

Étude de stratégies d'optimisation pour la prédiction de mouvement de personnages virtuels

Nicolas DESILETS

Encadré par Nicolas PRONOST et Thomas BONIS

Université Claude Bernard Lyon 1, France

Résumé Depuis plusieurs années, les domaines de l'intelligence artificielle et de la modélisation 3D ne cessent de se mêler, en proposant de nouvelles solutions innovantes pour la simulation. Ce travail vise à les étudier, puis explorer l'une de ces solutions, en mettant en place un environnement de simulation simplifié dédié à la prédiction de mouvements. Nous essayerons d'abord de générer de la façon la plus fidèle à la réalité un mouvement à partir de contraintes définies, puis on étudiera ce mouvement dans le cas de certaines pathologies.

Mots-clés: *Simulation prédictive, Apprentissage par renforcement profond, Proximal Policy Optimization, Modèle musculaire, Modèle articulaire, Optimisation*

Abstract. For years, artificial intelligence and 3D modelling have been getting closer and closer, always improving by coming up with innovative solutions for simulation. This work aims to study them, then explore one of them, by building a simplified simulation environment dedicated to movement prediction. We first try to generate a movement in the most realistic way from predefined constraints, then we study this movement in the context of known pathologies.

Keywords: *Predictive simulation, Deep reinforcement learning, Proximal Policy Optimization, Muscular model, Articular model, Optimization*

1 Introduction

Ce document s'inscrit dans le cadre de la formation du département informatique de l'université Lyon 1, M2 Intelligence Artificielle, en tant de rapport de stage de fin de master.

Il permettra de présenter le projet qui m'a été confié au cours de mon stage, et de détailler les différents travaux et tests effectués, ainsi et que les résultats obtenus.

Ce stage a été réalisé au sein du Laboratoire d'Informatique en Image et Systèmes d'information (LIRIS), et a été supervisé par Nicolas PRONOST et Thomas BONIS. Il a pour sujet l'"Étude de stratégies d'optimisation pour la prédiction de mouvement de personnages virtuels".

Le projet de stage s'inscrit dans le cadre du projet de recherche OMEGA, un projet international en coopération avec un laboratoire de bio-mécanique allemand.

1.1 Objectifs

L'objectif principal du projet OMEGA est la conception et la réalisation d'un contrôleur de mouvement humain, pour la simulation prédictive de marches pathologiques de différentes natures. Le but de cette simulation est, à partir d'un modèle simulé et de variables initiales connues (comme par exemple la présence d'une certaine pathologie qui définit l'état du patient), de rendre possible la prédiction de l'évolution d'un mouvement précis : en l'occurrence celui d'une marche humaine. Ainsi, il devient possible de prédire l'évolution de la pathologie en question et d'adapter au mieux les traitements ou opérations à envisager pour soigner cette pathologie.

On peut très facilement imaginer l'intégration de ce genre d'outil prédictif en tant qu'aide dans le processus de prise de décision d'un médecin. Ce projet de stage est donc profondément ancré dans le secteur de la médecine, et profondément lié aux différentes pathologies qu'on désire simuler.

Trois pathologies sont principalement étudiées dans le cadre du projet OMEGA:

- Défaillance de flexion du genou, qui se manifeste par une impossibilité du patient à effectuer le mouvement complet de flexion-extension du genou (les positions maximales de l'articulation ne peuvent pas être atteintes).
- "Drop-foot pathology", une défaillance musculaire au niveau de la cheville qui altère grandement la façon dont le pied entre en contact avec le sol.
- Amputation au niveau du fémur

Ces pathologies seront également au coeur du projet de stage, en tant que pathologies de référence pour les différentes simulations.

1.2 Projet de stage

L'objectif principal du projet est de faire de prédiction de mouvement, ou plus précisément de prédire l'évolution d'un mouvement défini selon certaines contraintes, et de comparer et explorer différentes techniques d'apprentissage permettant cette prédiction.

Dans le cadre de ce projet, on utilisera différents modèles 3D permettant de simuler des mouvements humains réalistes ("réaliste" signifie ici des mouvements pouvant être reproduits par un être humain réel, et idéalement simulés de manière naturelle et fluide). Deux moteurs de simulation seront évoqués au cours de ce rapport, mais un seul sera retenu pour des raisons qui seront explicitées ultérieurement.

L'idée ici est de travailler sur des modèles suffisamment réalistes pour évaluer l'efficacité des différentes méthodes de prédiction, tout en faisant abstraction de certains éléments qui peuvent nuire à la simplicité de la modélisation et à la mise en place d'un environnement de test exploitable. On pensera notamment aux modèles de collision qui peuvent introduire des sources d'erreurs indépendantes des méthodes de prédiction.

Pour cela, on choisira un modèle qui ne sera pas affecté par les contacts: ce modèle sera défini en détail dans une partie dédiée.

À terme, l'objectif du projet est de réussir à définir et extraire des pistes satisfaisantes, ou à minima intéressantes, dans une optique d'exploration dans le cadre du projet OMEGA. Ces pistes seront analysées en conclusion de ce rapport en adéquation avec les résultats obtenus.

2 État de l'art

Le domaine de la simulation de systèmes musculo-articulaires est un domaine qui est étudié depuis plusieurs dizaines d'années. La simulation basée sur des modèles musculaires en particulier semble de plus en plus prédominante, de part leurs résultats prometteurs en matière de réalisme, ce qui est intéressant dans le domaine de l'animation 3D. De nombreuses publications se sont intéressées à produire des environnements de simulation spécifiques, en particulier au niveau du type de modèle utilisé. Entre autres, on retrouve des études sur des simulations musculaires pour l'animation faciale [2], de la main [8], ou encore du cou [9]

L'intérêt de ce type de projet se limitant à un modèle cible très précis permet de se concentrer sur un aspect précis de la simulation.

