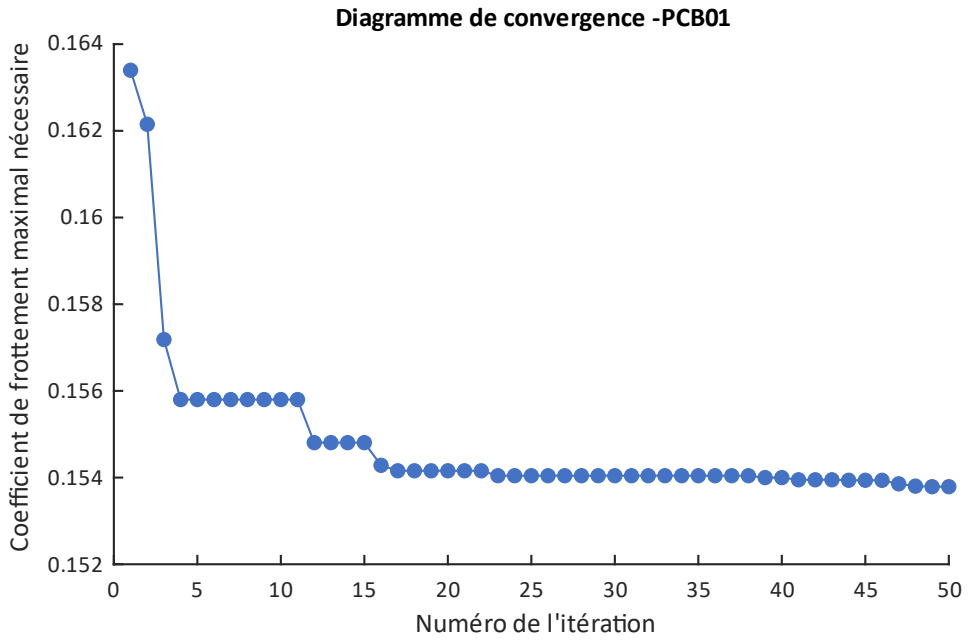


Figure 43

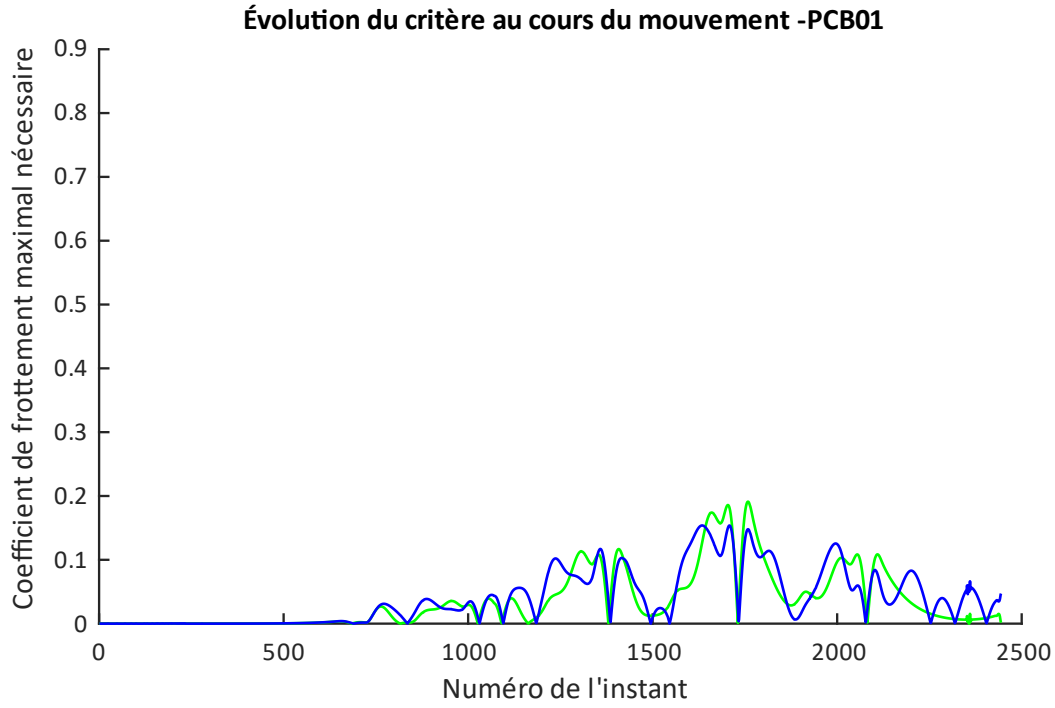


Figure 44

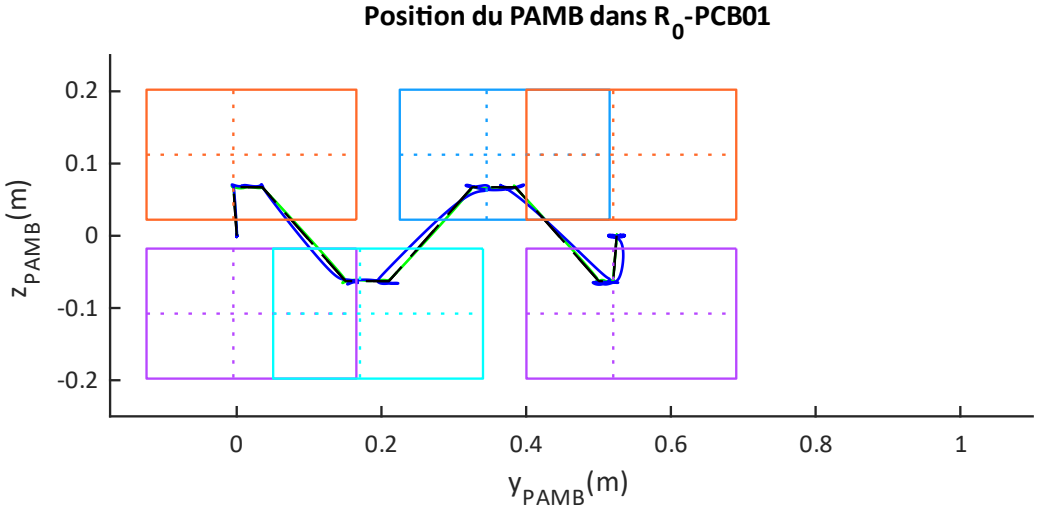


Figure 45

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-10	10	-15	15
Coude	0	0	0	0

Tableau 17. Valeurs des angles de flexion minimum et maximum

Vitesse de marche de 0,2 m/s

Code du cas : PCB02.

Durée de l'optimisation : 8,16 heures.
Durée moyenne d'un cycle : 0,588 seconde.

Paramètre	Valeur optimale PCB02	Position relative aux bornes
Hauteur des épaules (m)	0,519	
Écartement des épaules (m)	0,219	
Masse du bras (kg)	7,244	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	21,3	
Longueur du bras (m)	0,339	
Masse de l'avant-bras (kg)	6,387	
Position du centre d'inertie de l'avant-bras (m)	0,384	

Tableau 18. Valeurs optimales des paramètres d'optimisation

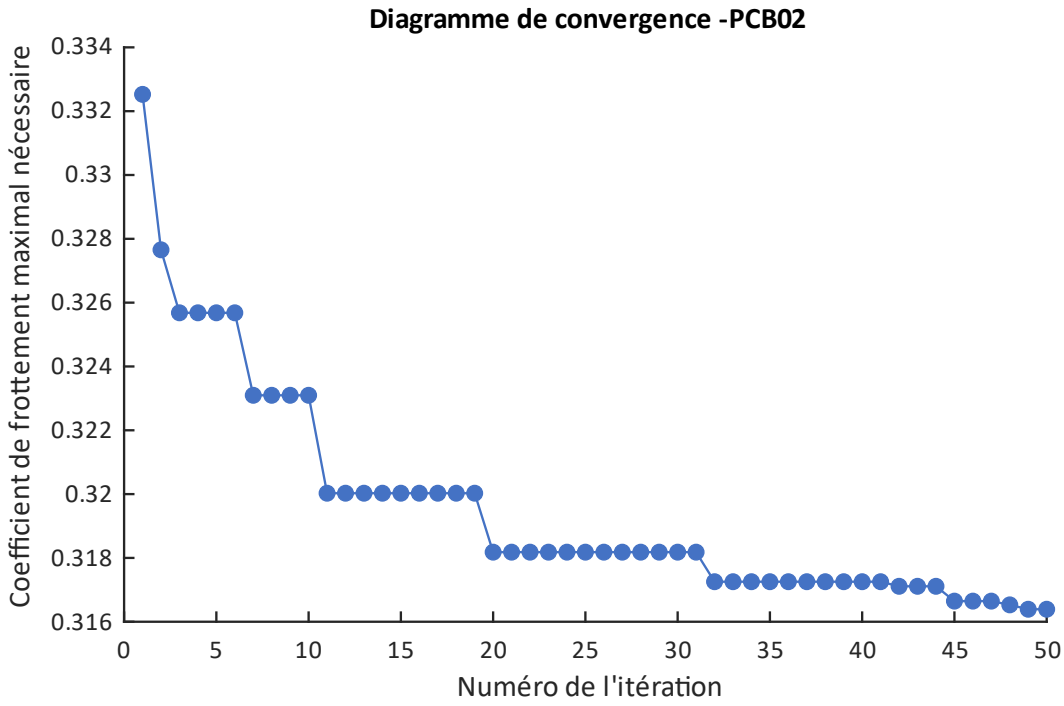


Figure 46

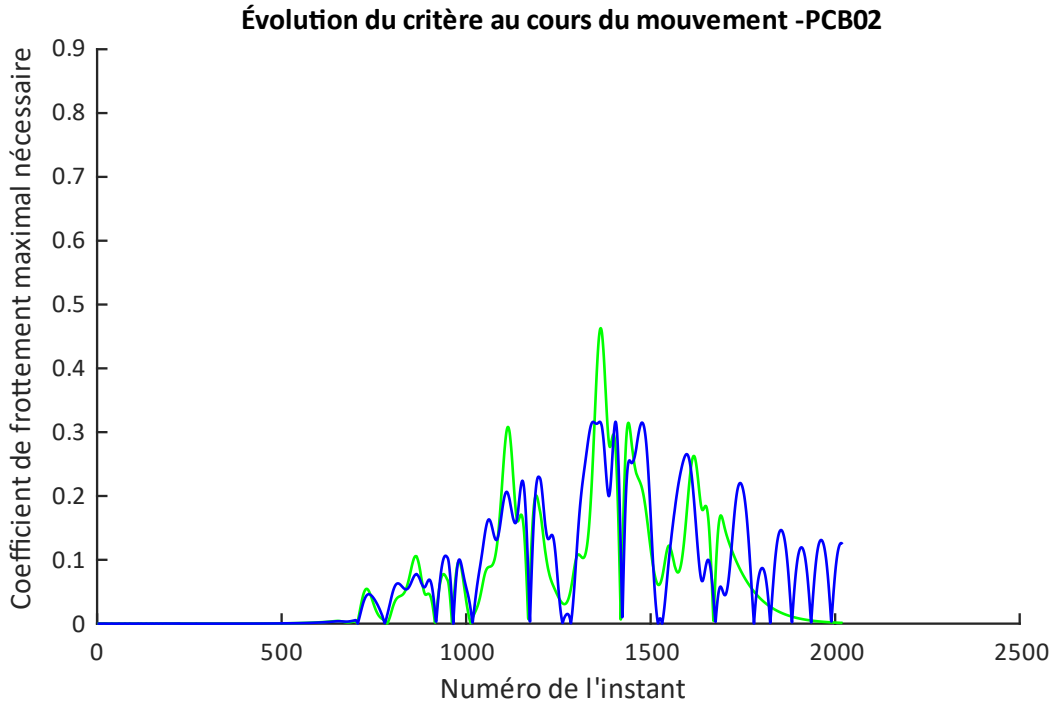


Figure 47

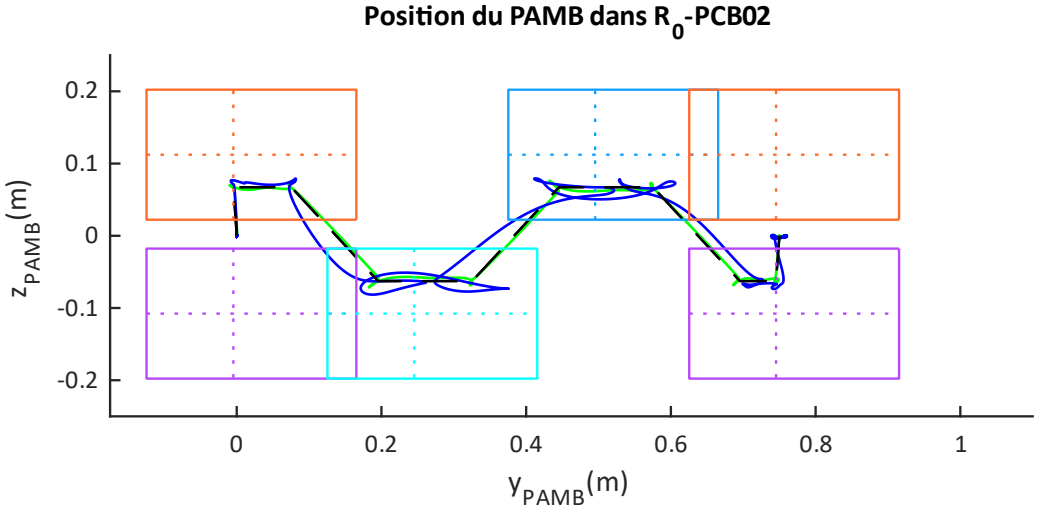


Figure 48

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-12	11	-19	18
Coude	0	0	0	0

Tableau 19. Valeurs des angles de flexion minimum et maximum

Vitesse de marche de 0,3 m/s

Code du cas : PCB03.

Durée de l'optimisation : 8,55 heures.
Durée moyenne d'un cycle : 0,615 seconde.

Paramètre	Valeur optimale PCB03	Position relative aux bornes
Hauteur des épaules (m)	0,622	
Écartement des épaules (m)	0,174	
Masse du bras (kg)	0,103	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	12,3	
Longueur du bras (m)	0,592	
Masse de l'avant-bras (kg)	8,143	
Position du centre d'inertie de l'avant-bras (m)	0,237	

Tableau 20. Valeurs optimales des paramètres d'optimisation

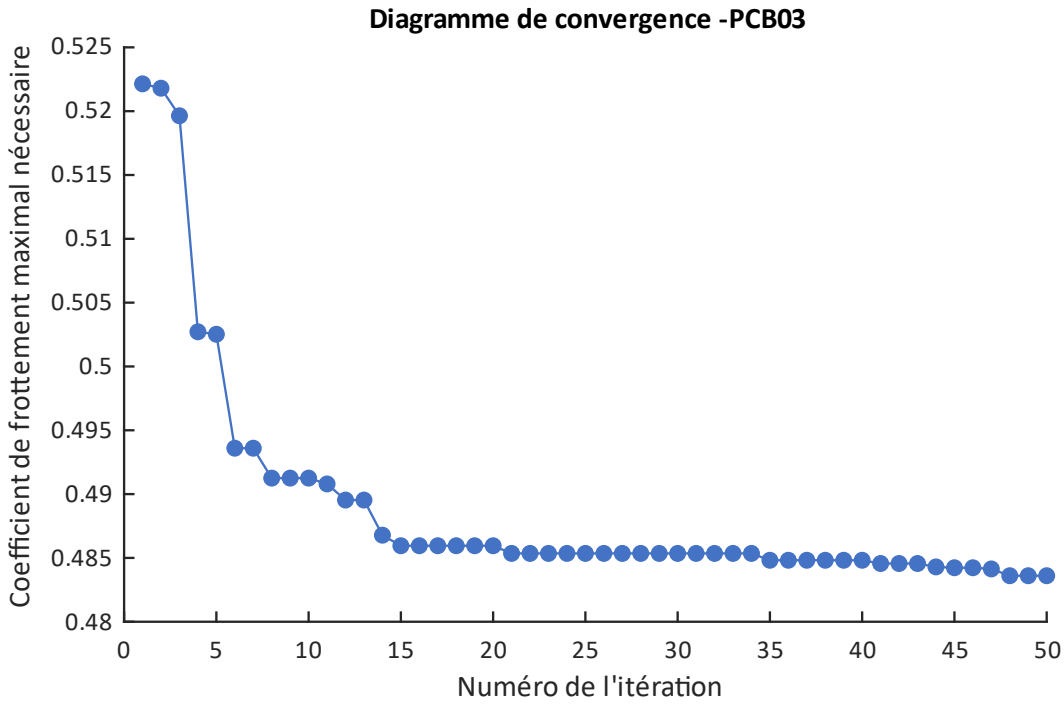


Figure 49

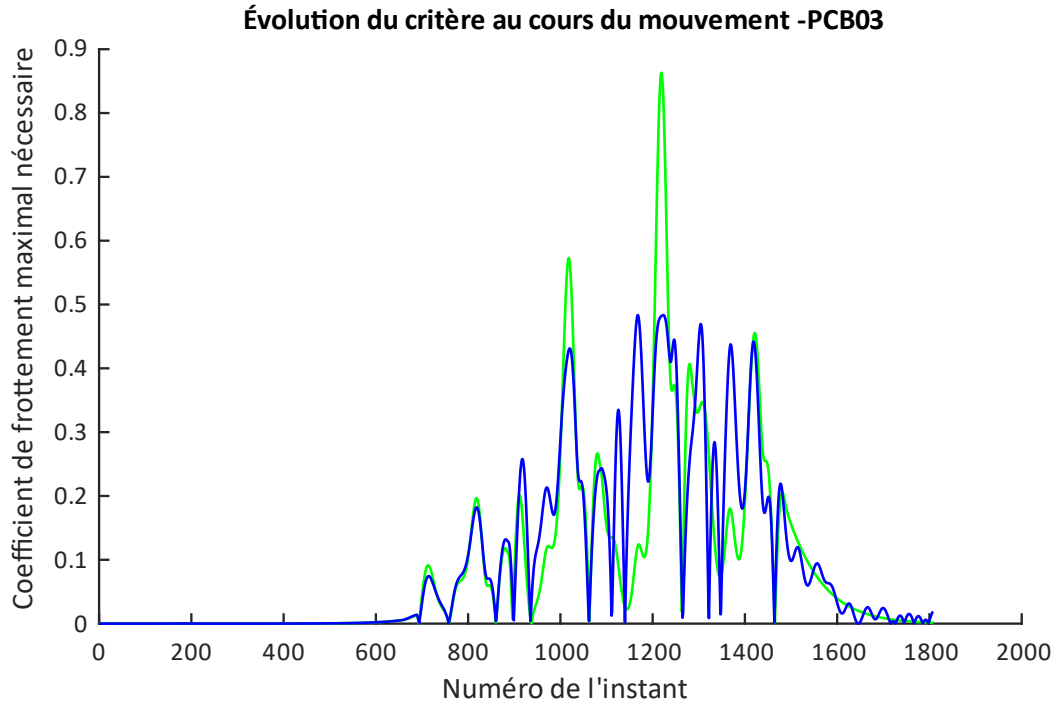


Figure 50

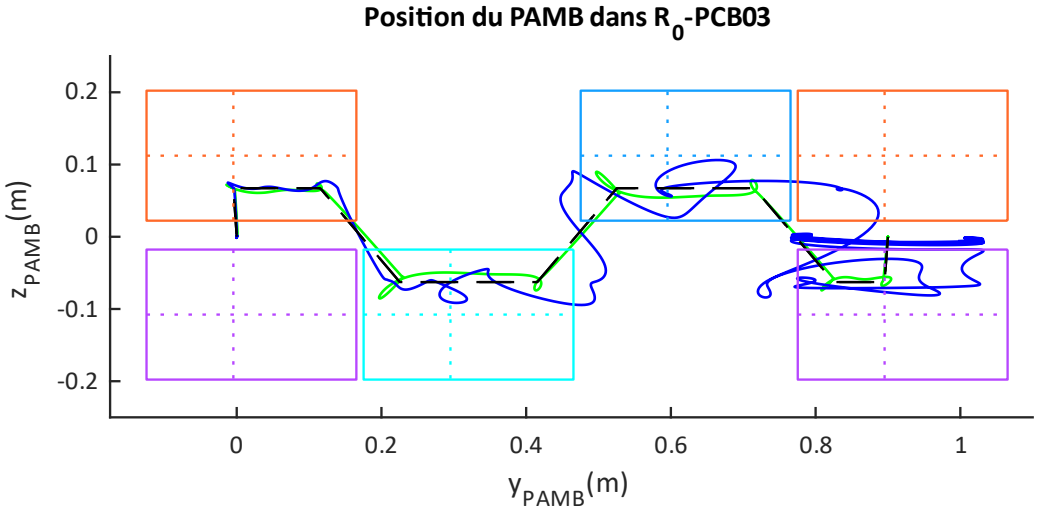


Figure 51

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-26	25	-25	26
Coude	0	0	0	0

Tableau 21. Valeurs des angles de flexion minimum et maximum

b. Passif avec l'articulation du coude libre

Vitesse de marche de 0,1 m/s

Code du cas : PCL01.

