Synthétisation de mouvements de lever du sol pour les personnages basés sur la physique

par

Anthony FREZZATO

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE DES TECHNOLOGIES DE L'INFORMATION
M. Sc. A.

MONTRÉAL, LE 19 AOÛT 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Sheldon Andrews, Directeur de mémoire
Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

Mme. Samira Ebrahimi Kahou, Présidente du jury
Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

M. Adrien Gruson, Membre du jury
Département de génie logiciel et des technologies de l'information à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 27 JUILLET 2022

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je tiens à remercier mon directeur de maîtrise, Sheldon Andrews, qui m'a accompagné dans les différentes étapes de la maîtrise. Il fut un excellent guide pour l'accomplissement de ce projet et ses conseils m'ont aidé à m'améliorer sur le plan professionnel. Je tiens aussi à le remercier pour toute l'aide apportée pour la publication d'un article scientifique portant sur le sujet de la maîtrise.

J'aimerais aussi remercier Arsh Tangri qui a participé au projet, qui fut d'une grande aide technique et critique qui nous a permis de mener à bien ce projet.

Je remercie également tous mes collègues du laboratoire de recherche multimédia de l'ÉTS pour leurs commentaires qui m'ont aidé à améliorer ce projet de recherche.

Finalement, je remercie ma compagne qui m'a soutenu pendant toutes ses années.

Synthétisation de mouvements de lever du sol pour les personnages basés sur la physique

Anthony FREZZATO

RÉSUMÉ

Nous proposons une méthode pour synthétiser des mouvements de lever du sol pour un personnage simulé par la physique. À partir d'une position au sol, notre objectif n'est pas d'imiter une séquence de mouvement en particulier, mais de produire des mouvements qui correspondent aux courbes d'entrées décrivant le style du mouvement de lever du sol. Notre cadriciel utilise l'apprentissage par renforcement profond pour entraîner une politique afin de contrôler un personnage simulé par la physique. Un espace latent de poses naturelles est fabriqué à partir d'une base de données de capture de mouvements. Les poses sont conditionnées par des caractéristiques d'entrée qui décrivent le mouvement. Nous démontrons que notre approche peut synthétiser des mouvements qui suivent le style des courbes créées par l'utilisateur, ainsi que des courbes extraites des mouvements de référence. Dans ce dernier cas, les mouvements du personnage basé sur la physique ressemblent aux mouvements de référence originaux. De nouveaux mouvements peuvent être synthétisés facilement en ne modifiant qu'un petit nombre de paramètres qui décrivent le mouvement. Nous démontrons également le succès de nos contrôleurs sur des terrains accidentés et inclinés.

Mots-clés: Synthétisation de mouvements de lever du sol, personnages basés sur la physique, apprentissage par renforcement profond.

Synthesizing Get-Up Motions for Physics-Based Characters

Anthony FREZZATO

ABSTRACT

We propose a method for synthesizing get-up motions for physics-based humanoid characters. Beginning from a supine or prone state, our objective is not to imitate individual motion clips, but to produce motions that match input curves describing the style of get-up motion. Our framework uses deep reinforcement learning to learn control policies for the physics-based character. A latent embedding of natural human poses is computed from a motion capture database, and the embedding is furthermore conditioned on the input features. We demonstrate that our approach can synthesize motions that follow the style of user authored curves, as well as curves extracted from reference motions. In the latter case, motions of the physics-based character resemble the original motion clips. New motions can be synthesized easily by changing only a small number of controllable parameters. We also demonstrate the success of our controllers on rough and inclined terrain.

Keywords: Synthesizing get-up motions, Physics-based characters, Deep reinforcement learning.

TABLE DES MATIÈRES

Page

INTRODUCTION 1

CHAPITRE 1 ÉTAT DE L'ART 5
1.1 Optimisation de trajectoire 5
1.2 Biomécanique 8
1.3 Robotique 9
1.4 Apprentissage par renforcement profond 9
1.4.1 Espace des actions 9
1.4.2 Approche basé sur l'imitation contradictoire 10
1.4.3 Apprentissage sans mouvements de référence 12
1.4.4 Apprentissage par imitation 14
1.4.5 Espace latents 16
1.4.6 Apprentissage supervisé 19
1.5 Synthèse 20

CHAPITRE 2 MÉTHODOLOGIE 21
2.1 Vue générale 21
2.2 Modélisation du personnage 23
2.3 Modélisation de l'environnement 27
2.4 Représentation de la politique de contrôle 28
2.4.1 Vecteur d'état 28
2.4.2 Vecteur d'action 30
2.5 Mouvement naturels 31
2.6 Contrôle du style 33
2.6.1 Représentation en courbe définit par morceau (spline) 33
2.6.2 Caractéristiques des mouvements 34
2.6.3 Édition des courbes 35
2.7 Entraînement de la politique de contrôle 36
2.7.1 Méthode d'apprentissage 37
2.7.2 Fonction de récompense générale 39
2.7.3 Fonction de récompense de tâche de lever du sol 40
2.7.4 Initialisation 42
2.7.5 Fin hâtive d'un épisode 42
2.8 Apprentissage par curriculum 43
2.9 Entraînement des courbes 43
2.10 Détails d'implémentation 44

CHAPITRE 3 RÉSULTATS ET DISCUSSIONS 47
3.1 Mouvements de référence 47
3.2 Édition de courbes 51