C'est également l'idée derrière ce projet: En simplifiant le modèle, on peut étudier avec précision l'application des différentes techniques d'intelligence artificielle sur notre système.

L'utilisation de techniques provenant du domaine de l'intelligence artificielle dans le domaine de la simulation est également de plus en plus courante. De la même façon que l'intelligence artificielle s'est imposée dans d'autres domaines comme solution efficace pour la résolution de problème, elle fait de même ici en se montrant capable de résoudre des ensembles de contraintes physiques lors des simulations.

De nombreuses publications ont été faites sur l'utilisation de l'IA sur des modèles articulaires et plus spécifiquement des algorithmes d'apprentissage par renforcement profond caractérisés par l'utilisation de réseaux de neurones pour optimiser

des paramètres physiques. On retrouve par exemple ce genre d'optimisation pour de la simulation de marche sur des modèles bipèdes [12] [11], qui est très certainement l'un des sujets d'étude parmi les plus représentés.

On s'intéresse plus spécifiquement à l'algorithme Proximal Policy Optimization (ou PPO) [4] qui jouera un rôle important dans ce projet en tant que technique d'apprentissage principale utilisée.

PPO est une méthode d'apprentissage profond basée sur le Deep-Q learning (DQL). Elle est devenu en peu de temps l'une des plus populaires grâce à son efficacité et sa simplicité d'implémentation. Tout comme celui de DQL, l'algorithme de PPO va chercher à déterminer une politique optimale en interagissant avec l'environnement.

La particularité principale de PPO est la "stabilité" de sa politique. Celle-ci va rester proche de la dernière politique choisie, permettant de réduire la variance pendant la phase d'apprentissage, au prix d'un biais plus élevé.

Enfin, concernant le moteur de simulation, Opensim[1] tend à devenir le standard du domaine de la recherche quant à la simulation de mouvements humains.

OpenSim[1] est un moteur de simulation musculo-articulaire, permettant de réaliser simulations physiques réalistes d'un point de vue biologique et physique à partir d'un modèle. Son efficacité et sa simplicité d'utilisation en font un très bon choix pour des études de mouvements humains, ou le développement de nouveaux outils dédiés pour la simulation prédictive (Comme le logiciel open-source SCONE [10])

Cependant bien qu'Opensim soit devenu une référence dans le domaine de la recherche, il existe d'autres moteurs plus discrets, dont la librairie open-source DART (Dynamic Animation and Robotics Toolkit)[3]

DART présente plusieurs avantages vis-à-vis d'OpenSim : Sa simplicité de manipulation de ses modèles ou encore que sa vitesse de simulation légèrement supérieure à celle d'Opensim en font une solution fiable pour la simulation musculo-articulaire.

De plus, la méthode de simulation dont s'inspire une grande partie de ce projet a été réalisée dans un environnement basé sur DART.

Pour ces raisons, DART sera le moteur de simulation qui sera utilisé au cours de ce projet.

3 Travaux réalisés

3.1 Définition du modèle

Il a d'abord fallu définir le modèle qui a été utilisé tout au long du projet. Ce modèle devait respecter plusieurs contraintes, tout en permettant une certaine flexibilité en cas de redéfinition des objectifs du projet tout en s'assurant un certain réalisme, qui est l'un des objectifs initiaux principaux. Les principales contraintes étaient les suivantes :

- Le modèle doit être simple. Cela signifie que, contrairement à un modèle complet du corps humain, on veut ici un modèle qui comporte peu d'éléments (Donc peu de corps solides et muscles), pour permettre d'une part de simplifier au maximum les problèmes pouvant survenir, et d'autre part de réduire drastiquement les temps de simulation et d'apprentissage pour pouvoir accélérer les différents tests à effectuer au cours du projet
- Le modèle doit s'affranchir des collisions, et donc de contacts potentiels dans son environnement virtuel. En effet, ajouter un modèle physique de collisions rajoute un certain niveau de complexité, et n'a que peu d'intérêt par rapport aux objectifs du projet. S'en affranchir permet de se concentrer sur la démonstration de l'efficacité des différentes techniques d'apprentissage, en supprimant toutes les sources de problèmes liés aux collisions (Notamment les forces de réaction pouvant s'appliquer sur les différents corps du modèles)
- Conformément à la précédente contrainte, le modèle choisi ne pouvait pas être un modèle classique de jambe, qui dépend énormément des collisions lors de son mouvement au niveau de l'articulation du pied. Il fallait donc s'assurer que le modèle, bien que différent, ait une structure osseuse et musculaire suffisamment proche de celle d'une jambe, pour assurer un comportement cinématique similaire lors des différents tests.

Le type de modèle qui respecte à priori au mieux ces trois contraintes est un modèle de bras : Suffisamment simple, aucune collision avec le sol, et une structure similaire à celle d'une jambe.

Conception

Il a alors fallu concevoir le modèle. À partir d'un modèle de corps entier, en isolant les éléments connectés et/ou appartenant au bras, il a été possible d'obtenir un modèle satisfaisant les contraintes citées. Bien que certaines modifications et abstractions ont du être faites pour rendre cette conversion possible, cela a permis d'obtenir assez rapidement un modèle exploitable dont les dimensions sont réalistes et en adéquation avec celles d'un homme adulte moyen. Après plusieurs essais, le modèle retenu est le suivant :

- De 4 solides représentant respectivement la main, l'avant-bras (regroupant l'ensemble ulna / radius), le bras, et l'épaule (regroupant l'ensemble clavicule / acromion / omoplate) . La main n'est pas décomposée en plusieurs solides pour conserver la simplicité du modèle. Le modèle possède donc 3 articulations entre chacun des solides, et une articulation supplémentaire entre l'épaule et un solide indépendant fixe, représentant un pseudo-torse. Cela permet d'ancrer notre modèle dans l'environnement et ainsi de s'affranchir des collisions.