Durée de l'optimisation : 9,94 heures.
Durée moyenne d'un cycle : 0,716 seconde.

Paramètre	Valeur optimale PCL01	Position relative aux bornes
Hauteur des épaules (m)	0,981	
Écartement des épaules (m)	0,500	
Masse du bras (kg)	3,265	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	87,2	
Longueur du bras (m)	0,130	
Masse de l'avant-bras (kg)	1,119	
Position du centre d'inertie de l'avant-bras (m)	0,066	

Tableau 22. . Valeurs optimales des paramètres d'optimisation

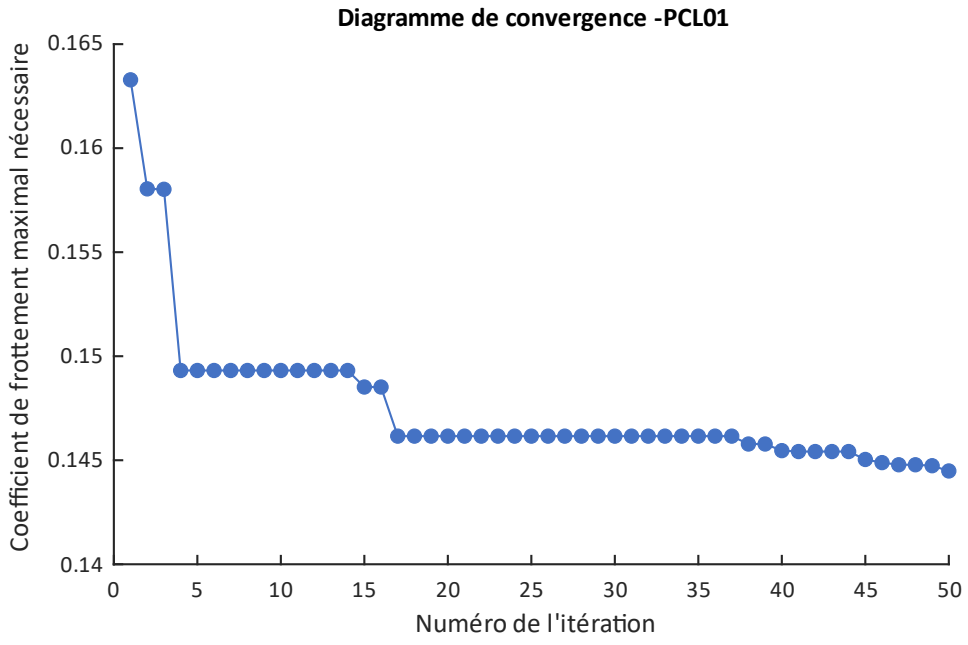


Figure 52

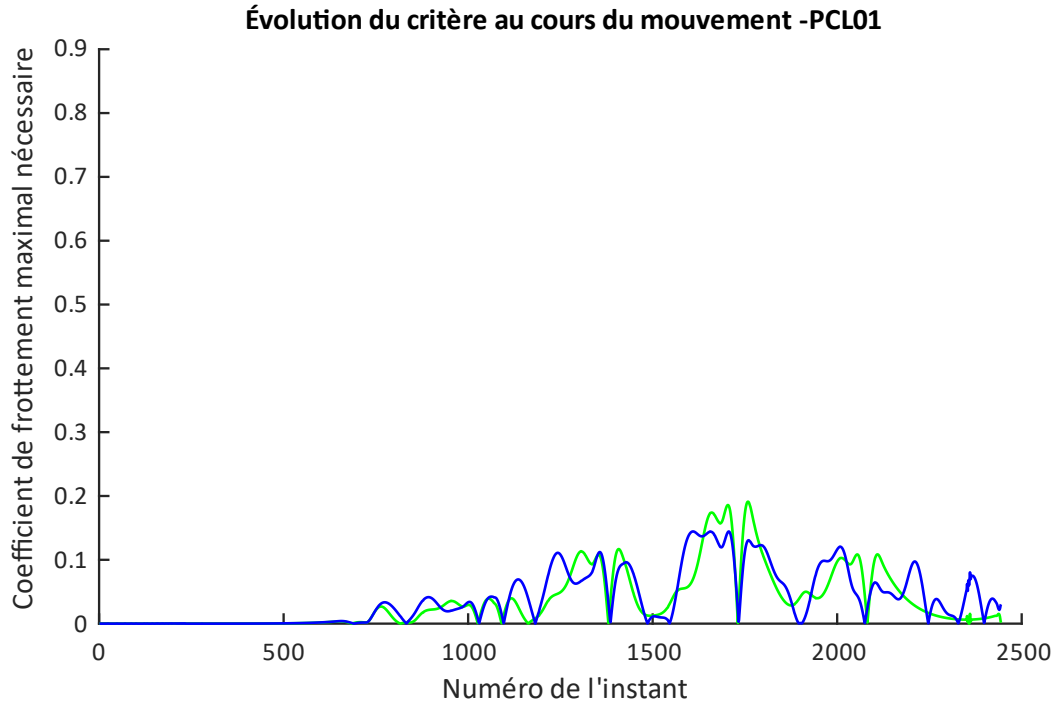


Figure 53

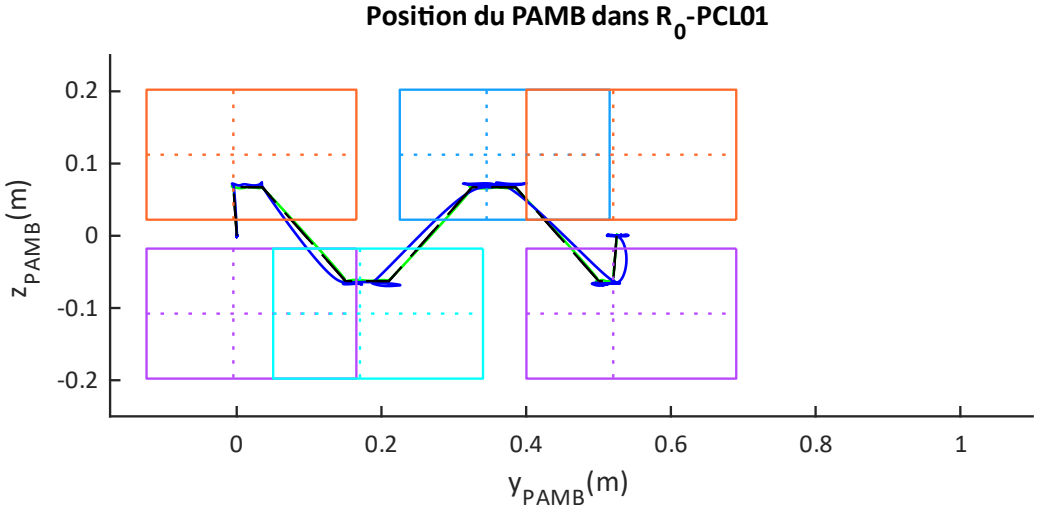


Figure 54

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-9	9	-13	15
Coude	-10	8	-14	11

Tableau 23. Valeurs des angles de flexion minimum et maximum

Vitesse de marche de 0,2 m/s

Code du cas : PCL02.

Durée de l'optimisation : 8,31 heures.
Durée moyenne d'un cycle : 0,598 seconde.

Paramètre	Valeur optimale PCL02	Position relative aux bornes
Hauteur des épaules (m)	0,965	
Écartement des épaules (m)	0,312	
Masse du bras (kg)	7,977	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	24,8	
Longueur du bras (m)	0,400	
Masse de l'avant-bras (kg)	0,445	
Position du centre d'inertie de l'avant-bras (m)	0,398	

Tableau 24. Valeurs optimales des paramètres d'optimisation

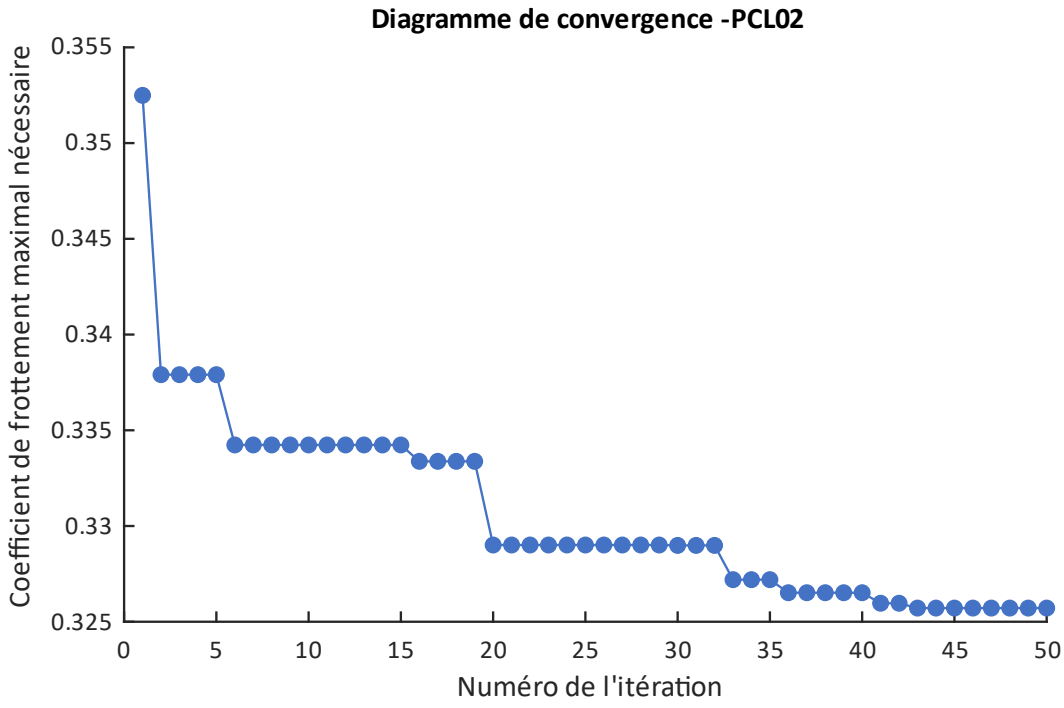


Figure 55

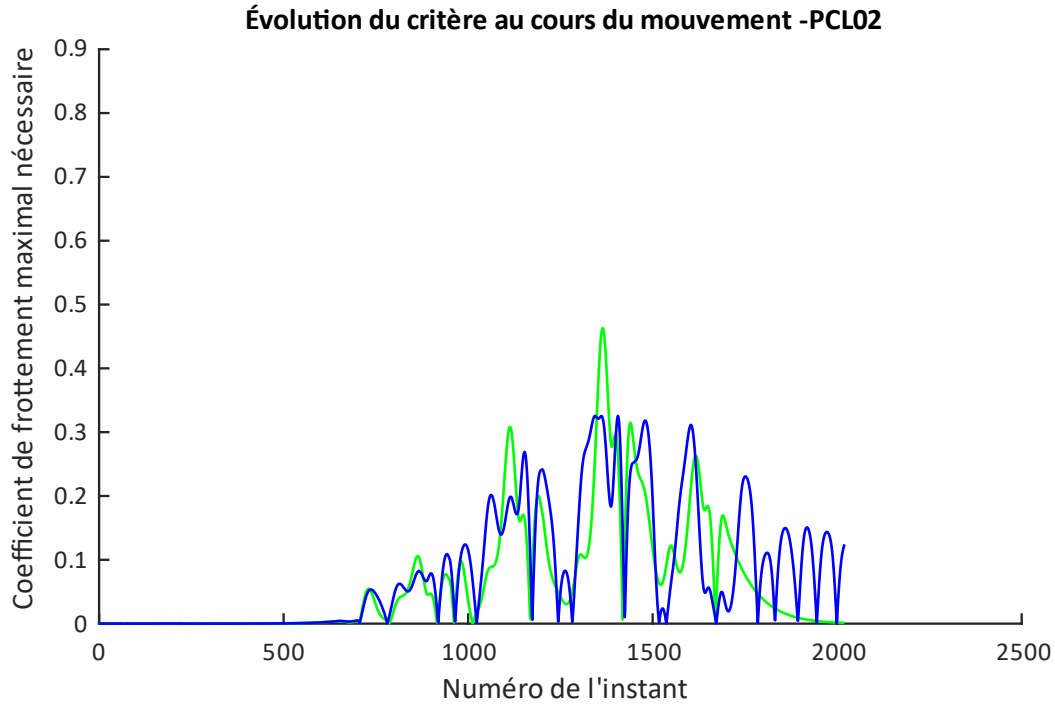


Figure 56

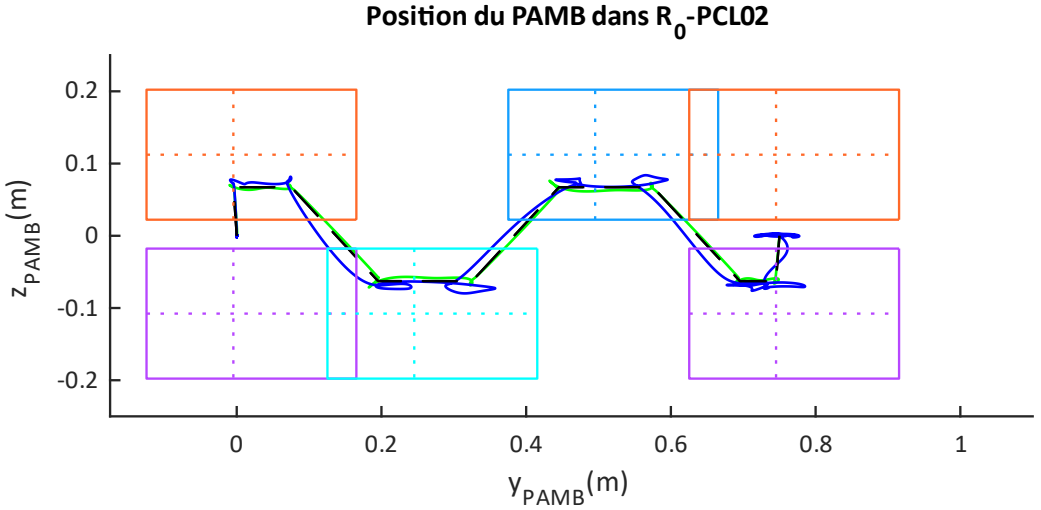


Figure 57

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-16	18	-13	17
Coude	-41	43	-34	39

Tableau 25. Valeurs des angles de flexion minimum et maximum

Vitesse de marche de 0,3 m/s

Code du cas : PCL03.

Durée de l'optimisation : 7,03 heures.
Durée moyenne d'un cycle : 0,506 seconde.