En plus de leurs coordonnées spatiales, chaque articulation possède des propriétés utiles lors de la simulation, telles que les limites angulaires des mouvements réalisables par cette articulation.

- De 53 muscles, permettant d'assurer le mouvement de tous les éléments du modèle. Ces muscles ne correspondent bien entendu pas tous à des muscles entiers, mais plutôt à des parties de muscles qui permettent, lorsqu'elles sont toutes actionnées par le système, de simuler au mieux le comportement du muscle réel.

Chaque muscle est défini par les solides qu'il relie, ainsi que plusieurs propriétés telles que la force maximale pouvant être générée, ou la longueur minimale et maximale de ce muscle.

De plus, bien que le modèle possède un aspect musculaire, ce dernier a pu être désactivé au cours du projet, lors de certains tests qui le nécessitaient. Le modèle adopte alors à une approche basée "torques", en s'affranchissant entièrement des activations musculaires et en se basant uniquement sur les forces de rotations générées au niveau des articulations (la génération de ces forces se fait de façon différente selon la méthode en question).

En tout, le modèle possède 10 degrés de libertés, répartis entre les 4 articulations citées précédemment (Voir tableau 1).

Main - Avant-bras	3 Rotations
Avant-bras - Bras	1 Rotation
Bras - Épaule	3 Rotations
Épaule - Torse (fixe)	3 Rotations

Table 1: Degrés de liberté par articulation

Adaptation des pathologies

Pour étudier les 3 pathologies du projet OMEGA citées précédemment, nous allons altérer le modèle de façon à analyser des cas similaires. De cette manière,

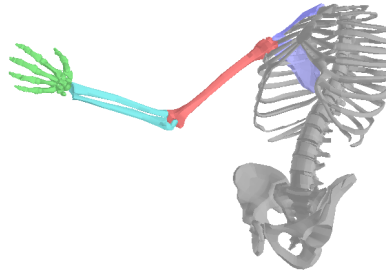


Fig. 1: Représentation du modèle. Chacun des 4 solides principaux est coloré, la partie non colorée représente la partie fixe du modèle. Les muscles sont masqués pour une meilleure lisibilité

on sera en mesure de justifier l'utilisation des différentes méthodes de prédiction du mouvement pour chacune de ces pathologies.

- Pour simuler la défaillance articulaire au niveau du genou, on limitera une articulation similaire dans notre modèle : l'articulation du coude (Car elle possède des propriétés cinématiques similaires). "Limiter" l'articulation signifie ici réduire l'amplitude du mouvement en imposant des limites angulaires plus petites que celles initialement prévues par le modèle.
- Pour simuler la pathologie "Drop-foot", on restreindra la force maximale pouvant être générée par un ensemble de muscles permettant d'agir sur un des degrés de liberté d'une articulation.
- Pour simuler une amputation, on supprimera un ou plusieurs solides initialement présents dans le modèle.

Définition du mouvement

Il a fallu ensuite définir un mouvement de référence, qui s'adaptait aux 3 contraintes initiales du modèle. Le but ici était, une fois de plus, de s'approcher au mieux d'un mouvement de marche, en utilisant toutes les capacités du modèle défini auparavant.

Après plusieurs esquisses de mouvements différents, le plus simple semblait être la meilleure solution, et c'est celui-ci qui a été retenu lors de la phase de développement et lors des différents tests effectués.

Le mouvement consiste en une levée de bras simple, accompagnée d'une rotation au niveau de chacune des articulations, permettant de générer un moment

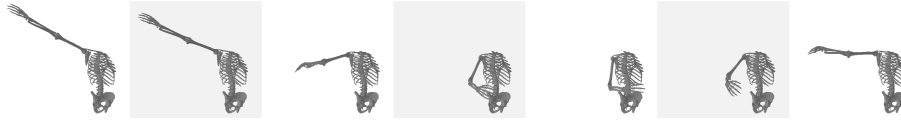


Fig. 2: Plusieurs frames successives du mouvement défini

cinétique sur chacun des degrés de libertés du modèle. De cette façon, on peut s'assurer de la qualité de l'apprentissage au niveau de chaque articulation.

Il a ensuite été généré et stocké (sous la forme d'un fichier .bvh, un format de fichier d'animation de personnages) à partir du modèle 3D défini précédemment. Cette génération a été faite en interpolant les positions des différents solides du modèle, entre une dizaine de positions-clé définies "à la main". De cette façon, on s'assure qu'aucun mouvement brutal ne sera fait au cours du mouvement.

Le mouvement est composé de 120 frames, et est cyclique. Une représentation de différentes étapes du mouvement peut être observée sur la figure 2

Ce mouvement, ou plutôt le fichier le contenant, n'a pas été systématiquement utilisé lors de l'ensemble des phases d'apprentissage de ce projet (seulement certaines, qui seront détaillées plus tard). Cependant, il a servi et servira tout au long de ce rapport de mouvement de référence pour les différents tests effectués et leurs résultats.

Il fera office de mouvement "optimal" du point de vue de l'objectif de réalisme du projet, car sa simplicité nous confirme qu'il est bel et bien réalisable par un humain réel.

3.2 Simulation prédictive musculaire

La majeure partie des travaux réalisés au cours de ce projet s'inspirent d'une publication de 2019 s'intitulant *Scalable Muscle-Actuated Human Simulation and Control* [7], ou MASS (Qui est l'acronyme de Muscle-Actuated Skeletal System, et qui sera utilisé dans la suite de ce rapport pour désigner cet article)

MASS a deux objectifs :

- Implémenter un environnement simple de simulation, permettant de contrôler un modèle de corps humain complet. Cet environnement de simulation est basé sur DART qui a été évoqué précédemment.
- Utiliser cet environnement pour tenter de reproduire un mouvement humain de référence fourni en entrée du système. Ce nouveau mouvement

est produit à l'aide d'un algorithme de Deep Reinforcement Learning. Plus spécifiquement, le système est basé sur l'algorithme PPO (Proximal Policy Optimization), qui présente plusieurs avantages dont son efficacité avérée ainsi que sa simplicité d'implémentation.

L'objectif de MASS est à priori assez différent de celui de ce projet. Au lieu de reproduire un mouvement prédéfini à partir d'un fichier de référence, on désire simuler un tout nouveau mouvement, satisfaisant des contraintes moins strictes. Cependant, les deux projets montrent de grandes similarités sur l'environnement de simulation et l'aspect prédictif du projet.

En modifiant la façon dont le système apprend, il devient alors possible de générer de façon prédictive un mouvement - sans mouvement de référence - comme cela sera montré dans la partie dédiée aux résultats.

L'architecture utilisée par MASS (figure 3) est très similaire à celle qui a été utilisée au sein de ce projet

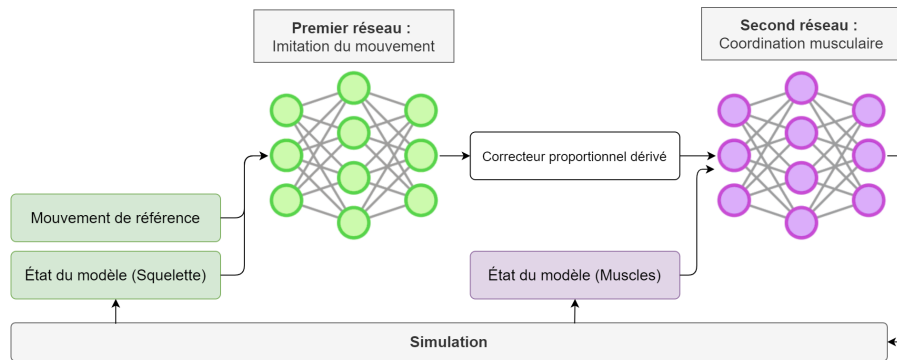


Fig. 3: Architecture de MASS

MASS utilise deux réseaux de neurones linéaires "fully-connected", ayant chacun une tâche précise :

- Le premier réseau reçoit en entrée l'état du modèle (c'est à dire la position spatiale et angulaire de tous les solides du modèle). Son rôle est de générer un mouvement similaire au mouvement de référence, et de retourner une action à effectuer
- Le second réseau va recevoir en entrée cette action (Après que l'erreur ait été réduite à l'aide d'un contrôleur proportionnel dérivé). Son rôle est de générer,

à partir de l'action choisie par le premier réseau, les activation musculaires correspondantes pour simuler ce mouvement. Il retourne un vecteur dont la taille correspond au nombre de muscles du modèle, qui contient des valeurs d'activation (Comprises entre 0 et 1).

La fonction de récompense utilisée dans MASS se concentre exclusivement sur la reproduction du mouvement de référence, en récompensant le système si la position simulée de chaque solide est proche de la position de référence souhaitée. À la place, on va chercher à introduire de nouvelles récompenses pour générer un nouveau mouvement suivant des règles simples et naturelles. On introduit donc deux nouvelles composantes à la fonction de récompense : Une composante dédiée au but du mouvement, et une composante dédiée au coût énergétique du mouvement.

Récompense de but (r_g)

La récompense de but (ou "Goal reward", notée r_g), permet de simuler un but à atteindre. En effet, dans un environnement réel, chaque mouvement à un but bien précis (déplacer ou actionner un objet, communiquer, etc.). Cette récompense va alors permettre au système de générer un mouvement capable d'atteindre le but imposé, de la façon la plus satisfaisante et optimale possible.

Dans le cas de l'étude dans mouvement de marche, un but évident serait bien évidemment lié à l'efficacité de la marche, avec des métriques telles que la distance parcourue avant la chute du modèle, ou encore la vitesse moyenne du modèle dans un temps donné.

Cependant, dans le cas d'un mouvement de bras, il est plus complexe de trouver un objectif naturel, sans ajouter un solide externe au système (pour la prise en main d'un objet par exemple). De plus, cela entrerait en conflit avec l'une des contraintes initiales du projet qui était de s'affranchir entièrement des collisions.

On choisit alors de définir un but simple, qui consiste à atteindre, avec une extrémité du modèle (en l'occurrence le solide correspondant à la main), des points définis spécifiques, à des moments précis du mouvement. Ces positions sont définies à partir du modèle de référence, et leur nombre restera faible pour éviter que cette récompense agisse comme la récompense initiale de MASS visant à reproduire un mouvement de référence.

À titre d'exemple, les tests effectués ont été réalisés avec 2 positions cible, pour un mouvement cyclique de 120 frames (et donc 120 positions de référence au total pour chaque solide du modèle). Ces positions ne sont donc en aucun cas représentatives du mouvement complet.

Enfin, la récompense obtenue est proportionnelle à la distance entre la position simulée et la position cible, lors des frames concernées.