Paramètre	Valeur optimale PCL03	Position relative aux bornes
Hauteur des épaules (m)	0,970	
Écartement des épaules (m)	0,500	
Masse du bras (kg)	10,000	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	66,0	
Longueur du bras (m)	0,600	
Masse de l'avant-bras (kg)	10,000	
Position du centre d'inertie de l'avant-bras (m)	0,400	

Tableau 26. Valeurs optimales des paramètres d'optimisation

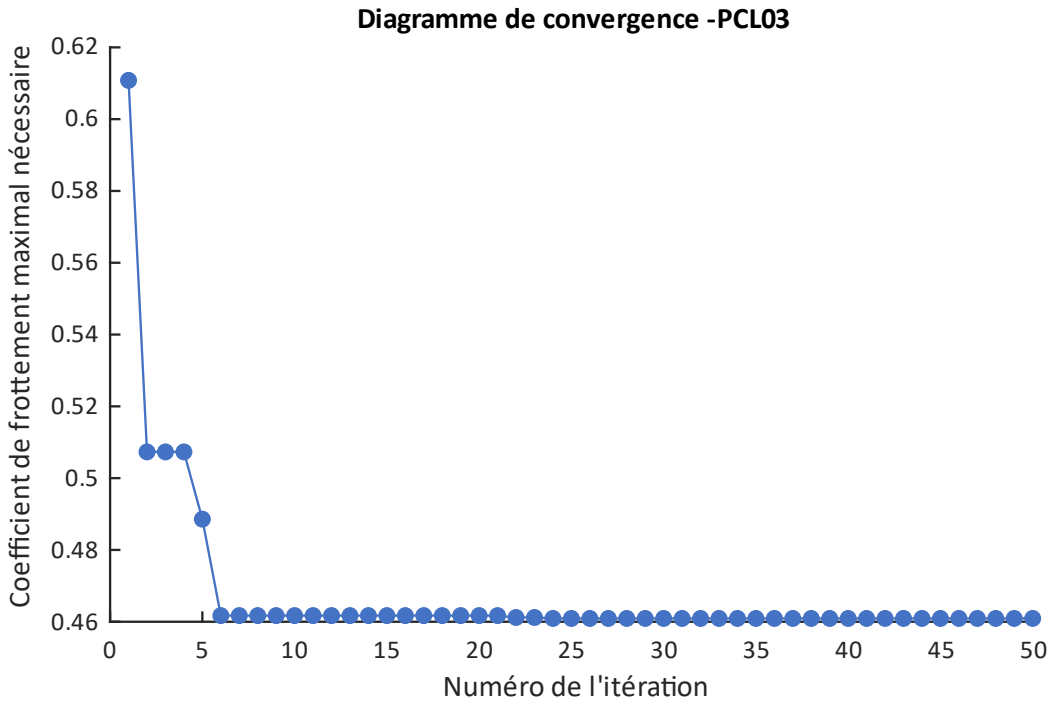


Figure 58

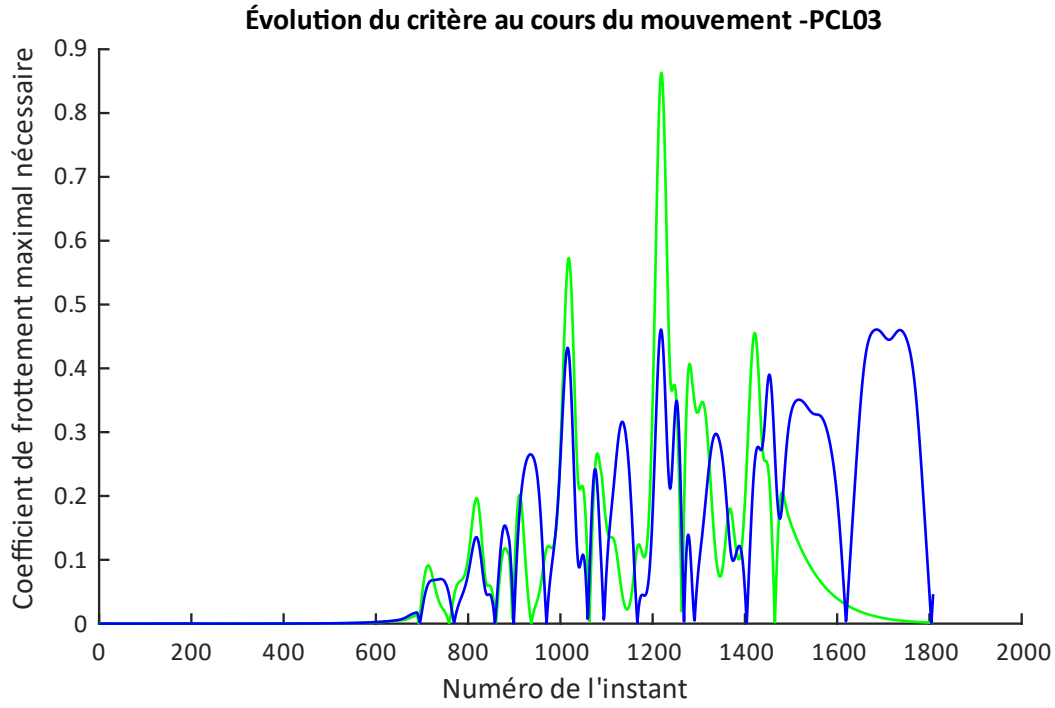


Figure 59

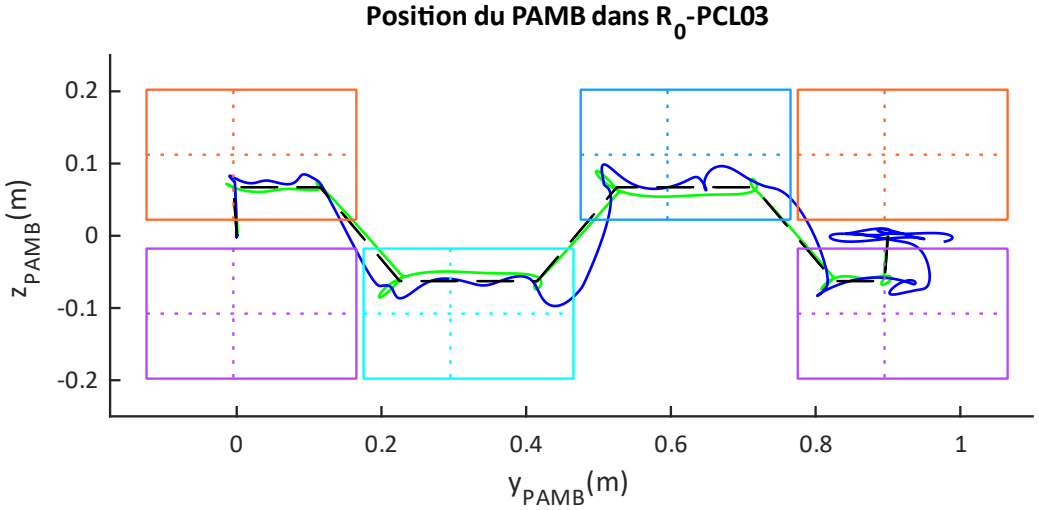


Figure 60

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-13	14	-19	19
Coude	-12	15	-13	16

Tableau 27. Valeurs des angles de flexion minimum et maximum

c. Actif commandé en proportion de l'angle de flexion de la hanche

Vitesse de marche de 0,1 m/s

Code du cas : ACFH01.

Durée de l'optimisation : 4,30 heures.
Durée moyenne d'un cycle : 0,310 seconde.

Paramètre	Valeur optimale ACFH01	Position relative aux bornes
Hauteur des épaules (m)	0,624	
Écartement des épaules (m)	0,500	
Masse du bras (kg)	0,104	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	90,0	
Longueur du bras (m)	0,600	
Masse de l'avant-bras (kg)	1,031	
Position du centre d'inertie de l'avant-bras (m)	0,400	
Facteur de proportion pour la flexion à l'épaule	-4,565	
Facteur de proportion pour la flexion au coude	0,244	

Tableau 28. Valeurs optimales des paramètres d'optimisation

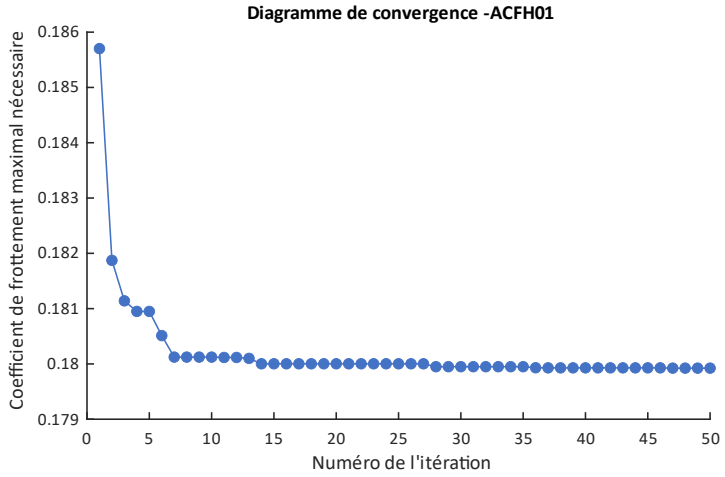


Figure 61

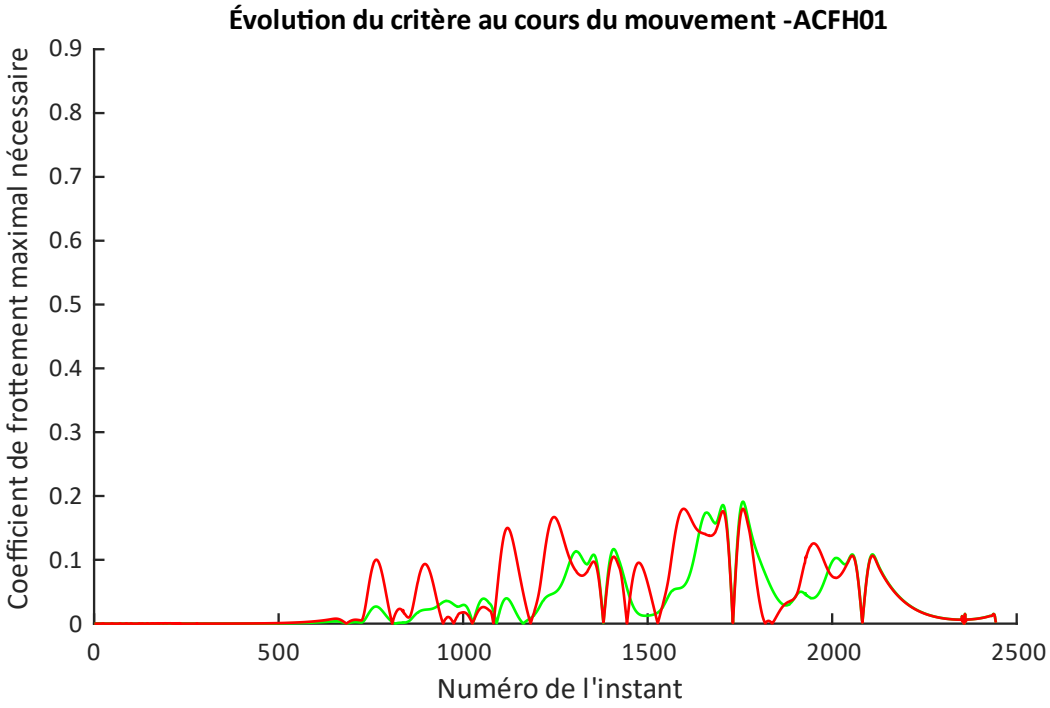


Figure 62

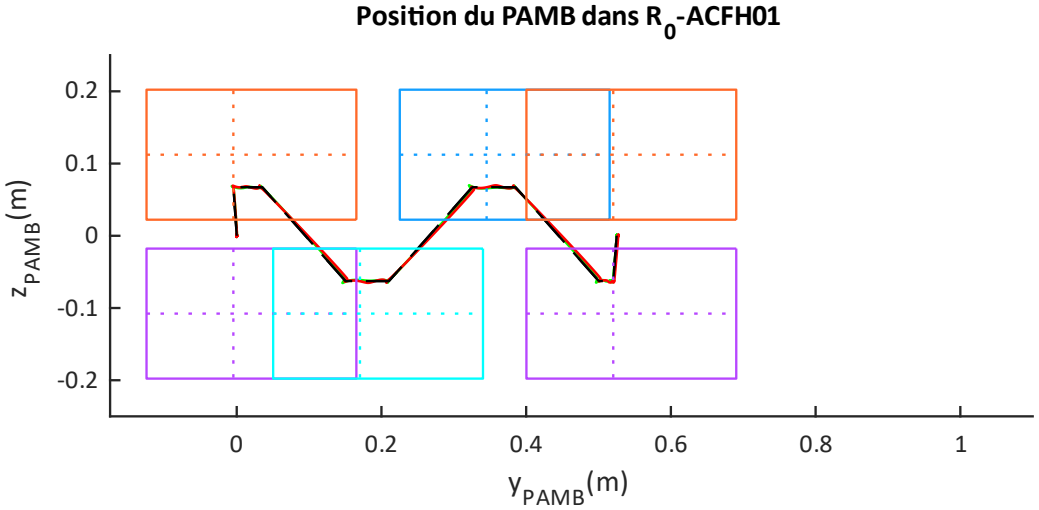


Figure 63

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-29	42	-26	43
Coude	-2	2	-2	1

Tableau 29. Valeurs des angles de flexion minimum et maximum

Vitesse de marche de 0,2 m/s

Code du cas : ACFH02.

Durée de l'optimisation : 7,14 heures.
Durée moyenne d'un cycle : 0,514 seconde.

Paramètre	Valeur optimale ACFH02	Position relative aux bornes
Hauteur des épaules (m)	0,810	
Écartement des épaules (m)	0,125	
Masse du bras (kg)	10,000	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	52,0	
Longueur du bras (m)	0,600	
Masse de l'avant-bras (kg)	10,000	
Position du centre d'inertie de l'avant-bras (m)	0,304	
Facteur de proportion pour la flexion à l'épaule	-3,777	
Facteur de proportion pour la flexion au coude	10,000	

Tableau 30. Valeurs optimales des paramètres d'optimisation

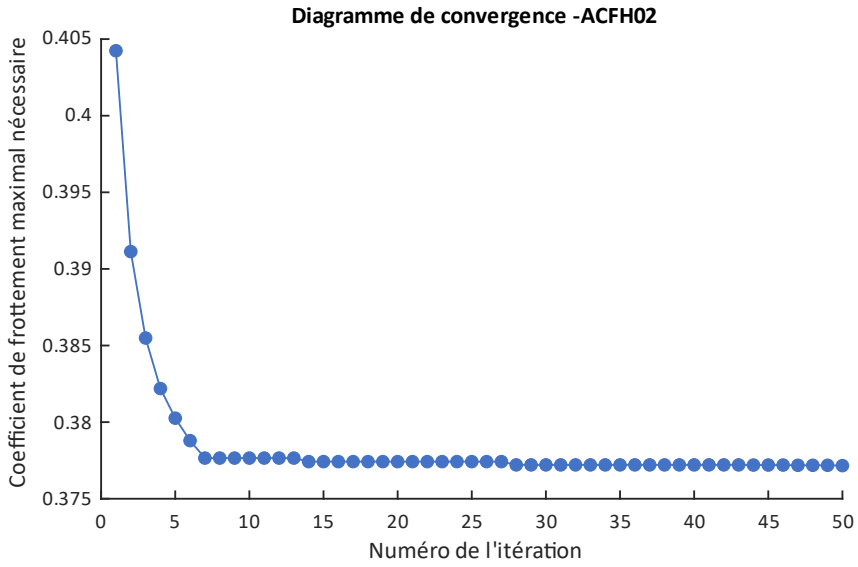


Figure 64

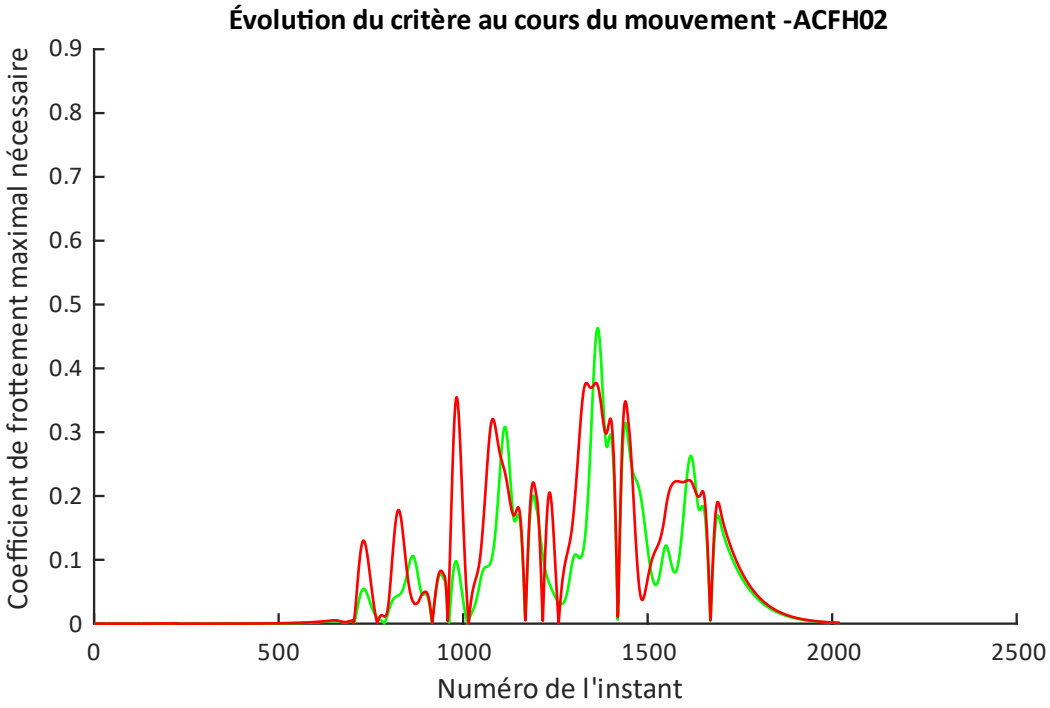


Figure 65

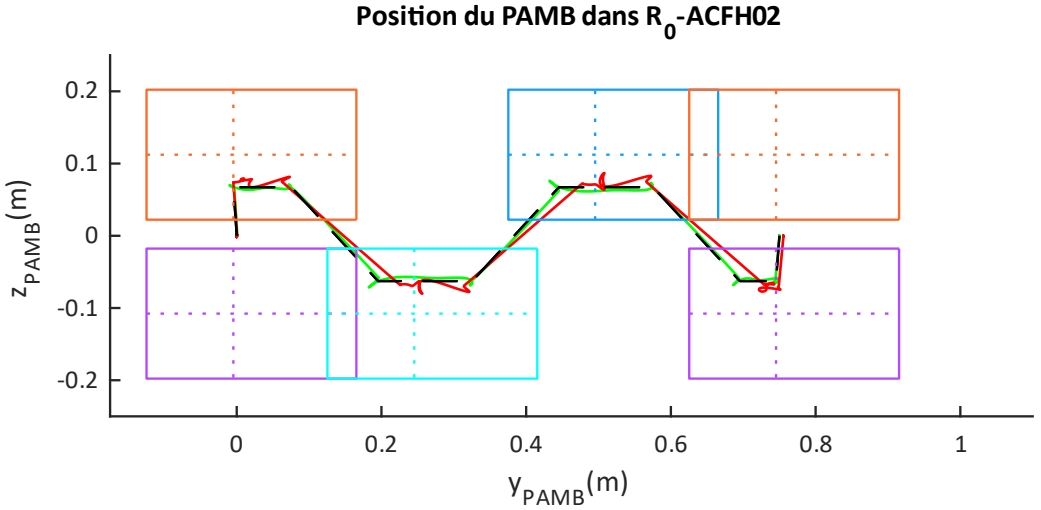


Figure 66

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-33	48	-32	47
Coude	-127	87	-125	84

Tableau 31. Valeurs des angles de flexion minimum et maximum

Vitesse de marche de 0,3 m/s

Code du cas : ACFH03.

Durée de l'optimisation : 3,15 heures.
Durée moyenne d'un cycle : 0,227 seconde.