Par souci de normalisation, cette récompense sera pondérée par le nombre de frames total, pour que la récompense maximale générée sur ces deux frames corresponde à la récompense totale cumulée générée lors des 120 frames du mouvement par les autres composantes de la fonction de récompense.

Récompense énergétique (r_j)

La récompense énergétique (notée r_j) permet d'introduire une notion de coût énergétique dans l'apprentissage du système. L'intérêt est assez évident : Simuler une contrainte réelle cruciale lorsqu'il s'agit d'étudier ou de simuler un mouvement. La recherche a prouvé plus d'une fois que le coût énergétique est l'un des facteurs prédominant dans la réalisation d'une action par un individu, ce qui influe naturellement sur le mouvement qui sera produit.

La fonction de coût énergétique s'inspire de celle provenant d'une publication de 2019 s'intitulant *Optimizing Locomotion Controllers Using Biologically-Based Actuators and Objectives* [6]

On utilisera une version simplifiée (pour s'adapter aux variables de notre système) de la fonction de dépense énergétique métabolique qui y est utilisée. Cette fonction permet de mesurer l'énergie dépensée par un muscle lors de son activation, et utilise plusieurs variables propres aux muscles simulés : La masse, la taille au repos, la vitesse de contraction, et la force actuelle générée.

L'objectif principal de cette récompense est de limiter les variations brutales d'activation musculaires qui peuvent survenir lors de la simulation. Si un muscle se contracte trop soudainement par exemple (la valeur d'activation du muscle passe soudain de sa valeur minimale à sa valeur maximale), la récompense engendrée sera négative pour le système.

La fonction de récompense finale R est donc de la forme :

$$R = w_p r_p + w_g r_g - w_j r_j \quad (1)$$

Où :

- w_p et r_p sont respectivement le poids et la récompense liés à la similarité entre le mouvement généré et le mouvement de référence
- w_g et r_g sont respectivement le poids et la récompense liés à la fonction de but
- w_j et r_j sont respectivement le poids et la récompense liés à la fonction de coût énergétique

La récompense liée à la reproduction du mouvement de référence (r_p) sera conservée pour réaliser les tests sur son utilité et son efficacité vis-à-vis des autres composantes. Son poids sera, dans la majorité des cas, gardé très bas, voire même nul.

Chaque terme est normalisé, de façon à ce que les différentes récompenses partagent des valeurs homogènes. Tous les termes de la fonction de récompense sont donc compris dans l'intervalle $[0, 1]$.

Quant aux poids, ils seront toujours définis tels que la somme de tous les poids soit égale à 1.

On peut voir sur la figure 4 l'architecture de MASS modifiée pour s'adapter à notre projet.

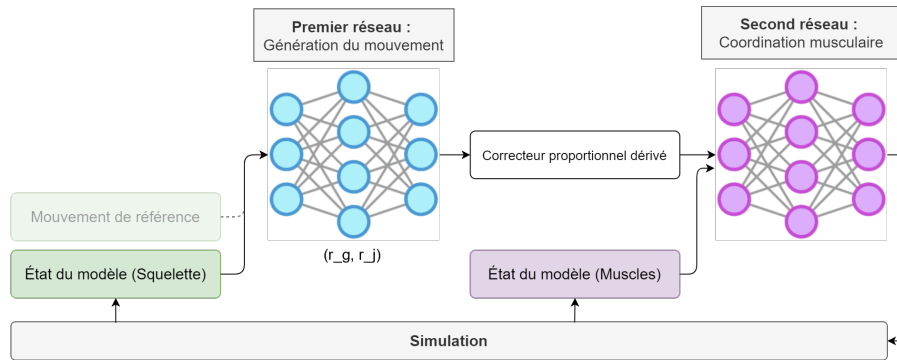


Fig. 4: Architecture modifiée. Le mouvement de référence n'est plus nécessaire, et le premier réseau possède une tâche différente

3.3 Résultats

On cherche dans cette partie à mesurer ici l'efficacité de l'apprentissage de notre système. Conformément aux objectifs du projet, on désire simuler un mouvement réaliste à partir d'un système basé sur de l'apprentissage par renforcement.

La métrique principale qui sera utilisée, nommée p_{diff} , est une valeur comprise entre 0 et 1, calculée à partir de 2 termes :

- La différence de position entre deux frames successives. Plus cette différence est grande, moins le mouvement sera "fluide" et considéré comme donc naturel.

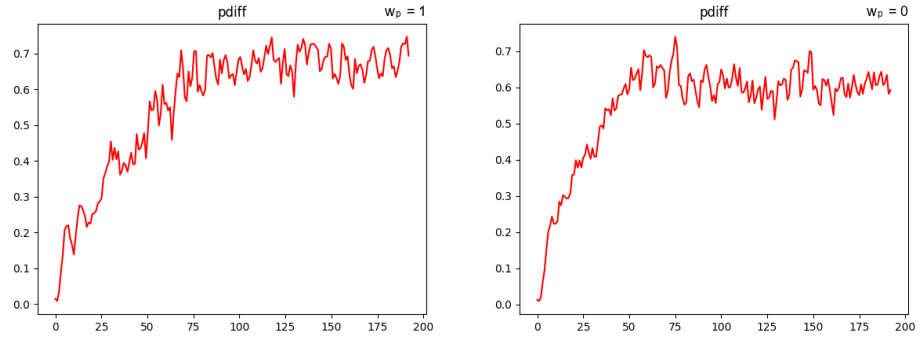


Fig. 5: Évolution de p_{diff} au cours de la phase d'apprentissage : À gauche, pour $w_p = 1$, à droite pour $w_p = 0$

- La différence de position avec le mouvement de référence. Comme dit plus tôt, on considère ce mouvement comme "optimal" du point de vue de l'objectif de réalisme du projet.
L'objectif visé par ce second terme est donc le même que le premier : s'assurer que le mouvement reste le plus naturel possible.