Paramètre	Valeur optimale ACFH03	Position relative aux bornes
Hauteur des épaules (m)	0,404	
Écartement des épaules (m)	0,100	
Masse du bras (kg)	10,000	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	26,4	
Longueur du bras (m)	0,600	
Masse de l'avant-bras (kg)	10,000	
Position du centre d'inertie de l'avant-bras (m)	0,400	
Facteur de proportion pour la flexion à l'épaule	-4,036	
Facteur de proportion pour la flexion au coude	10,000	

Tableau 32. Valeurs optimales des paramètres d'optimisation

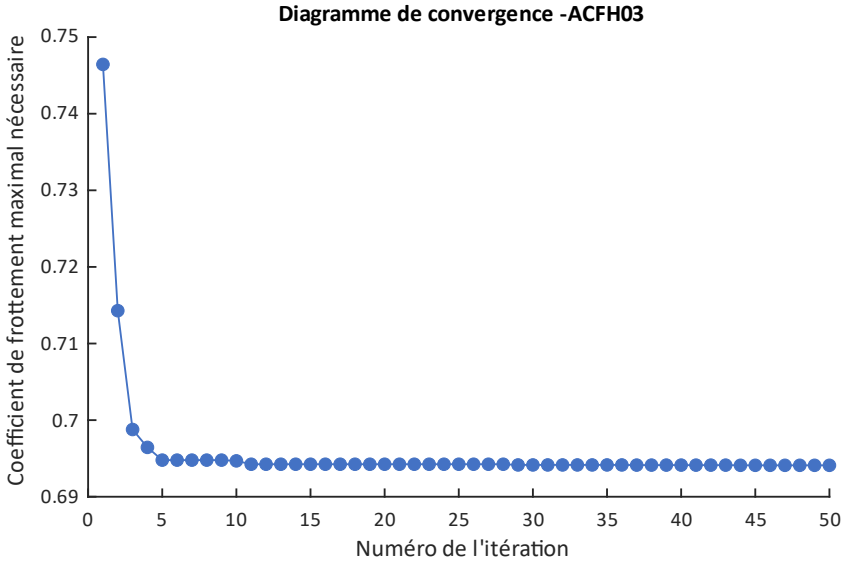


Figure 67

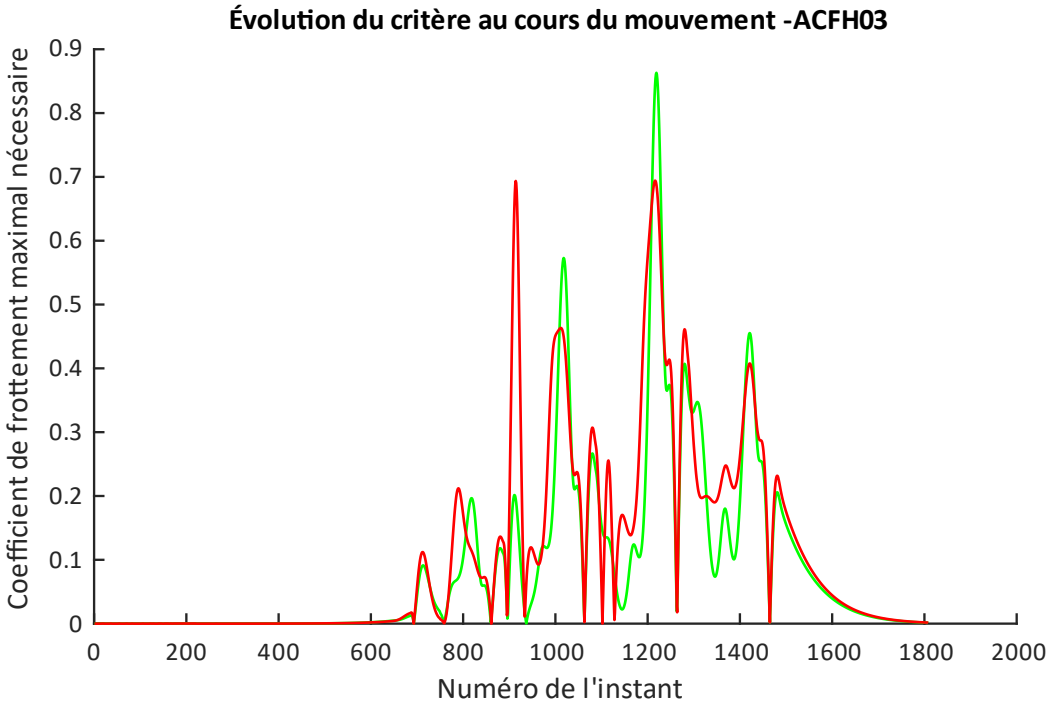


Figure 68

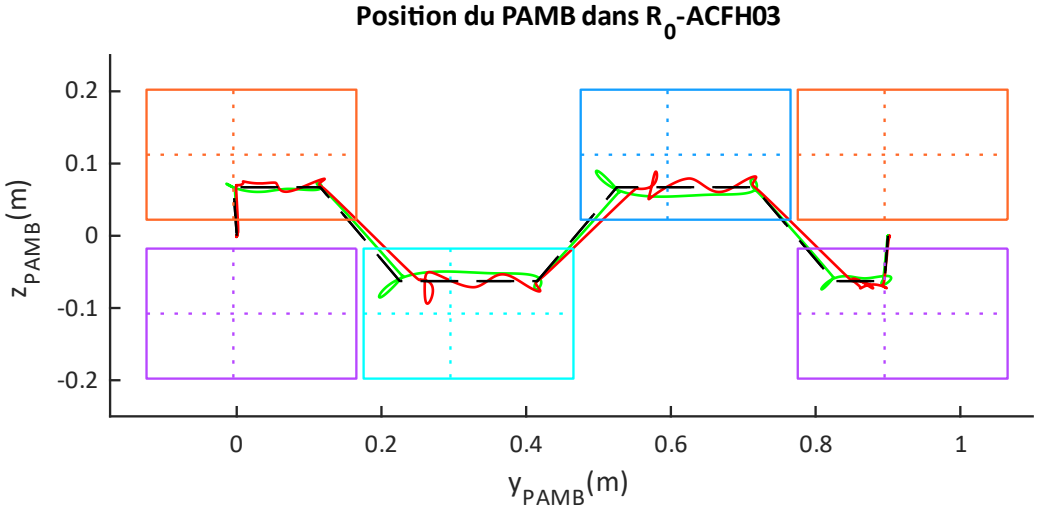


Figure 69

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-43	62	-41	61
Coude	-154	106	-150	102

Tableau 33. Valeurs des angles de flexion minimum et maximum

d. Actif commandé en proportion de l'angle de lacet du tronc

Vitesse de marche de 0,1 m/s

Code du cas : ACLT01.

Durée de l'optimisation : 4,37 heures.
Durée moyenne d'un cycle : 0,315 seconde.

Paramètre	Valeur optimale ACLT01	Position relative aux bornes
Hauteur des épaules (m)	0,600	
Écartement des épaules (m)	0,500	
Masse du bras (kg)	0,100	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	90,0	
Longueur du bras (m)	0,600	
Masse de l'avant-bras (kg)	0,269	
Position du centre d'inertie de l'avant-bras (m)	0,400	
Facteur de proportion pour la flexion à l'épaule	-8,427	
Facteur de proportion pour la flexion au coude	0,838	

Tableau 34. Valeurs optimales des paramètres d'optimisation

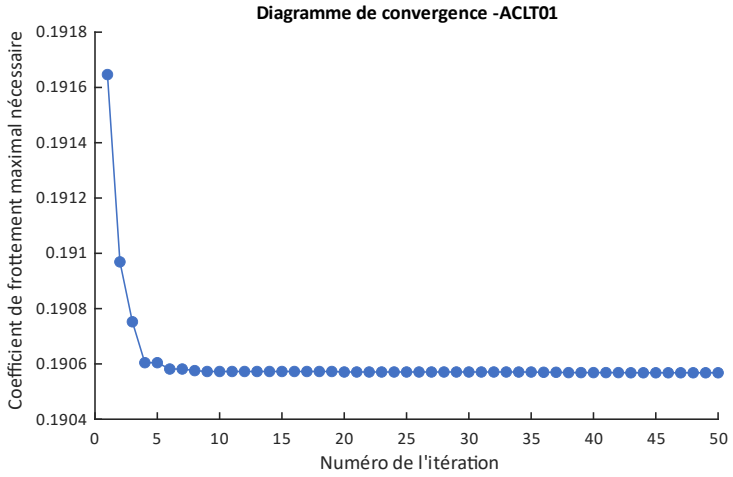


Figure 70

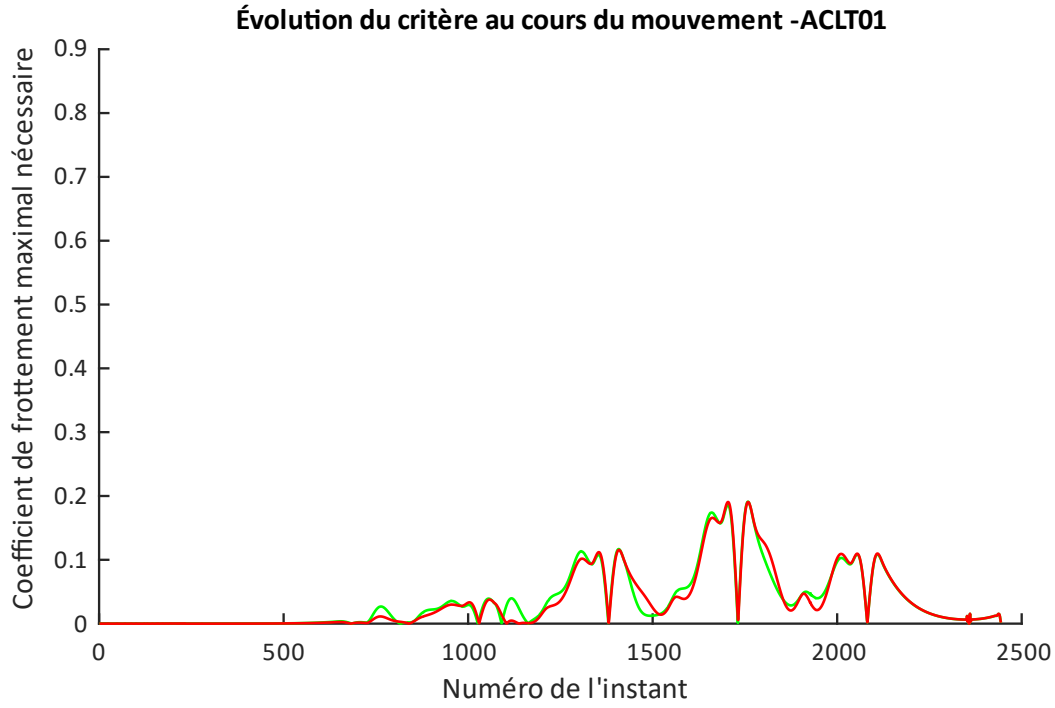


Figure 71

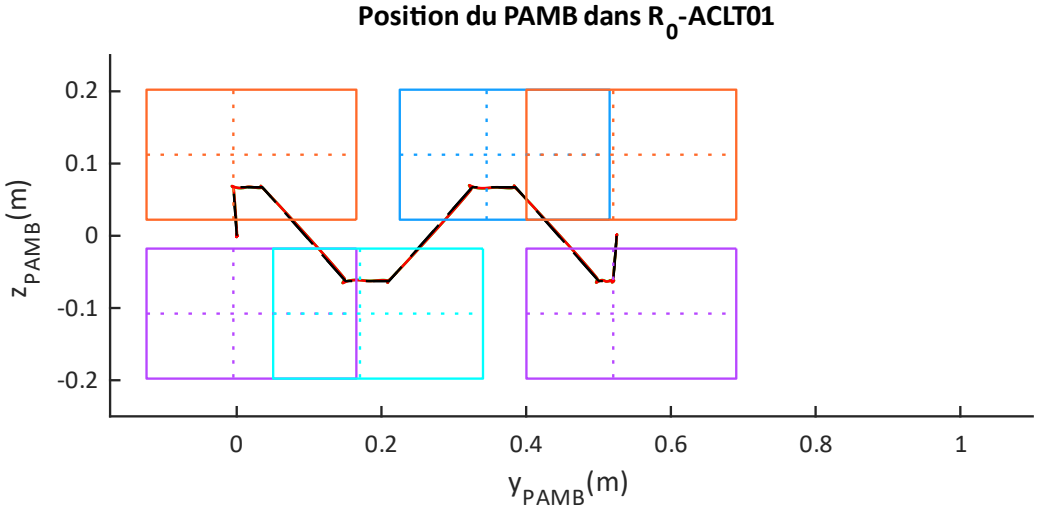


Figure 72

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-34	29	-29	34
Coude	-3	3	-3	3

Tableau 35. Valeurs des angles de flexion minimum et maximum

Vitesse de marche de 0,2 m/s

Code du cas : ACLT02.

Durée de l'optimisation : 3,51 heures.
Durée moyenne d'un cycle : 0,253 seconde.

Paramètre	Valeur optimale ACLT02	Position relative aux bornes
Hauteur des épaules (m)	0,650	
Écartement des épaules (m)	0,387	
Masse du bras (kg)	7,949	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	70,1	
Longueur du bras (m)	0,495	
Masse de l'avant-bras (kg)	9,780	
Position du centre d'inertie de l'avant-bras (m)	0,020	
Facteur de proportion pour la flexion à l'épaule	-1,889	
Facteur de proportion pour la flexion au coude	0,256	

Tableau 36. Valeurs optimales des paramètres d'optimisation

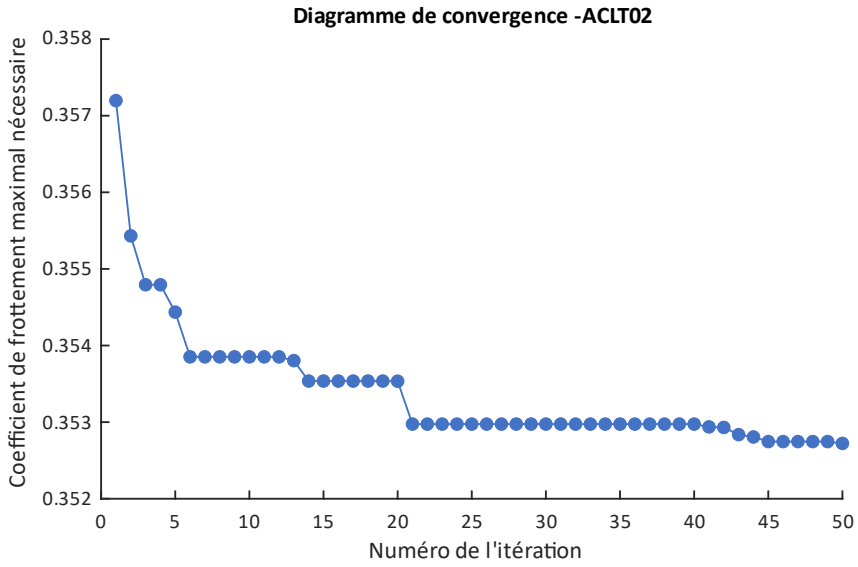


Figure 73

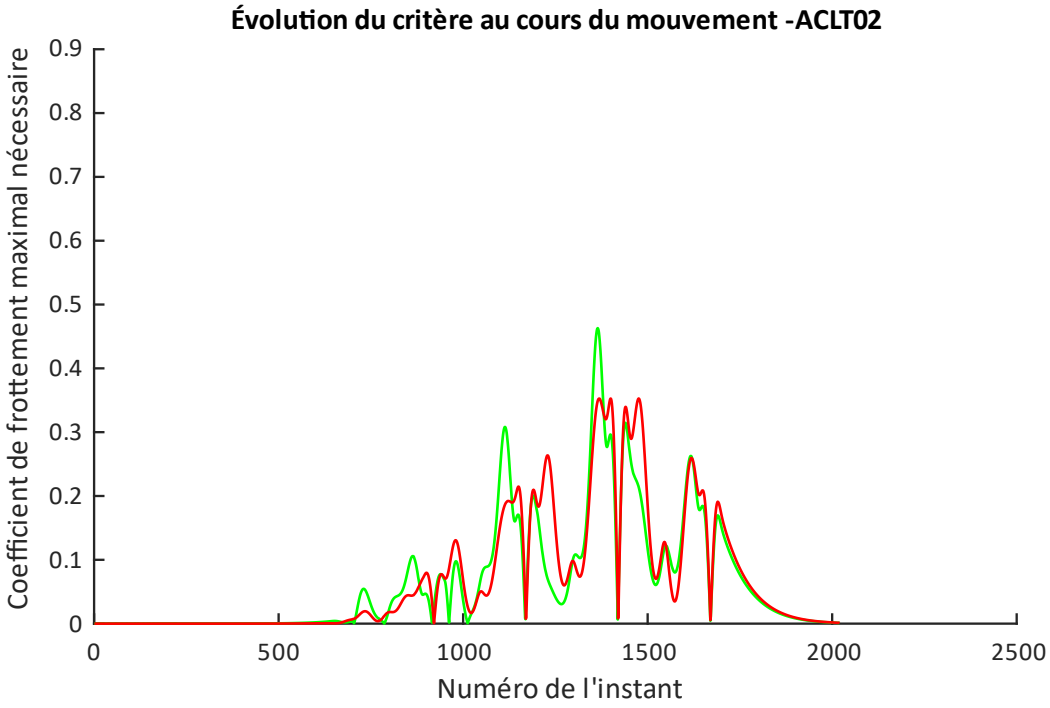


Figure 74

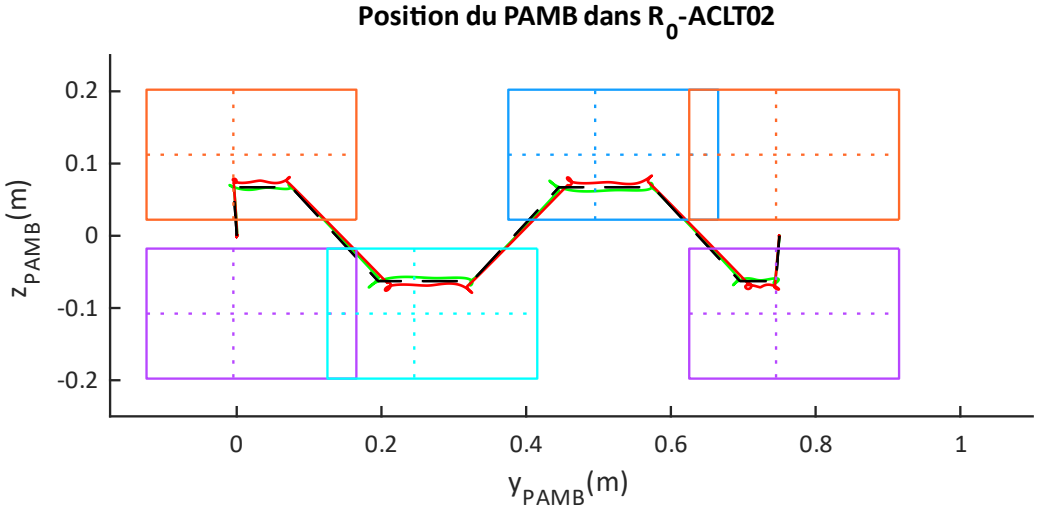


Figure 75

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-11	9	-9	11
Coude	-1	1	-1	1

Tableau 37. Valeurs des angles de flexion minimum et maximum

Vitesse de marche de 0,3 m/s

Code du cas : ACLT03.

Durée de l'optimisation : 3,19 heures.
Durée moyenne d'un cycle : 0,230 seconde.