Au plus p_{diff} se rapprochera de 1, plus le mouvement généré sera considéré comme idéal.

On commence par mesurer la capacité du système à générer un mouvement optimal. Pour cela, on va comparer deux simulations, avec des poids différents au niveau de la fonction de récompense.

Les phases d'apprentissage se feront sur une durée de 200 épisodes, ce qui est dans le cas de notre modèle simple suffisant pour converger vers la solution optimale.

Dans un premier temps, on génère un mouvement où $w_p = 1$. C'est à dire que le seul but de l'apprentissage sera de générer un mouvement similaire à celui de référence.

Puis dans un second temps, on réalise un apprentissage où $w_p = 0$, et où le poids de la récompense est répartie sur les 2 autres composantes ($w_g = 0.6$ et $w_j = 0.4$). Le but ici est de montrer la capacité du système à générer un mouvement réaliste et efficace sans l'aide d'un mouvement prédéfini.

On peut voir sur la figure 5 que malgré le fait que le second mouvement montre un p_{diff} légèrement inférieur, il reste très proche du mouvement appris à partir du mouvement de référence.

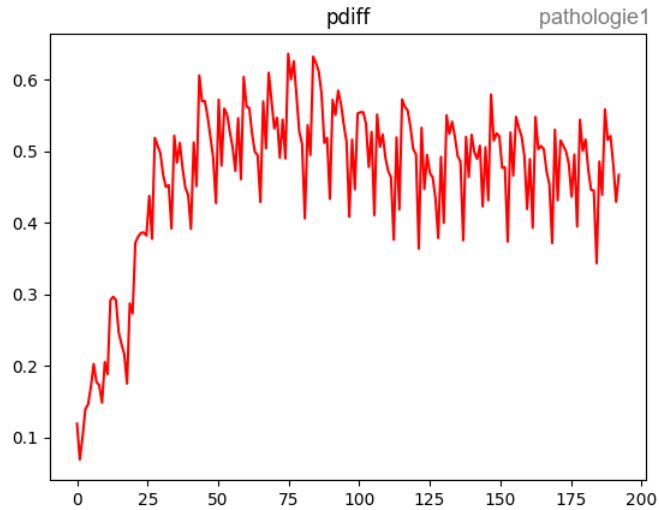


Fig. 6: Évolution de p_{diff} au cours de la phase d'apprentissage : Cas de la défaillance articulaire

Si on considère uniquement les valeurs de p_{diff} après le 100e épisode (lorsque l'apprentissage converge vers sa solution), on obtient une valeur moyenne de 0.68 pour $w_p = 1$, et de 0.60 pour $w_p = 0$.

Cela signifie que sans référence, et seulement avec un but primitif, le système est capable de générer un mouvement d'une qualité très proche d'un mouvement que l'on a initialement défini comme optimal.

On s'intéresse maintenant à la capacité du système à générer ce même mouvement, dans le cas des 3 pathologies étudiées au cours de ce projet. w_p sera gardé à 0 pendant les prochains tests.

Pour la pathologie 1 (Défaillance articulaire), on limite le mouvement de l'articulation du coude à 20% de sa course normale. On observe alors l'effet de cette modification lors de l'apprentissage du mouvement

On observe sur la figure 6 des résultats moins convaincants que ceux observés précédemment. D'une part car le p_{diff} moyen après convergence est moins élevé (légèrement inférieur à 0.5), et d'autre part car sa valeur est instable. Le système n'arrive pas à générer un mouvement aussi fluide que dans le cas d'un modèle avec une articulation fonctionnelle.

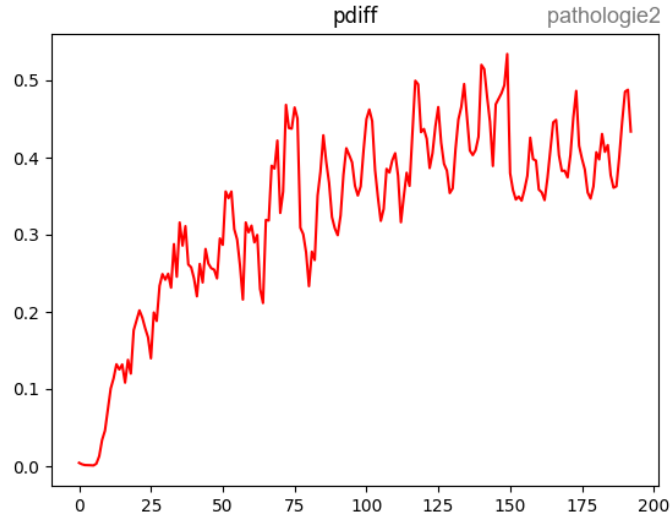


Fig. 7: Évolution de p_{diff} au cours de la phase d'apprentissage : Cas de la défaillance musculaire

Cela s'explique par le fait que le système est incapable de satisfaire les conditions pour obtenir les récompenses optimales à cause de la défaillance induite dans le modèle. La simulation oscille sans cesse entre des solutions proches de la meilleure, mais n'arrive pas à sélectionner les solutions naturelles comme étant les meilleures.

Malgré tout, p_{diff} reste suffisamment proche de sa valeur optimale pour que le résultat reste satisfaisant.

Dans le cas de la pathologie 2, on réduit de moitié la force maximale pouvant être générée par l'ensemble des muscles permettant d'actionner l'articulation Épaule-Bras. Cette modification concerne 22 muscles et permet d'induire une défaillance musculaire importante au niveau de la partie supérieure du modèle.

On observe ici aussi (figure 7) des résultats plus bas que ceux observés précédemment. Le p_{diff} moyen est nettement plus faible que la valeur optimale (moyenne de 0.4), mais est cette fois-ci plus stable. Le système n'arrive tout simplement pas à générer un mouvement satisfaisant vis-à-vis de sa fonction de récompense, et il essaye tant bien que mal d'arriver à un état satisfaisant mais sans grand succès. Cela a pour effet d'augmenter l'instabilité des activations musculaires dans le

temps, augmentant le nombre de mouvement erratiques générés.

Augmenter le poids de la composante énergétique de la fonction de récompense pourrait améliorer p_{diff} , mais au détriment de la composante liée au but du mouvement (r_g) qui doit rester prédominante compte tenu des objectifs du projet.

Enfin, le cas de la 3e pathologie est assez spécial. En effet, il est dans notre cas compliqué de retirer plusieurs solides du modèle sans "sur-simplifier" le problème.

Supprimer deux solides sur les quatre présents dans le modèle limite bien trop la génération du mouvement, à tel point que la grande majorité des mouvements produits seraient identiques, même avec une phase d'apprentissage de quelques minutes. De plus, la définition de la récompense de but pose également un problème, puisqu'il serait bien trop simple à satisfaire dans un cas avec deux solides.

Nous sommes donc dans l'impossibilité de tester efficacement cette pathologie dans le cadre simplifié de ce projet.

Les résultats obtenus avec notre système de simulation sont satisfaisants et prometteurs. Cependant, il subsiste un problème inhérent à la simulation : Le temps d'apprentissage et les ressources nécessaires à celui-ci.

En effet, sur un modèle simple comme celui qui a été utilisé dans le cadre de ce projet, les temps de calcul sont très rapides. Mais dans le cas d'un apprentissage sur un modèle complet de corps humain (comme c'est le cas lors de l'étude d'un mouvement de marche entier), les temps de calculs deviennent rapidement gigantesques.

Cela est principalement à cause du temps nécessaire pour simuler un grand nombre de muscles, qui sont bien plus nombreux que le nombre de solides présents dans le modèle. On cherche alors un moyen d'alléger ce temps de calcul.

3.4 Simulation prédictive pseudo-musculaire

Cette partie s'inspire d'une publication de 2019 s'intitulant *Synthesis of Biologically Realistic Human Motion Using Joint Torque Actuation* [5]. Le but de ce projet est très intéressant puisqu'il est le même que celui que l'on recherche : Générer un mouvement réaliste en s'affranchissant d'un modèle musculaire.

Le principe derrière cette génération est simple : À l'aide de deux réseaux de neurones, il devient possible de transformer une simulation basée muscle en un simulation basée torque physiquement équivalente.

Chaque réseau pour tâche de générer des données additionnelles depuis le modèle musculaire :

- Le premier va, à partir de l'état du modèle, générer des données représentant la "faisabilité" du mouvement. Plus la position du squelette est éloignée d'une position réalisable par le modèle musculaire initial, plus cette valeur sera basse.
- Le second va, à partir de l'état du modèle, générer des données représentant le coût énergétique qu'aurait le modèle s'il était un modèle musculaire équivalent

À partir de ces nouvelles données, et du modèle simulé (cette fois-ci sans muscles), il devient alors possible de générer un mouvement à l'aide d'un solveur de contraintes classique (En l'occurrence, Matlab).

Architecture hybride

Cette façon de procéder, en générant des données à partir d'une simulation musculaire pour s'en servir par la suite, présente un avantage majeur : La simulation basée muscle ne sera lancée qu'une fois, lors de la phase d'apprentissage des 2 nouveaux réseaux de neurones. Bien que cette phase d'apprentissage soit bien plus longue que celle de la précédente architecture, cela signifie que, tant que le modèle utilisé reste identique, toutes les simulations peuvent se faire sans muscles.

On obtiendrait alors un gain de temps majeur, qui permettrait d'accélérer drastiquement la phase d'apprentissage du réseau principal,

La figure 8 représente l'architecture utilisant cette nouvelle méthode. On cherche ici à reproduire la structure et les performances de l'architecture précédente (en réutilisant le même environnement de simulation), tout en s'affranchissant de toute simulation musculaire durant la phase d'apprentissage.

L'idée est d'utiliser ces deux réseaux de neurones P_NN et J_NN, ou plutôt les données qu'ils génèrent, en tant que récompense de notre environnement de simulation précédent. On peut ainsi remplacer l'ancienne récompense énergétique par cette nouvelle récompense, et introduire une nouvelle récompense de faisabilité du mouvement.

La nouvelle fonction de récompense finale R est donc de la forme :