Paramètre	Valeur optimale ACLT03	Position relative aux bornes
Hauteur des épaules (m)	0,475	
Écartement des épaules (m)	0,452	
Masse du bras (kg)	10,000	
Position du centre d'inertie du bras (% de la longueur proximo-distale du bras)	88,8	
Longueur du bras (m)	0,600	
Masse de l'avant-bras (kg)	10,000	
Position du centre d'inertie de l'avant-bras (m)	0,054	
Facteur de proportion pour la flexion à l'épaule	-1,302	
Facteur de proportion pour la flexion au coude	-1,535	

Tableau 38. Valeurs optimales des paramètres d'optimisation

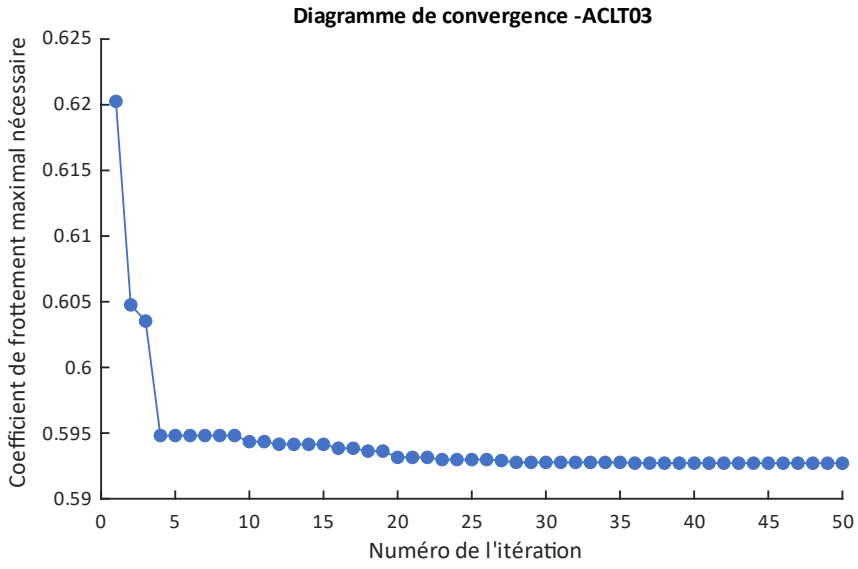


Figure 76

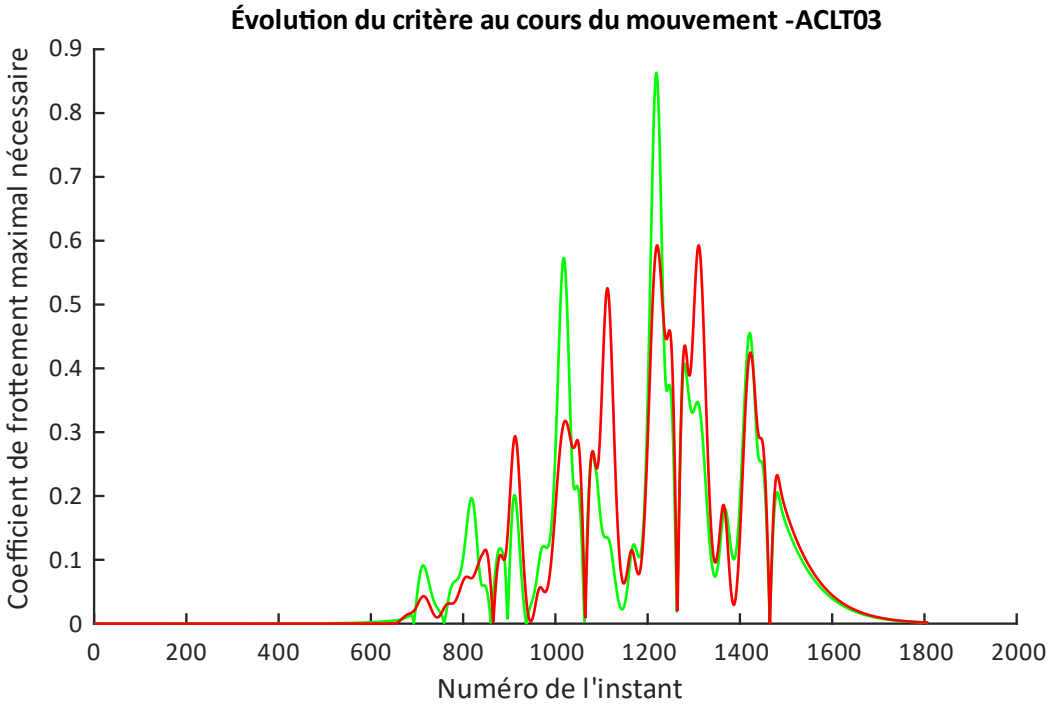


Figure 77

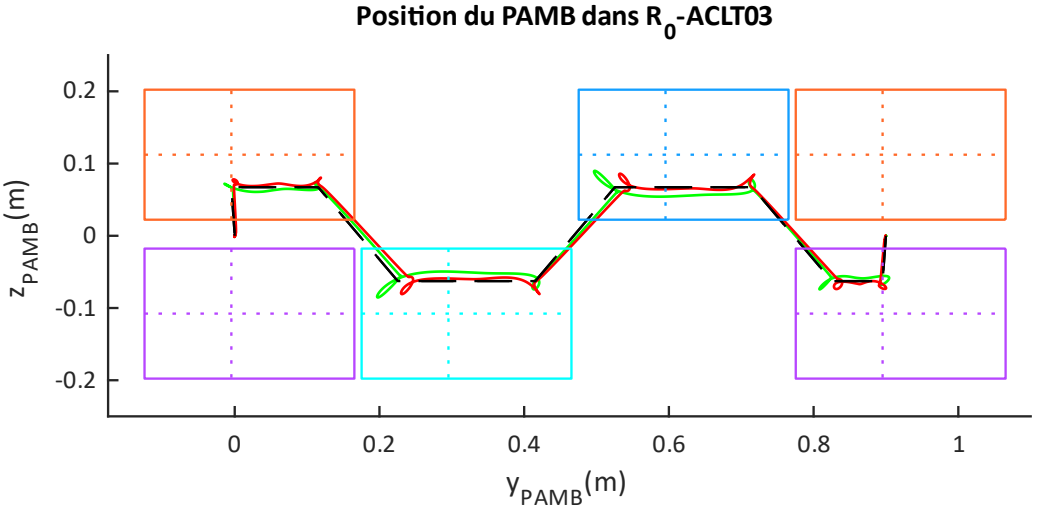


Figure 78

Articulation	Angle pour le membre supérieur gauche (°)		Angle pour le membre supérieur droit (°)	
	Minimum	Maximum	Minimum	Maximum
Épaule	-9	8	-8	9
Coude	-11	9	-9	11

Tableau 39. Valeurs des angles de flexion minimum et maximum

7. Comparaison et discussion

a. Valeurs optimisées des paramètres inertiels et géométriques

Chacun des graphiques suivants illustre les valeurs optimisées d'un paramètre pour tous les cas étudiés.

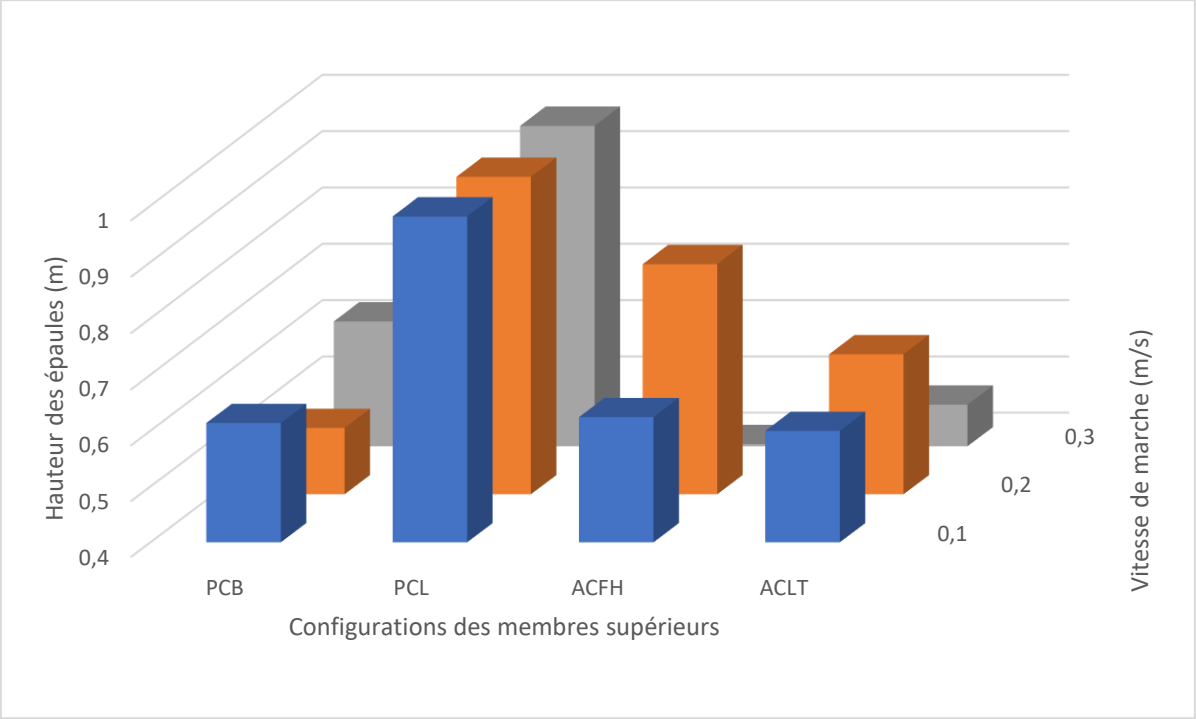


Figure 79

La hauteur des épaules s'est comportée de deux manières en fonction des configurations des membres supérieurs.

Pour les cas PCB, ACFH et ACLT, la majorité de ses valeurs se sont stabilisées entre 0,5 et 0,6 mètre. C'est un résultat particulièrement intéressant pour trois raisons : la hauteur des épaules est un paramètre qui agit peu sur le coefficient de frottement, les bornes ne sont pas atteintes et les balancements obtenus ont des allures très différentes.

- La hauteur des épaules agit peu en effet sur le coefficient de frottement car ce déplacement s'effectue selon la verticale – ORHRO gardant son tronc aligné avec cette direction. Un déplacement selon la verticale ne modifie pas l'expression du couple de frottement entre les pieds et le sol lorsque ce dernier est exprimé à l'intersection entre l'axe vertical du tronc et le plan du sol. Ce sont donc les variations de la position de ce point d'expression dans le plan du sol qui donne de l'influence à ce paramètre.

- Le fait que les bornes ne soient pas atteintes donne confiance qu'entre 0,5 et 0,6 mètre se trouve un bon compromis pour de nombreuses allures de balancements différentes, l'optimisation n'ayant pas cherché à sortir de l'intervalle et s'étant stabilisée autour de valeurs proches.

Pour le cas PCL, qu'importe la vitesse de marche, la hauteur des épaules est égale à la valeur de la borne supérieure soit 1 mètre. Cela indique que pour cette configuration en particulier, augmenter la valeur de ce paramètre est bénéfique à la diminution du coefficient de frottement maximal nécessaire.

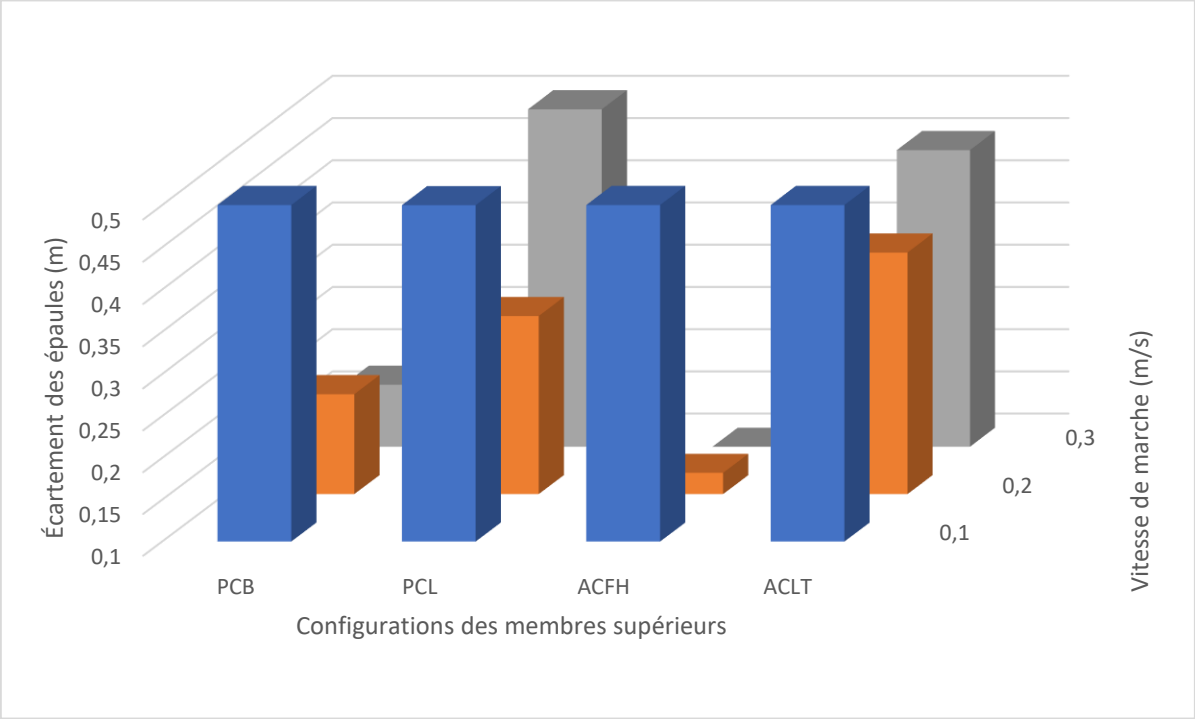


Figure 80

L'écartement des épaules en revanche, éloignant les bras de la verticale du tronc, influe directement sur la valeur du couple de frottement entre les pieds et le sol et donc sur le coefficient de frottement maximal nécessaire. Pour un balancement des membres supérieurs donné, écarter ces derniers augmente son impact. De même, pour un mouvement donné du tronc, éloigner les épaules générera des déplacements plus amples. L'écartement des épaules agit, de manière simplifiée, comme un facteur multiplicateur. Il n'est ainsi pas étonnant de voir ce paramètre maximisé pour la vitesse la plus faible, qui génère les balancements les moins importants.

Il est possible de distinguer ici encore deux comportements mais en fonction de la vitesse de marche cette fois-ci. PCB et ACFH ont vu l'écartement des épaules diminuer grandement avec l'augmentation de la vitesse tandis que celui pour PCL et ACLT est resté élevé dans l'ensemble.

Pour comprendre ce phénomène, il faut considérer les courbes d'évolution du coefficient de frottement maximal nécessaire. La plupart du temps, les courbes bleues et rouges sont en-dessous des courbes vertes, ce qui signifie que l'action conjointe de la présence des membres supérieurs et de leurs mouvements a réduit le coefficient de frottement nécessaire pour ces instants. Mais ce n'est pas

toujours le cas ! Lorsque les courbes bleues et rouges sont au-dessus des courbes vertes, cela signifie l'inverse. Augmenter l'écartement des épaules affecte également ces instants, ce qui peut conduire à une augmentation de la valeur maximale du coefficient de frottement nécessaire et donc à une dépréciation du critère d'optimisation.

C'est donc la combinaison des mouvements des membres supérieurs avec les instants où ils ont lieu qui importe, l'écartement des épaules jouant le rôle de variable d'ajustement. PCL et ACLT en bénéficiaient, PCB et ACFH non.