$$R = w_g r_g + w_f r_f - w_j r_j \quad (2)$$

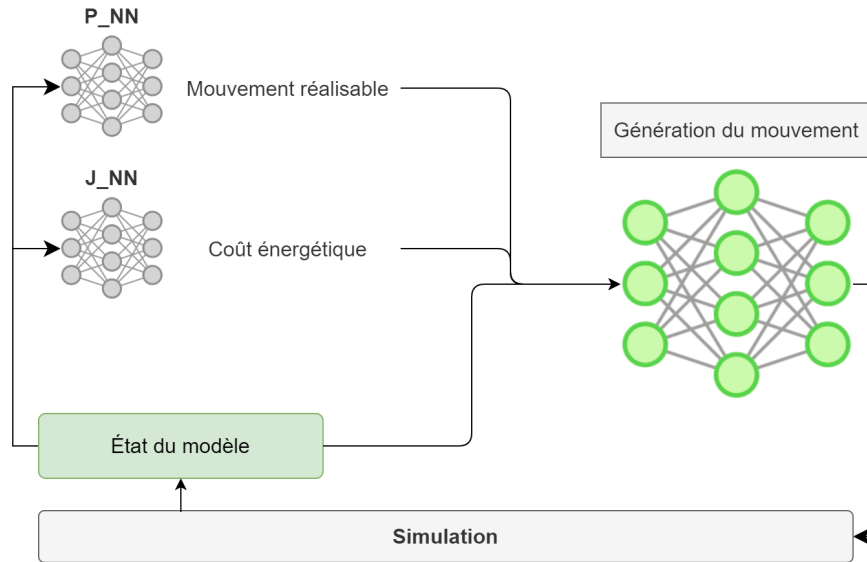


Fig. 8: Architecture du nouvel environnement de simulation

Où :

- w_g et r_g sont respectivement le poids et la récompense liés à la fonction de but.
- w_f et r_f sont respectivement le poids et la récompense liés à la faisabilité du mouvement appliqué à un modèle sans muscles
- w_j et r_j sont respectivement le poids et la récompense liés à la fonction de coût énergétique du modèle musculaire équivalent

Malheureusement, par manque de temps, il n'a pas été possible d'implémenter cette architecture entièrement, et d'obtenir des résultats satisfaisants de simulation.

Il n'y aura donc pas de partie consacrée aux résultats de cette nouvelle architecture.

Cependant, elle reste une piste très intéressante à explorer pour la suite du projet OMEGA, dans l'optique de remplacer complètement les simulations basées muscle par des simulations équivalentes basées torque, bien plus rapides.

4 Conclusion

Au cours de ce projet, on désirait étudier l'efficacité de techniques d'intelligence artificielle, et notamment d'apprentissage par renforcement profond, au sein d'un problème de simulation prédictive.

Le système mis en place permet de réaliser un mouvement cohérent et réaliste, à partir d'un modèle simple.

PPO, ou les techniques d'apprentissage profond de façon générale, semblent être une piste à privilégier lorsqu'on désire effectuer de la simulation prédictive. Leur simplicité de résolution (s'opposant à la complexité d'un solveur de contraintes physiques classique) permet de générer des mouvements très naturels.

De plus, en comparant ce mouvement avec des données réelles et en adaptant les récompenses lors de l'apprentissage, on peut facilement envisager une simulation très proche de la réalité, en utilisant des objectifs extrinsèques.

De cette façon, on pourrait simuler un mouvement à partir d'un modèle pathologique fictif, permettant d'évaluer l'évolution d'une maladie de façon réaliste, et permettant de mieux appréhender un patient réel.

Le principal défaut de ces méthodes d'apprentissage est le temps nécessaire pour entraîner les réseaux. En effet, sur un modèle musculo-articulaire, la phase d'apprentissage peut facilement durer plusieurs jours sur un modèle humain complet avant de converger vers une solution acceptable.

Cependant, grâce à notre prototype d'environnement de simulation pseudo-musculaire, on pourrait réduire drastiquement le temps total de simulation, rendant le temps d'apprentissage sur un modèle complet acceptable.

Dans le cadre du projet OMEGA, qui repose en grande partie sur l'étude de modèles complets, un tel gain de temps serait non-négligeable, et permettrait de combiner un faible temps de simulation d'une approche basée torques, avec le réalisme d'une approche basée muscles.

References

1. A. Seth, M. Sherman, J.R.S.D.: OpenSim: A musculoskeletal modeling and simulation framework for in silico investigations and exchange. *Procedia Iutam* 2 (2011)
2. E.s Sifakis, I. Neverov, R.F.: Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph* 24, 3 (2005)
3. J. Lee, M.X. Grey, S.H.T.K.S.J.Y.Y.S.S.M., Liu., C.: DART: Dynamic Animation and Robotics Toolkit. *The Journal of Open Source Software* 3, 22 (2018)
4. J. Schulman, F. Wolski, P.D.A.R.O.K.: Proximal policy optimization algorithms. *CoRR* (2017)
5. Jack M., Wang Samuel R., H.S.L.D.V.K.: Synthesis of Biologically Realistic Human Motion Using Joint Torque Actuation. *ACM Trans. Graph* (2012)
6. Jiang, Van Wouwe, D.G.L.: Optimizing Locomotion Controllers Using Biologically-Based Actuators and Objectives. *ACM Trans. Graph* 38 (2019)
7. S. Lee, M. Park, K.L.J.L.: Scalable Muscle-Actuated Human Simulation and Control. *ACM Trans. Graph* 38 (2019)
8. S. Sueda, A. Kaufman, D.P.: Musculotendon Simulation for Hand Animation. *ACM Trans. Graph* 27, 3 (2008)
9. S.H. Lee, D.T.: Heads Up!: Biomechanical Modeling and Neuromuscular Control of the Neck. *ACM Trans. Graph* 25, 3 (2006)
10. T, G.: SCONE: Open Source Software for Predictive Simulation of Biological Motion. *Journal of Open Source Software*, 4(38), 1421 (2019)
11. W. Yu, G. Turk, C.L.: Learning Symmetric and Low-energy Locomotion. *ACM Trans. Graph* 37, 4 (2018)
12. X.B. Peng, G. Berseth, K.Y.M.V.D.P.: DeepLoco: Dynamic Locomotion Skills Using Hierarchical Deep Reinforcement Learning. *ACM Trans. Graph* 36, 4 (2017)