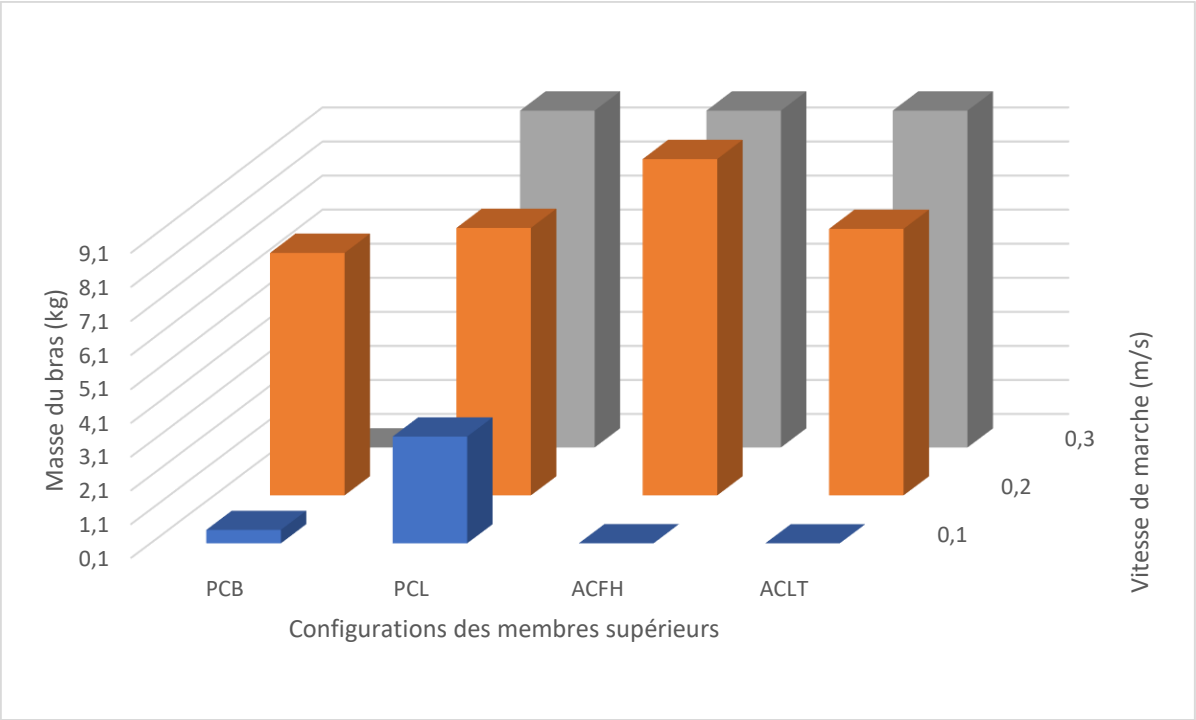


Figure 81

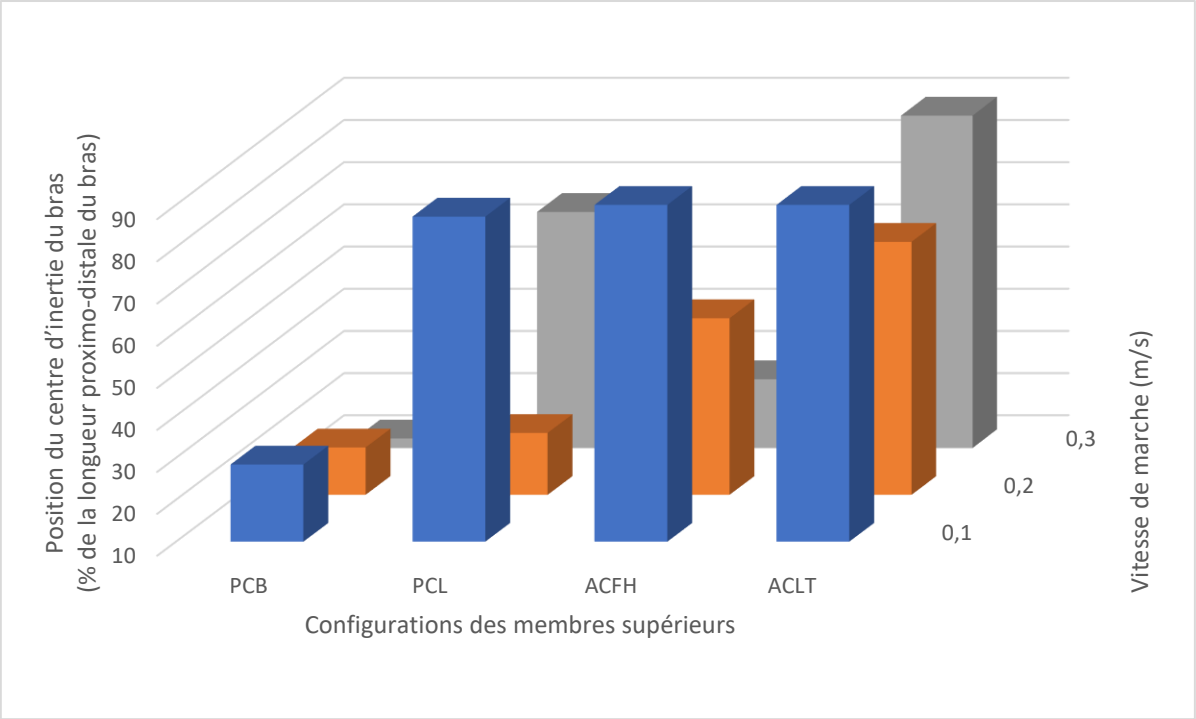


Figure 82

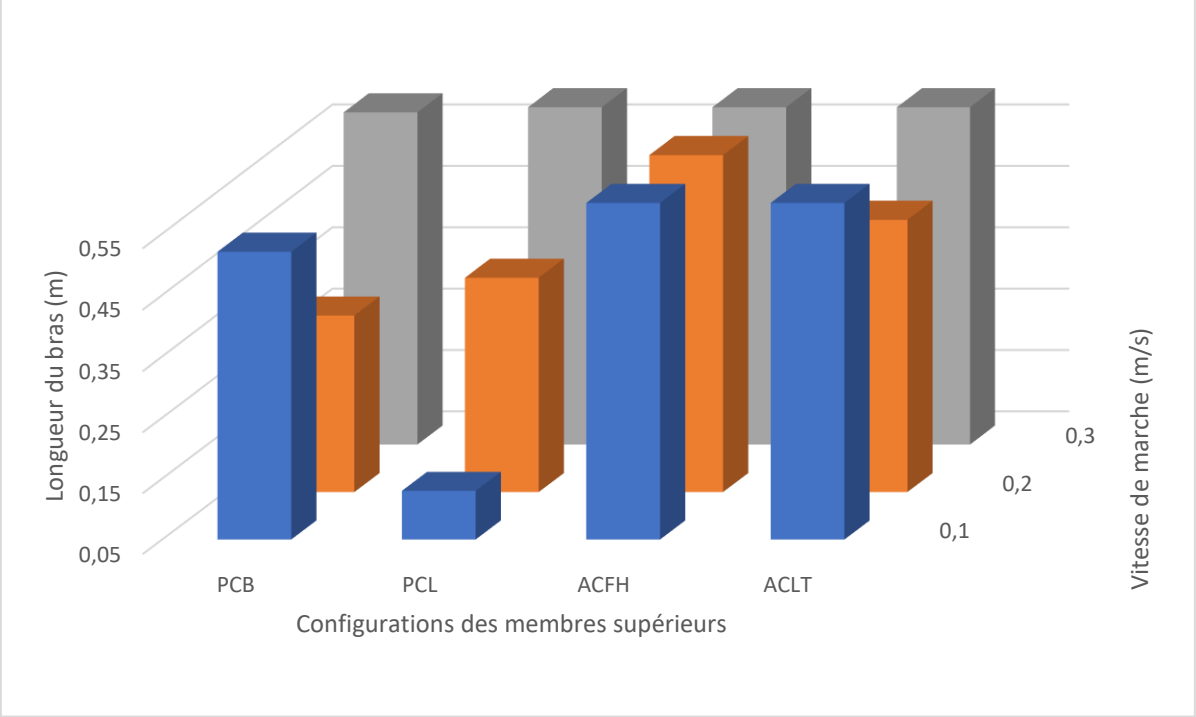


Figure 83

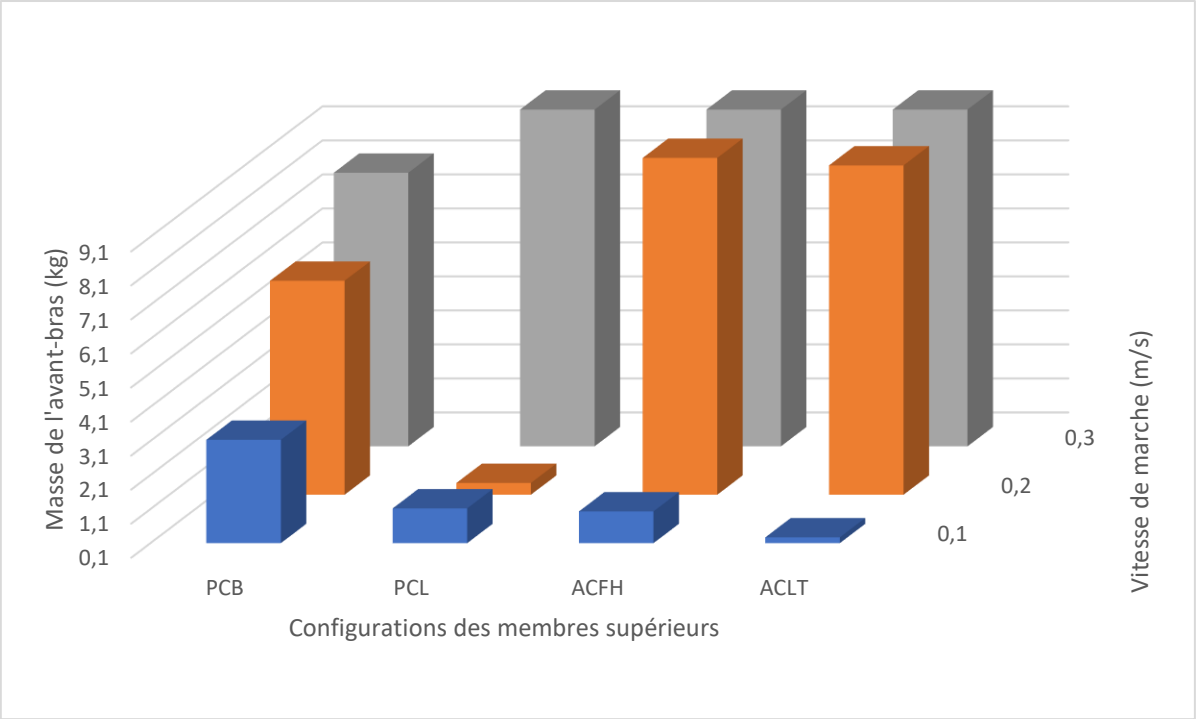


Figure 84

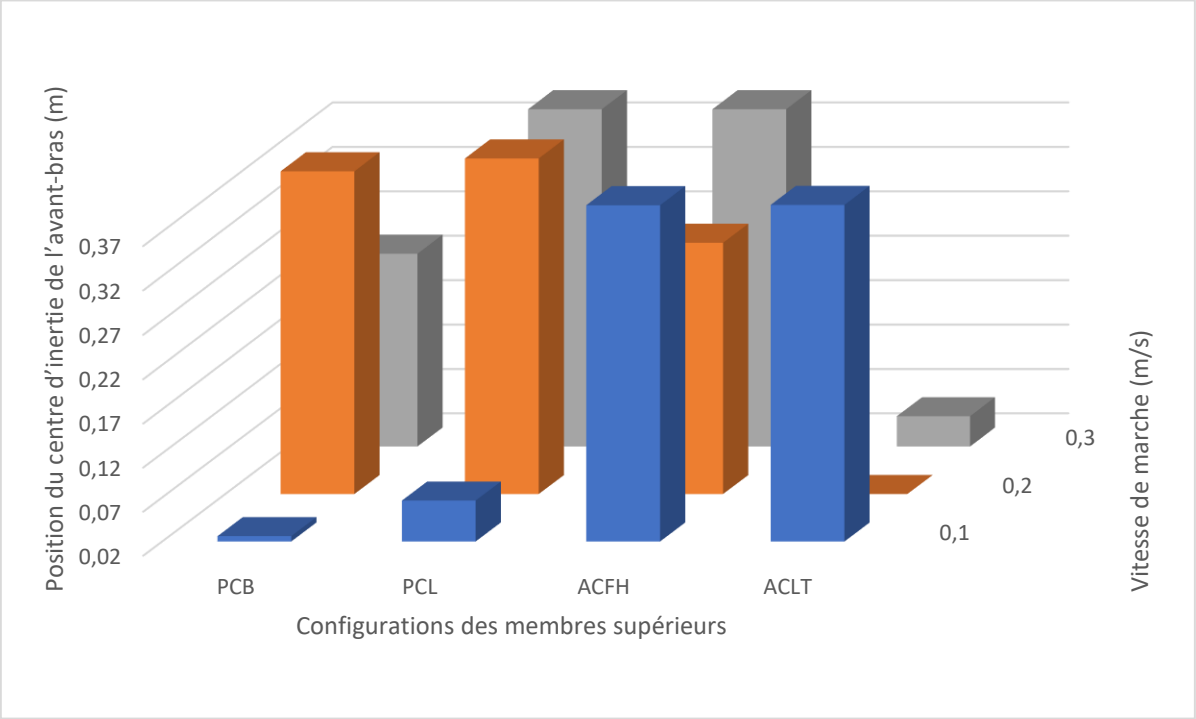


Figure 85

Les cinq paramètres précédents, à savoir les masses et les positions du centre d'inertie des bras et des avant-bras ainsi que la longueur des bras, doivent être considérés ensemble pour interpréter leurs évolutions.

En résumé, plus les valeurs numériques de ces paramètres sont élevées, plus les membres supérieurs et leurs mouvements ont de l'impact.

Les masses en rotation autour d'un axe dont elles sont excentrées, ce qui est le cas ici, génèrent de l'inertie, propriété d'un corps à s'opposer aux changements de sa vitesse. Grossièrement, plus un corps est lourd plus les variations de sa vitesse seront faibles pour une sollicitation mécanique donnée. Les masses jouent en quelques sortes le rôle de stabilisateurs. Il est donc cohérent de constater que, plus la vitesse de marche augmente, plus l'algorithme d'optimisation cherche à alourdir les bras et les avant-bras afin de stabiliser leurs balancements dans les cas passifs. Réciproquement, un corps plus lourd générera des efforts plus grands pour un même mouvement Ainsi, pour tous les cas étudiés, augmenter les masses des membres supérieurs permet de générer plus d'efforts de rotation selon la verticale, pour contrer ceux issus du mouvement des membres inférieurs. PCB est une configuration particulière car il n'y a pas de mouvements relatifs entre le bras et l'avant-bras et il faut donc considérer le barycentre des deux masses pour obtenir celle d'un solide unique équivalent.
En rappelant l'expression du coefficient de frottement en fonction des composantes du torseur de l'action mécanique exercée par le sol sur les pieds

$$\mu = \frac{3 \times L}{2 \times r \times X}$$

il est aisé de constater qu'augmenter les masses des membres supérieurs, donc leurs poids, fera diminuer le coefficient de frottement puisqu'elles sont comprises dans X, réaction verticale du sol qui intègre le poids du robot tout entier.

Les positions des centres d'inertie agissent d'une manière similaire à l'écartement des épaules, c'est-à-dire qu'elles se comportent comme variables d'ajustement. En reprenant la notion d'inertie, une masse donnée en générera davantage plus elle est loin de l'axe de rotation. Les positions des centres d'inertie permettent donc elles-aussi de faire varier les efforts que le balancement des membres supérieurs génère sur le tronc. Le lien entre ces deux paramètres est particulièrement visible pour la plus faible des vitesses de marche : les masses des bras et des avant-bras, petites pour favoriser le mouvement, sont positionnées loin de l'articulation proximale pour augmenter les efforts transmis. Pour la configuration PCB là aussi particulière, puisque les avant-bras sont alignés avec les bras, il faut ajouter la longueur des bras à la position des centres d'inertie des avant-bras.
En revanche et comme pour l'écartement des épaules également, leurs conséquences dépendent de la combinaison des mouvements des membres supérieurs avec les instants où ils ont lieu. En fonction des cas étudiés, elles peuvent être bénéfiques ou non.

La longueur des bras définit la distance entre l'articulation de l'épaule et celle du coude. Plus elle est grande et plus l'avant-bras est éloigné. Étant donné les phénomènes précédemment mentionnés, il est cohérent qu'elle augmente avec la vitesse de marche.

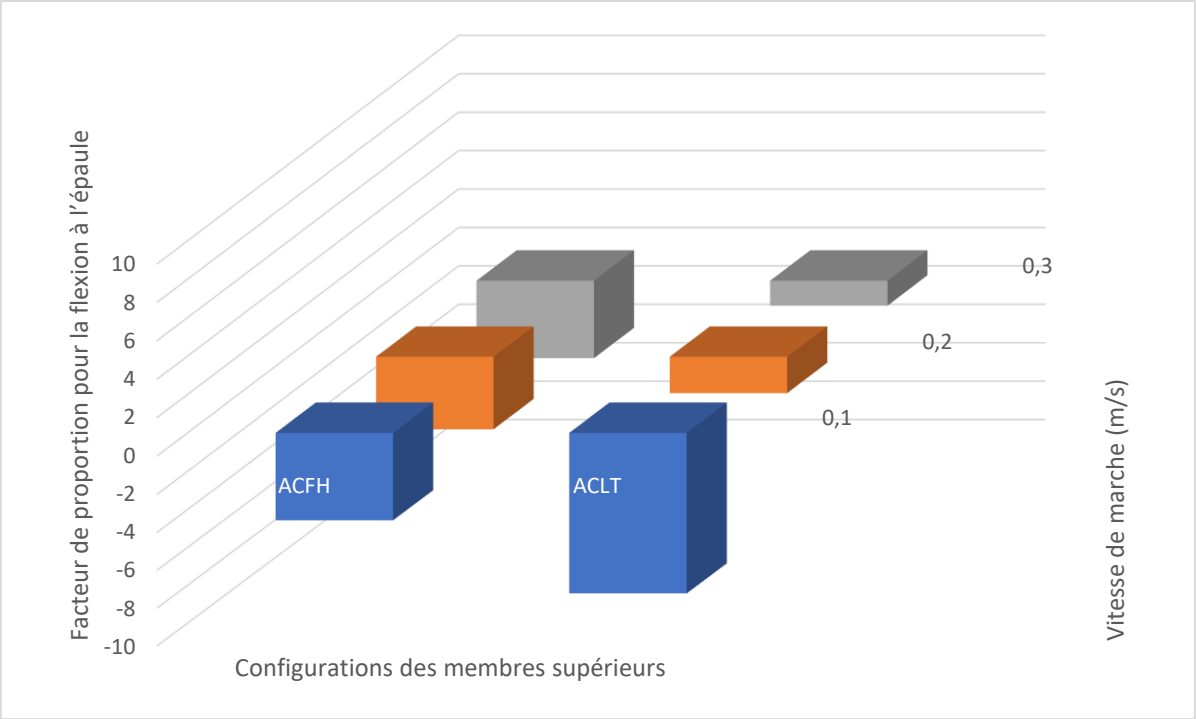


Figure 86

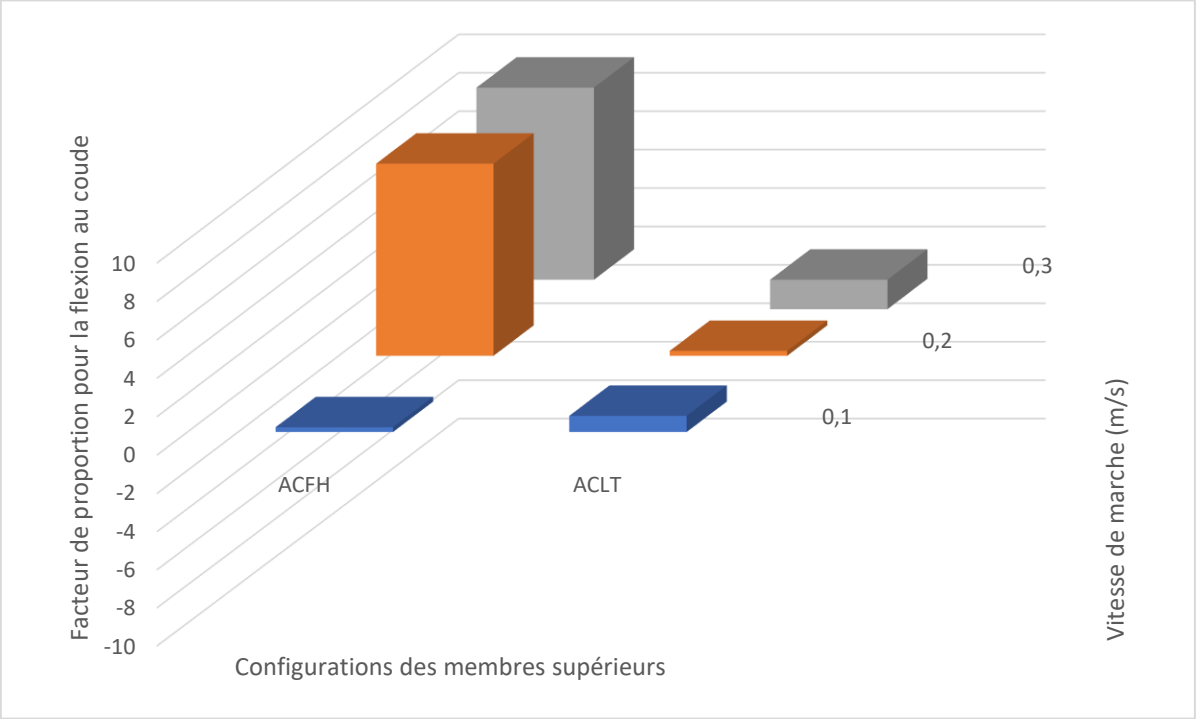


Figure 87

Les facteurs de proportion pour les flexions à l'épaule et au coude ne concernent que les cas où les flexions étaient commandées. Les deux graphiques précédents montrent les effets de l'origine de cette commande.

Les mouvements obtenus sont radicalement différents. Par exemple, pour les flexions pilotées à partir de l'angle de flexion aux hanches (ACFH), la vitesse de marche a peu d'importance pour le facteur de proportion à l'épaule alors qu'elle a un effet notable sur celui au coude. En revanche, lorsque qu'elles sont pilotées à partir de l'angle de lacet du tronc (ACLT), c'est le facteur de proportion à l'épaule qui est sensible à la vitesse de marche tandis que le facteur de proportion au coude l'est beaucoup moins. Ces valeurs posent notamment des questions vis-à-vis des performances quant au pivotement des robots dont le mouvement des membres supérieurs, constitués d'un seul solide, est commandé à partir de l'angle de flexion de hanche (exemple : [27]) au vu des valeurs élevées de flexion au coude pour ACFH qui sont par ailleurs de signe opposé à celles de flexion à l'épaule. Il est bon de préciser néanmoins qu'il s'agit d'une loi de commande très simpliste et que les évolutions de ces angles sont obtenues à partir de la marche singulière d'ORHRO.

Le résultat le plus intéressant est certainement que, pour les deux commandes et bien que le signe du facteur de proportion ne fût pas imposé, l'algorithme d'optimisation a choisi celui qui ferait balancer les membres supérieurs en opposition avec les membres inférieurs.

b. Coefficients de frottement maximaux nécessaires

Le tableau suivant rassemble les valeurs maximales du critère de tous les cas étudiés :

		Vitesse de marche (m/s)		
		0,1	0,2	0,3
Configuration des membres supérieurs	Sans membres supérieurs	0,191	0,463	0,863
	Avec membres supérieurs passifs avec l'articulation du coude bloquée	0,154	0,316	0,484
	Avec membres supérieurs passifs avec l'articulation du coude libre	0,145	0,326	0,461
	Avec membres supérieurs actifs commandés en proportion de l'angle de flexion de la hanche	0,18	0,377	0,694
	Avec membres supérieurs actifs commandés en proportion de l'angle de lacet du tronc	0,191	0,353	0,593

Tableau 40. Tableau des coefficients de frottement maximaux nécessaire pour chaque cas

Deux conclusions peuvent être tirées ce tableau :

1. Commander simplement les flexions aux épaules et aux coudes, à partir d'une proportion de l'angle de flexion aux hanches ou, un peu mieux, de l'angle de lacet du tronc, permet de réduire le coefficient de frottement maximal nécessaire mais pas autant que de laisser les membres supérieurs se balancer librement.

2. Que les membres supérieurs soient composés d'un ou deux solides, un balancement généré de façon passive uniquement, couplé à des paramètres dimensionnels adaptés, permet de diviser par presque deux le coefficient de frottement maximal nécessaire pour des vitesses de marche de 0,3 mètre par seconde.

Cela signifie que le seul fait d'ajouter des doubles pendules aux paramètres optimisés permet déjà de faire grandement diminuer les efforts de rotation selon la verticale à transmettre à l'interface entre les pieds d'un robot bipède et le sol sur lequel il marche.

L'une des prochaines étapes sera de réitérer le même processus d'optimisation mais avec des lois de commande plus complexes, en se basant par exemple sur les mouvements passifs et en modifiant les parties où le coefficient de frottement nécessaire dépasse celui de la version d'ORHRO sans membres supérieurs, de sorte que le balancement des bras ne soit que bénéfique.

c. Écarts entre les positions du PAMB et les références

Configuration des membres supérieurs	Vitesse de marche (m/s)		
	0,1	0,2	0,3
Sans membres supérieurs	7,65	23,87	50,52
Avec membres supérieurs passifs avec l'articulation du coude bloquée	33,5	107,6	189,4
Avec membres supérieurs passifs avec l'articulation du coude libre	45,6	65,63	125,9
Avec membres supérieurs actifs commandés en proportion de l'angle de flexion de la hanche	9,66	76,16	76,61
Avec membres supérieurs actifs commandés en proportion de l'angle de lacet du tronc	7,16	24,6	29,69

Tableau 41. Tableau des distances maximales (en mm) entre le PAMB et sa référence pour chaque cas

Ce tableau vient nuancer le précédent. Car en effet, bien que le coefficient de frottement maximal nécessaire soit bien diminué par l'ajout de membres supérieurs passifs, leurs mouvements incontrôlés dégradent fortement la stabilité, doublant au minimum l'écart maximal entre le PAMB et sa référence pour des vitesses de marche de 0,3 mètre par seconde.

Plus surprenant encore, le balancement actif commandé en proportion de l'angle de lacet du tronc réduit cet écart. Il s'agit de la seule configuration y parvenant. La position du PAMB n'était pourtant qu'un critère de validation *a posteriori* que le robot ne basculait pas, pas un critère d'optimisation.

L'une des prochaines étapes sera d'intégrer la minimisation de l'écart entre la position du PAMB et sa référence au critère d'optimisation.

Les membres supérieurs, s'ils disposent des bonnes propriétés dimensionnelles et si leurs mouvements sont correctement pilotés, ont donc la possibilité de diminuer à la fois le risque de glissement et le risque de basculement d'un humanoïde lors de sa marche. Il sera possible de tester cela expérimentalement car tous les modèles générés par la MDA peuvent être directement utilisés par des automates (cf. partie deux de ce document).

La MDA a donc permis de calculer les modèles dynamiques direct et inverse de membres supérieurs aux paramètres inconnus et de calculer des grandeurs nécessaires à la qualification et à la quantification de la marche.

Cette quatrième partie étant terminée, la partie suivante résumera les travaux présentés dans ce document ainsi que leurs principaux résultats et présentera la feuille de route des prochains développements de l'étude du phénomène du balancement des membres supérieurs lors de la marche humanoïde.

V. Le résumé de ce qui a été fait et de ce qu'il reste à faire

Dans cette cinquième et dernière partie sont rappelées les motivations et les problématiques des recherches présentées dans ce document, les principales avancées réalisées et les derniers jalons à venir.

La première partie a fait un tour d'horizon sur le phénomène du balancement des membres supérieurs lors de la marche humanoïde. Ce mouvement a été décrit et de nombreux travaux à son sujet ont été mentionnés afin de montrer sa complexité. Le fait qu'il ne soit pas nécessaire à la marche suscita des interrogations quant à son existence.
Les premières études portèrent sur ses causes et plus précisément sur la présence d'une activité musculaire des muscles des membres supérieurs, qui signifiait qu'une forme de contrôle était exercée. Ce contrôle n'étant pas continu sur l'ensemble du cycle de marche, d'autres mécanismes devaient exister pour entretenir ce mouvement. C'est ainsi que la question des proportions des parties actives et passives dans le balancement des bras fut formulée. Beaucoup de modèles furent générés pour y répondre, plébiscitant parfois l'une ou l'autre, mais aucun n'y parvint du fait de leurs approches et/ou de leur simplicité, la difficulté de simuler les comportements dynamiques de l'être humain en étant la cause.
Les conséquences de ce mouvement furent également considérées. Les conséquences directes sur les caractéristiques de la marche tout d'abord, telles que sa vitesse, les fréquences et les longueurs des pas et des foulées, puis de manière plus générale les notions d'économie d'énergie, de compensation des efforts de rotation dus aux mouvements des membres inférieurs et de stabilité firent l'objet de publications s'accordant pour la plupart sur le fait que le balancement des bras est bénéfique à la bipédie humanoïde.
Car en effet, cela ne concerne pas que les êtres humains, les robots humanoïdes le sont aussi. Là encore les mêmes thématiques s'y retrouvent avec le sujet en plus du choix de la meilleure stratégie à employer pour le pilotage de ces robots. Ils disposent enfin de deux qualités très importantes : leur modélisation est plus simple que celle des humains et leur comportement est celui de leur commande. Ils constituent dès lors un sujet d'étude de prédilection qui pourrait apporter des réponses transposables aux questions concernant l'humain.
La problématique de la modélisation de l'humanoïde est centrale pour comprendre les origines du balancement des membres supérieurs, dynamiques par nature. Plusieurs outils existants sont nommés, appartenant aux deux grandes familles du numérique et du symbolique possédant chacune ses forces et ses faiblesses. Pour la simulation du phénomène du balancement des bras, aux nombreuses exigences, les modèles symboliques ont été retenus en raison de leur capacité à calculer rapidement et facilement la dynamique de mécanismes aux paramètres méconnus ou inconnus. Aucun outil disponible actuellement ne permettait malheureusement d'effectuer ce travail et la décision fut prise d'en développer un respectant ces exigences.

La seconde partie a décrit la Modélisation Dynamique d'Arborescences (MDA), un générateur symbolique de modèles dynamiques direct et inverse de systèmes mécaniques à chaînes ouvertes. Sans revenir dans le détail sur ses fonctionnalités, les principaux atouts de la MDA sont la flexibilité et la fidélité avec lesquelles un système peut être transcrit ainsi que la grande diversité de grandeurs qu'elle est capable de formuler. Sa polyvalence provient également des fichiers qu'elle produit contenant des expressions formelles ou des calculs. Ces derniers, étant particulièrement rapides puisqu'optimisés en nombre d'opérations et ne dépendant d'aucune bibliothèque extérieure, peuvent être directement utilisés pour piloter des dispositifs robotiques permettant la validation expérimentale des résultats obtenus par la simulation.
Du fait de l'impact de la méthode numérique sur la performance des modèles dynamiques directs, c'est-à-dire le rapport de leurs précisions par leurs coûts calculatoires, une comparaison sur des exemples élémentaires entre des méthodes usuelles et une nouvelle méthode est présentée. Cette dernière propose un nouveau compromis offrant une forte augmentation de la précision pour une faible augmentation des coûts calculatoires.

La troisième partie expose le premier exemple d'application de la MDA, qui concerne la figure de gymnastique du passement filé, réalisée aux barres parallèles et asymétriques. Mouvement particulièrement dynamique, physique et technique, il constituait une étude de choix pour évaluer la capacité de la MDA à modéliser l'être humain.
Deux grandes étapes la constituent : l'identification des valeurs numériques de certains paramètres du mécanisme représentant le gymnaste et la synthèse d'un geste optimisé permettant au gymnaste de réaliser la figure du passement filé en minimisant ses efforts.
Cet exemple a validé l'aptitude de la MDA à déterminer rapidement les paramètres d'un modèle afin que son comportement en simulation corresponde au comportement réel observé. Cela est d'autant plus remarquable étant donnée la complexité dudit modèle, possible grâce à la flexibilité de la méthode de description des arborescences utilisée par la MDA.

La quatrième partie détaille le second exemple d'application de la MDA, qui porte sur le développement d'Open Robotics Humanoid RObot, un robot humanoïde de grande taille, et notamment de sa bipédie. À l'heure actuelle, il n'est doté que d'un tronc et de deux membres inférieurs ce qui fait de lui un parfait candidat à l'étude des implications de l'ajout de membres supérieurs sur la marche.
La MDA a généré les modèles de bras aux paramètres entièrement paramétrables et ce sont les relations entre les valeurs de ces paramètres et deux conditions de stabilité – de non-glissement et de non-basculement – qui furent quantifiées et qualifiées. Quatre mouvements – passif avec l'articulation du coude bloquée, passif avec l'articulation du coude libre, actif commandé en proportion de l'angle de flexion de la hanche et actif commandé en proportion de l'angle de lacet du tronc – ont été simulés pour trois vitesses de marche – 10, 20 et 30 centimètres par seconde.
Les résultats de ces simulations ont non seulement démontré que la MDA permet de calculer la dynamique directe et inverse de très nombreuses versions d'un même modèle mais également de calculer les valeurs d'autres grandeurs nécessaires à l'analyse de la marche. De plus, ces résultats dévoilent de nouvelles pistes pour guider la conception de membres supérieurs les plus bénéfiques possibles à la bipédie humanoïde.

Beaucoup de perspectives de recherche s'ouvrent à la suite des travaux présentés dans ce document. La MDA, en premier lieu, peut encore évoluer et s'enrichir. Proposer toujours plus de façons de décrire les arborescences, en intégrant l'orientation par quaternion par exemple, et permettre de modéliser les arborescences à chaînes fermées sont ses deux prochaines fonctionnalités.
Le modèle de gymnaste utilisé pour l'étude de la figure du passement filé a déjà sa complexification entamée afin de se comporter de manière encore plus proche de la réalité et son usage lors d'entraînements d'athlètes de haut niveau est prévu.
ORHRO devrait voir encore pléthore de versions de membres supérieurs. Que ce soit en modifiant les liaisons les constituant, en augmentant le nombre de leurs paramètres optimisés ou en les intégrant dans la génération des trajectoires de marche, la quantité d'expériences réalisables semble infinie.
Et puisque la MDA a fait ses preuves au travers ces deux applications, l'étude du phénomène de balancement des bras lors de la marche humaine, en particulier par la détermination des proportions des parties active et passive de son apparition, peut débuter. Elle se fera via la conception d'un modèle de membres supérieurs humains aussi fidèle que possible dont les paramètres et la commande seront définis à l'aide de comparaison avec des enregistrements de mouvements réels.

VI. Annexes

Voici la liste des fichiers joints à ce document :

a. La Modélisation Dynamique d'Arborescences, description de l'outil et de son utilisation

Documents :

- V1p85_VIERGE_Description_arborescence.txt
- V1p85_VIERGE_Parametres_utilisateur.txt
- MS_Description_arborescence.txt
- Inverse_MS_MD_COMP.tex
- Inverse_MS_MD_COMP.pdf
- Direct_MS_MD_COMP.tex
- Direct_MS_MD_COMP.pdf

b. La figure du passement filé, exemple d'application à l'étude du mouvement humain

Documents :

- GB12DDLLMSVRD_Description_arborescence.txt
- GS1PBNDRD_Parametres_utilisateur.txt
- GS1R6DDL_Parametres_utilisateur.txt

- GB12DDLLMSVRD_GS1PBNDRD_MD_DPS_CMJB_PPO_Libres_Dictionnaire.txt
- GB12DDLLMSVRD_GS1PBNDRD_MD_DPS_CMJB_PPO_Libres_MATLAB.cpp
- GB12DDLLMSVRD_GS1PBNDRD_MD_APS_CVD_PCLSetCI_R0_Dictionnaire.txt
- GB12DDLLMSVRD_GS1PBNDRD_MD_APS_CVD_PCLSetCI_R0_MATLAB.cpp

- GB12DDLLMSVRD_GS1R6DDL_MD_DPS_CMJB_PPO_Libres_Dictionnaire.txt
- GB12DDLLMSVRD_GS1R6DDL_MD_DPS_CMJB_PPO_Libres_MATLAB.cpp
- GB12DDLLMSVRD_GS1R6DDL_MD_APS_CVD_PCLSetCI_R0_Dictionnaire.txt
- GB12DDLLMSVRD_GS1R6DDL_MD_APS_CVD_PCLSetCI_R0_MATLAB.cpp

Vidéos :

- Gymnaste_Identification_Modele_Capture_Mouvement_Agrandissement.avi
- Gymnaste_Identification_Modele_Capture_Mouvement_Ralenti.avi
- Gymnaste_Identification_Modele_Capture_Mouvement_Ralenti_Agrandissement.avi
- Gymnaste_Geste_Optimise.avi
- Gymnaste_Geste_Optimise_Ralenti.avi

c. La bipédie d'ORHRO, exemple d'application à la robotique

Documents :

- ORHROV1p5TetMI_Description_arborescence.txt
- ORHROV1p5MSG_Description_arborescence.txt
- ORHROV1p5MSD_Description_arborescence.txt
- PBSR2DDL_Parametres_utilisateur.txt

- ORHROV1p5TetMI_PBSR2DDL_MD_APS_TAMGAMPI_Dictionnaire.txt
- ORHROV1p5TetMI_PBSR2DDL_MD_APS_TAMGAMPI_MATLAB.cpp
- ORHROV1p5TetMI_PBSR2DDL_MD_APS_CVD_PLSetI_R0_Dictionnaire.txt
- ORHROV1p5TetMI_PBSR2DDL_MD_APS_CVD_PLSetI_R0_MATLAB.cpp

- ORHROV1p5MSG_PBSR2DDL_MD_DPS_CMJB_MouvementBrasGauche_Dictionnaire.txt
- ORHROV1p5MSG_PBSR2DDL_MD_DPS_CMJB_MouvementBrasGauche_MATLAB.cpp
- ORHROV1p5MSG_PBSR2DDL_MD_DPS_CE_AccArticulationsBrasGauche_Dictionnaire.txt
- ORHROV1p5MSG_PBSR2DDL_MD_DPS_CE_AccArticulationsBrasGauche_MATLAB.cpp
- ORHROV1p5MSG_PBSR2DDL_MD_IPS_TAML_EpauleGauche_Dictionnaire.txt
- ORHROV1p5MSG_PBSR2DDL_MD_IPS_TAML_EpauleGauche_MATLAB.cpp
- ORHROV1p5MSG_PBSR2DDL_MD_APS_CVD_PLSetI_R0_Dictionnaire.txt
- ORHROV1p5MSG_PBSR2DDL_MD_APS_CVD_PLSetI_R0_MATLAB.cpp

- ORHROV1p5MSD_PBSR2DDL_MD_DPS_CMJB_MouvementBrasDroit_Dictionnaire.txt
- ORHROV1p5MSD_PBSR2DDL_MD_DPS_CMJB_MouvementBrasDroit_MATLAB.cpp
- ORHROV1p5MSD_PBSR2DDL_MD_DPS_CE_AccArticulationsBrasDroit_Dictionnaire.txt
- ORHROV1p5MSD_PBSR2DDL_MD_DPS_CE_AccArticulationsBrasDroit_MATLAB.cpp
- ORHROV1p5MSD_PBSR2DDL_MD_IPS_TAML_EpauleDroit_Dictionnaire.txt
- ORHROV1p5MSD_PBSR2DDL_MD_IPS_TAML_EpauleDroit_MATLAB.cpp
- ORHROV1p5MSD_PBSR2DDL_MD_APS_CVD_PLSetI_R0_Dictionnaire.txt
- ORHROV1p5MSD_PBSR2DDL_MD_APS_CVD_PLSetI_R0_MATLAB.cpp

Vidéos :

- ORHRO_PCB01.avi
- ORHRO_PCB02.avi
- ORHRO_PCB03.avi
- ORHRO_PCL01.avi
- ORHRO_PCL02.avi
- ORHRO_PCL03.avi
- ORHRO_ACFH01.avi

- ORHRO_ACFH02.avi
- ORHRO_ACFH03.avi
- ORHRO_ACLT01.avi
- ORHRO_ACLT02.avi
- ORHRO_ACLT03.avi

VII. Bibliographie

[1] M. P. Murray, S. B. Sepic, et E. J. Barnard, « Patterns of Sagittal Rotation of the Upper Limbs in Walking », *Phys. Ther.*, vol. 47, n° 4, p. 272-284, avr. 1967, doi: 10.1093/ptj/47.4.272.

[2] M. L. F. Ballesteros, F. Buchthal, et P. Rosenfalck, « The Pattern of Muscular Activity During the Arm Swing of Natural Walking », *Acta Physiol. Scand.*, vol. 63, n° 3, p. 296-310, 1965, doi: 10.1111/j.1748-1716.1965.tb04069.x.

[3] J. V. Basmajian et R. Tuttle, « EMG of Locomotion in Gorilla and Man », in *Control of Posture and Locomotion*, R. B. Stein, K. G. Pearson, R. S. Smith, et J. B. Redford, Éd. Boston, MA: Springer US, 1973, p. 599-609. doi: 10.1007/978-1-4613-4547-3_46.

[4] K. M. Jackson, J. Joseph, et S. J. Wyard, « A mathematical model of arm swing during human locomotion », *J. Biomech.*, vol. 11, n° 6-7, p. 277-289, 1978, doi: 10.1016/0021-9290(78)90061-1.

[5] K. M. Jackson, « Why the upper limbs move during human walking », *J. Theor. Biol.*, vol. 105, n° 2, p. 311-315, janv. 1983, doi: 10.1016/S0022-5193(83)80010-1.

[6] S. Eke-Okoro, M. Gregoric, et L. Larsson, « Alterations in gait resulting from deliberate changes of arm-swing amplitude and phase », *Clin. Biomech.*, vol. 12, n° 7, p. 516-521, oct. 1997, doi: 10.1016/S0268-0033(97)00050-8.

[7] S. Topuz, E. Kirdi, A. I. Yalcin, O. Ulger, H. Keklicek, et G. Sener, « Effects of arm swing on spatiotemporal characteristics of gait in unilateral transhumeral amputees », *Gait Posture*, vol. 68, p. 95-100, févr. 2019, doi: 10.1016/j.gaitpost.2018.11.010.

[8] J. Park, « Synthesis of natural arm swing motion in human bipedal walking », *J. Biomech.*, vol. 41, n° 7, p. 1417-1426, janv. 2008, doi: 10.1016/j.jbiomech.2008.02.031.

[9] B. R. Umberger, « Effects of suppressing arm swing on kinematics, kinetics, and energetics of human walking », *J. Biomech.*, vol. 41, n° 11, p. 2575-2580, août 2008, doi: 10.1016/j.jbiomech.2008.05.024.

[10] Z. Yizhar, S. Boulos, O. Inbar, et E. Carmeli, « The effect of restricted arm swing on energy expenditure in healthy men », *Int. J. Rehabil. Res.*, vol. 32, n° 2, p. 115-123, juin 2009, doi: 10.1097/MRR.0b013e32830d3675.

[11] H. Pontzer, J. H. Holloway, D. A. Raichlen, et D. E. Lieberman, « Control and function of arm swing in human walking and running », *J. Exp. Biol.*, vol. 212, n° 4, p. 523-534, févr. 2009, doi: 10.1242/jeb.024927.

[12] S. H. Collins, P. G. Adamczyk, et A. D. Kuo, « Dynamic arm swinging in human walking », *Proc. R. Soc. B Biol. Sci.*, vol. 276, n° 1673, p. 3679-3688, oct. 2009, doi: 10.1098/rspb.2009.0664.

[13] D. Barthelemy et J. B. Nielsen, « Corticospinal contribution to arm muscle activity during human walking », *J. Physiol.*, vol. 588, n° 6, p. 967-979, mars 2010, doi: 10.1113/jphysiol.2009.185520.

[14] J. P. Kuhtz-Buschbeck et B. Jing, « Activity of upper limb muscles during human walking », *J. Electromyogr. Kinesiol.*, vol. 22, n° 2, p. 199-206, avr. 2012, doi: 10.1016/j.jelekin.2011.08.014.

[15] M. Goudriaan, I. Jonkers, J. H. van Dieen, et S. M. Bruijn, « Arm swing in human walking: What is their drive? », *Gait Posture*, vol. 40, n° 2, p. 321-326, juin 2014, doi: 10.1016/j.gaitpost.2014.04.204.

[16] « Getting Started - OpenSim Documentation - Site global ». https://simtk-confluence.stanford.edu:8443/display/OpenSim/Getting+Started (consulté le 30 octobre 2022).

[17] B. Hejrati, S. Chesebrough, K. Bo Foreman, J. J. Abbott, et A. S. Merryweather, « Comprehensive quantitative investigation of arm swing during walking at various speed and surface slope conditions », *Hum. Mov. Sci.*, vol. 49, p. 104-115, oct. 2016, doi: 10.1016/j.humov.2016.06.001.

[18] S. Canton et M. J. MacLellan, « Active and passive contributions to arm swing: Implications of the restriction of pelvis motion during human locomotion », *Hum. Mov. Sci.*, vol. 57, p. 314-323, févr. 2018, doi: 10.1016/j.humov.2017.09.009.

[19] J. B. Weersink, N. M. Maurits, et B. M. de Jong, « EEG time-frequency analysis provides arguments for arm swing support in human gait control », *Gait Posture*, vol. 70, p. 71-78, mai 2019, doi: 10.1016/j.gaitpost.2019.02.017.

[20] P. Meyns, S. M. Bruijn, et J. Duysens, « The how and why of arm swing during human walking », *Gait Posture*, vol. 38, n° 4, p. 555-562, sept. 2013, doi: 10.1016/j.gaitpost.2013.02.006.

[21] J. Romkes et K. Bracht-Schweizer, « The effects of walking speed on upper body kinematics during gait in healthy subjects », *Gait Posture*, vol. 54, p. 304-310, mai 2017, doi: 10.1016/j.gaitpost.2017.03.025.

[22] S. M. Bruijn, O. G. Meijer, P. J. Beek, et J. H. van Dieën, « The effects of arm swing on human gait stability », *J. Exp. Biol.*, vol. 213, n° 23, p. 3945-3952, déc. 2010, doi: 10.1242/jeb.045112.

[23] K. B. Cheng, Y.-C. Huang, et S.-Y. Kuo, « Effect of arm swing on single-step balance recovery », *Hum. Mov. Sci.*, vol. 38, p. 173-184, déc. 2014, doi: 10.1016/j.humov.2014.08.011.

[24] K. B. Cheng, K.-M. Wang, et S.-Y. Kuo, « Role of arm motion in feet-in-place balance recovery », *J. Biomech.*, vol. 48, n° 12, p. 3155-3162, sept. 2015, doi: 10.1016/j.jbiomech.2015.07.008.

[25] B. Kaddar, Y. Aoustin, et C. Chevallereau, « Arm swing effects on walking bipedal gaits composed of impact, single and double support phases », *Robot. Auton. Syst.*, vol. 66, p. 104-115, avr. 2015, doi: 10.1016/j.robot.2014.12.005.

[26] C. G. Atkeson *et al.*, « Using humanoid robots to study human behavior », *IEEE Intell. Syst.*, vol. 15, n° 4, p. 46-56, juill. 2000, doi: 10.1109/5254.867912.

[27] S. H. Collins et A. Ruina, « A Bipedal Walking Robot with Efficient and Human-Like Gait », in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, avr. 2005, p. 1983-1988. doi: 10.1109/ROBOT.2005.1570404.

[28] K. Osaku, H. Minakata, et S. Tadakuma, « A study of CPG based walking utilizing swing of arms », in *9th IEEE International Workshop on Advanced Motion Control, 2006.*, mars 2006, p. 392-396. doi: 10.1109/AMC.2006.1631691.

[29] Y. Aoustin et A. M. Formal'skii, « On optimal swinging of the biped arms », in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, sept. 2008, p. 2922-2927. doi: 10.1109/IROS.2008.4650725.

[30] L. Boutin, A. Eon, S. Zeghloul, et P. Lacouture, « From human motion capture to humanoid locomotion imitation Application to the robots HRP-2 and HOAP-3 », *Robotica*, vol. 29, p. 325-334, mars 2011, doi: 10.1017/S0263574710000172.

[31] S. Zhang *et al.*, « The Mechanism of Yaw Torque Compensation in the Human and Motion Design for Humanoid Robots », *Int. J. Adv. Robot. Syst.*, janv. 2013, doi: 10.5772/54211.

[32] « Havok ». https://www.havok.com/ (consulté le 30 octobre 2022).

[33] « DART: Dynamic Animation and Robotics Toolkit ». http://dartsim.github.io/ (consulté le 30 octobre 2022).

[34] « NVIDIA PhysX 4.5 and 5.0 SDK », *NVIDIA Developer*, 28 novembre 2018. https://developer.nvidia.com/physx-sdk (consulté le 30 octobre 2022).

[35] J. Carpentier *et al.*, « The Pinocchio C++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives », in *2019 IEEE/SICE International Symposium on System Integration (SII)*, janv. 2019, p. 614-619. doi: 10.1109/SII.2019.8700380.

[36] « Overview — MuJoCo documentation ». https://mujoco.readthedocs.io/en/latest/overview.html (consulté le 30 octobre 2022).

[37] M. L. Felis, « RBDL: an efficient rigid-body dynamics library using recursive algorithms », *Auton. Robots*, vol. 41, n° 2, p. 495-511, févr. 2017, doi: 10.1007/s10514-016-9574-0.

[38] « Open Dynamics Engine ». http://ode.org/ (consulté le 30 octobre 2022).

[39] M. A. Sherman, A. Seth, et S. L. Delp, « Simbody: multibody dynamics for biomedical research », *Procedia IUTAM*, vol. 2, p. 241-261, janv. 2011, doi: 10.1016/j.piutam.2011.04.023.

[40] M. Racković, M. Vukobratović, et D. Surla, « Generation of dynamic models of complex robotic mechanisms in symbolic form », *Robotica*, vol. 16, n° 1, p. 23-36, janv. 1998, doi: 10.1017/S0263574798000125.

[41] J. McPhee, C. Schmitke, et S. Redmond, « Dynamic Modelling of Mechatronic Multibody Systems With Symbolic Computing and Linear Graph Theory », *Math. Comput. Model. Dyn. Syst.*, vol. 10, n° 1, p. 1-23, mars 2004, doi: 10.1080/13873950412331318044.

[42] E. Dean-Leon, S. Nair, et A. Knoll, « User friendly Matlab-toolbox for symbolic robot dynamic modeling used for control design », in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, déc. 2012, p. 2181-2188. doi: 10.1109/ROBIO.2012.6491292.

[43] H. Dallali *et al.*, « Development of a dynamic simulator for a compliant humanoid robot based on a symbolic multibody approach », in *2013 IEEE International Conference on Mechatronics (ICM)*, févr. 2013, p. 598-603. doi: 10.1109/ICMECH.2013.6519110.

[44] N. Docquier, A. Poncelet, et P. Fisette, « ROBOTRAN: a powerful symbolic gnerator of multibody models », *Mech. Sci.*, vol. 4, n° 1, p. 199-219, mai 2013, doi: 10.5194/ms-4-199-2013.

[45] W. Khalil, A. Vijayalingam, B. Khomutenko, I. Mukhanov, P. Lemoine, et G. Ecorchard, « OpenSYMORO: An open-source software package for symbolic modelling of robots », in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, juill. 2014, p. 1206-1211. doi: 10.1109/AIM.2014.6878246.

[46] « optimize - Maple Help ». https://de.maplesoft.com/support/help/Maple/view.aspx?path=codegen/optimize (consulté le 2 novembre 2022).

[47] P. L. Davidson, S. J. Wilson, B. D. Wilson, et D. J. Chalmers, « Estimating subject-specific body segment parameters using a 3-dimensional modeller program », *J. Biomech.*, vol. 41, n° 16, p. 3506-3510, déc. 2008, doi: 10.1016/j.jbiomech.2008.09.021.

[48] P. de Leva, « Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters », *J. Biomech.*, vol. 29, n° 9, p. 1223-1230, sept. 1996, doi: 10.1016/0021-9290(95)00178-6.

[49] M. J. Hiley et M. R. Yeadon, « Optimum Technique for Generating Angular Momentum in Accelerated Backward Giant Circles Prior to a Dismount », *J. Appl. Biomech.*, vol. 19, n° 2, p. 119-130, mai 2003, doi: 10.1123/jab.19.2.119.

[50] S. Mirjalili, S. M. Mirjalili, et A. Lewis, « Grey Wolf Optimizer », *Adv. Eng. Softw.*, vol. 69, p. 46-61, mars 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[51] M. J. Hiley et M. R. Yeadon, « Optimization of Backward Giant Circle Technique on the Asymmetric Bars », *J. Appl. Biomech.*, vol. 23, n° 4, p. 300-308, nov. 2007, doi: 10.1123/jab.23.4.300.

[52] R. L. Williams, « Simplified Robotics Joint-Space Trajectory Generation with a via Point Using a Single Polynomial », *J. Robot.*, vol. 2013, p. e735958, févr. 2013, doi: 10.1155/2013/735958.

[53] « Find minimum of constrained nonlinear multivariable function - MATLAB fmincon - MathWorks France ». https://fr.mathworks.com/help/optim/ug/fmincon.html (consulté le 28 octobre 2022).

[54] E. Petković, S. Veličković, E. Kolar, R. Stanković, et D. Stanković, « THE OPTIMAL KINEMATIC MODEL OF THE PERFORMANCE OF THE CLEAR HIP CIRCLE TO HANDSTAND ON THE UNEVEN BARS – A CASE STUDY », *Facta Univ. Ser. Phys. Educ. Sport*, n° 0, Art. n° 0, nov. 2018, doi: 10.22190/FUPES170314021P.

[55] D. J. Lura, S. L. Carey, et R. V. Dubey, « Validation of Functional Methods for Calculating Shoulder Joint Centers Using 3D Motion Analysis », présenté à ASME 2010 International Mechanical Engineering Congress and Exposition, avr. 2012, p. 577-581. doi: 10.1115/IMECE2010-40386.

[56] A. Eon, J. Gastebois, P. Laguillaumie, P. Vulliez, P. Seguin, et J. P. Gazeau, « ORHRO: OPEN Robotics Humanoid RObot - from a mechanical retrofit to a new real time industrial controller based walking robot », *Comput. Methods Biomech. Biomed. Engin.*, vol. 22, n° sup1, p. S502-S504, mai 2019, doi: 10.1080/10255842.2020.1714997.

[57] A. Eon, « Contribution à la génération de mouvements dynamiques pour les robots humanoïdes et au dimensionnement de leurs actionneurs », thesis, Poitiers, 2009. Consulté le: 7 novembre 2019. [En ligne]. Disponible sur: http://www.theses.fr/2009POIT2337

[58] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, et H. Hirukawa, « The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation », in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, oct. 2001, vol. 1, p. 239-246 vol.1. doi: 10.1109/IROS.2001.973365.

[59] *ORHRO : The human sized french humanoid Robot*, (6 novembre 2019). Consulté le: 31 octobre 2022. [En ligne Vidéo]. Disponible sur: https://www.youtube.com/watch?v=iXL29vLakfI

[60] P. Sardain et G. Bessonnet, « Forces acting on a biped robot. Center of pressure-zero moment point », *IEEE Trans. Syst. Man Cybern. - Part Syst. Hum.*, vol. 34, n° 5, p. 630-637, sept. 2004, doi: 10.1109/TSMCA.2004.832811.

VIII. Productions scientifiques de l'auteur pendant la thèse :

J.-B. Riccoboni, T. Monnet, A. Eon, P. Lacouture, J.-P. Gazeau, et M. Campone, « Biomechanical comparison between manual and motorless device assisted patient handling: sitting to and from standing position », *Appl. Ergon.*, vol. 90, p. 103284, janv. 2021, doi: 10.1016/j.apergo.2020.103284.

J.-B. Riccoboni, T. Monnet, A. Eon, "Tree structure dynamics modelling", JNRH 2021, 17 et 18 juin 2021, Angers, France, https://jnrh2021.sciencesconf.org/program/graphic/date/2021-06-18

J.-B. Riccoboni, T. Monnet, A. Eon, « Modélisation Dynamique d'Arborescences », Workshop Réseau Régional de Recherche en Robotique Nouvelle-Aquitaine, 8 et 9 novembre 2021, Bidart, France, https://r4-robotique.fr/workshop

J.-B. Riccoboni, T. Monnet, A. Eon, « Modélisation Dynamique d'Arborescences », Dépôt APP numéro IDDN .FR .001 .160032 .000 .S .P .2022 .000 .31235 au nom du CNRS, 21 avril 2022.

Modélisation dynamique d'arborescence – application à la robotique et à l'étude du mouvement humain

Bien qu'un consensus sur les conséquences du balancement des bras dans la marche humaine semble s'établir, ce n'est pas encore le cas pour ses causes. Ceci est principalement dû à la difficulté de dissocier les parties passives et actives de la génération du balancement des bras, notamment liée à la complexité de l'identification des paramètres nécessaires à la modélisation du corps humain. L'hypothèse que l'étude de la bipédie d'un robot humanoïde, plus simple, pourrait apporter des éléments de réponse à la problématique est le point de départ de cette thèse. Par ailleurs, une optimisation dimensionnelle des bras des robots humanoïdes permettrait d'améliorer la performance dynamique de leur marche et constituerait une aide à la conception.

La conduite de ces deux études nécessite de disposer d'un modèle de dynamique directe et inverse permettant la modification à la volée des paramètres inertiels et géométriques afin de pouvoir qualifier leurs impacts. Compte tenu de son utilisation dans les contextes d'identification et d'optimisation, l'évaluation d'un tel modèle doit être la plus rapide possible pour considérer de grands nombres de configurations dans des temps de simulation raisonnables et ainsi pouvoir accorder de la confiance aux valeurs obtenues.

Un algorithme de génération de modèles dynamiques symboliques pour les structures polyarticulées ouvertes répondant à ces exigences a donc été écrit et testé sur plusieurs exemples dont deux sont présentés dans cette thèse :

- La figure du passement filé effectuée par une gymnaste à la barre, dont la réalisation est très sensible à la finesse du modèle et à la précision des calculs, afin d'évaluer la fidélité du modèle en comparaison à une référence mesurée optoélectroniquement.

- La marche du robot humanoïde ORHRO, équipé en simulation de bras aux paramètres modifiables, afin d'étudier le balancement des bras généré passivement par le déplacement de la partie locomotrice ainsi que l'impact de ce balancement sur des critères de stabilité et de déterminer les valeurs optimales des paramètres de conception.

Cet algorithme de modélisation dynamique d'arborescence a montré son efficacité à modéliser et simuler des mouvements dynamiques sur des architectures complexes et variées permettant son utilisation dans la synthèse de mouvement et l'optimisation dimensionnelle de mécanismes.

Tree structure dynamics modelling – application to robotics et to human movement study

Summary proposal: Although there seems to be a consensus on the consequences of arm swing in human walking, this is not yet the case for its causes. This is mainly due to the difficulty of dissociating the passive and active parts of the generation of the arm swing, notably linked to the complexity of identifying the parameters necessary to model the human body. The starting point of this work is the hypothesis that the study of the bipedalism of a humanoid robot, which is simpler, could provide some answers to the problem. In addition, a dimensional optimization of the arms of humanoid robots would improve the dynamic performance of their walking and would constitute an assistance to design.

The conduct of these two studies requires a direct and inverse dynamics model allowing the modification on the fly of inertial and geometrical parameters in order to qualify their impacts. Given its use in identification and optimization contexts, the evaluation of such a model must be as fast as possible in order to consider large numbers of configurations in reasonable simulation times and thus be able to give confidence to the values obtained.

A symbolic dynamic model generation algorithm for open polyarticulated structures meeting these requirements was therefore written and tested on several examples, two of which are presented in this thesis:

- The clear hip circle figure performed by a gymnast at the bar, which execution is very sensitive to the model fineness and to the precision of the calculations, in order to evaluate the fidelity of the model in comparison to an optoelectronically measured reference.

- The walking of the humanoid robot ORHRO, equipped in simulation with arms with editable parameters, in order to study the arm swing generated passively by the displacement of the locomotor part as well as the impact of this swing on stability criteria and to determine the optimal values of the design parameters.

This tree structure dynamics modelling algorithm has shown its effectiveness in modelling and simulating dynamic movements on complex and varied architectures allowing its use in motion synthesis and dimensional optimization of mechanisms.