3.3 Étude d'ablation 55
3.3.1 VAE Standard 56
3.3.2 Retrait du pose C-VAE 57
3.3.3 Retrait de l'apprentissage par curriculum 57
3.3.4 Discussion 58

CONCLUSION 61

BIBLIOGRAPHIE 65

LISTE DES TABLEAUX

Page

Tableau 2.1 Paramètres de couple des axes servo PD 25

Tableau 2.2 Limitation des axes 26

LISTE DES FIGURES

Page

Figure 2.1 Vue générale de l'architecture 21

Figure 2.2 Modélisation du personnage physique 3D 23

Figure 2.3 Le terrain est composé de trois zones : facile, moyen et difficile 27

Figure 2.4 Interface d'édition des courbes 35

Figure 2.5 Visualisation d'une édition de courbes 36

Figure 2.6 Entraînement de la politique de contrôle 37

Figure 3.1 Exemples de mouvements sur terrain plat et incliné 48

Figure 3.2 Exemples de mouvements sur terrain plat et incliné 49

Figure 3.3 Exemples de mouvements sur terrain plat avec animation de référence. 50

Figure 3.4 Exemples de mouvements sur terrain accidenté 51

Figure 3.5 Modification d'une trajectoire pour se lever plus tôt et plus tard. 52

Figure 3.6 Classification des stratégies de mouvement de lever du sol 53

Figure 3.7 Trajectoires éditées manuellement. 54

Figure 3.8 Courbe de progression pour l'étude d'ablation 56

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

DOF	Degré de liberté / *Degree of freedom*
PD	Proportionnel dérivée / *Proportional derivative*
VAE	Auto-encodeur variationnel / *Variational AutoEncoder*
C-VAE	Auto-encodeur variationnel conditionnel / *Conditional Variational AutoEncoder*
GAN	Réseaux antagonistes génératifs / *Generative adversarial networks*
GAIL	Réseaux antagonistes par imitation contradictoire / *Generative adversarial imitation learning*
PPO	Optimisation de politique proximal / *Proximal policy optimization*
RNN	Réseau de neurones récurrents / *Recurrent neural network*
ReLu	Fonction d'activation linéaire rectifiée / *ReLU activation function*
tanH	Fonction d'activation tangente / *tanH activation function*
PCA	Analyse des composants principaux / *Principal component analysis*
GAE	Estimation de l'avantage généralisé / *Generalized advantage estimate*

LISTE DES SYMBOLES ET UNITÉS DE MESURE

Nm	Unité de mesure du couple
$kg.m^2s^{-1}$	Unité de mesure de l'amortissement
$Nm.rad^{-1}$	Unité de mesure de la raideur
π	Politique de contrôle
$\mathbf{a}_t$	Vecteur d'action de la politique de contrôle
$\mathbf{s}_t$	Vecteur d'état de la politique de contrôle
$\bar{\mathbf{q}}$	Posture finale cible du personnage
$\mathbf{q}$	Pose naturelle
$\mathbf{\Delta_q}$	Décalages appliqués à la pose naturelle
$\mathbf{c}$	Vecteur auxiliaire pour conditionner l'espace latent
$\mathbf{c}'$	Vecteur auxiliaire pré-calculé pour entraîner l'espace latent
$\mathbf{z}$	Vecteur de paramètres de l'espace latent
$\mathbf{y}_t$	Vecteur d'action filtré
$\mathbf{y}_{t-1}$	Vecteur d'action filtré au pas de temps précèdent
β	Coefficient de lissage des actions
λ	Paramètre d'ajustement de l'estimation de l'avantage généralisé (GAE)
γ	Facteur de diminution
ϵ	Valeur de coupure

INTRODUCTION

L'animation basée sur la physique est un phénomène de plus en plus présent. Elle est très populaire dans les applications de robotiques et les applications interactives par ordinateur. L'avantage de cette approche est de rendre les animations plus naturel tout en réagissant plus fidèlement aux perturbations extérieures. Les animations basées sur la physique ont démontré d'excellents résultats dans les domaines de la simulation de contacts, de fluides et de corps mous. Cependant, dans notre travail nous utilisons la simulation physique pour produire des mouvements naturels uniquement pour les personnages humanoïdes.

Des recherches récentes ont démontré que l'utilisation de l'apprentissage par renforcement est très prometteur. Il permet d'étudier l'apprentissage d'un agent pour des compétences variées par essai-erreur dans un environnement inconnu. Par exemple, Peng, Abbeel, Levine & van de Panne (2018a) utilisent cette méthode pour apprendre un contrôleur dans le but de suivre une animation de référence avec un corps simulé par la physique tout en accomplissant une tâche supplémentaire, tels que frapper une cible. Ils s'intéressent plus particulièrement aux mouvements de locomotion. Ce type de mouvement est en général plus dynamique et implique peu de contacts. Plus récemment, Peng, Ma, Abbeel, Levine & Kanazawa (2021) se sont intéressés brièvement aux mouvements impliquant des contacts pour lever un personnage du sol après une chute. Cependant, leurs travaux visent à accomplir la tâche sans contrôle de style supplémentaire. Des méthodes plus anciennes telles que Mordatch, Todorov & Popović (2012) ou Liu, Yin, van de Panne, Shao & Xu (2010) se sont intéressés aux mouvements impliquant de nombreux contacts, mais leurs méthodes ne fonctionnent pas en temps réel.

Inspiré par les récents résultats démontrés par Peng *et al.* (2018a, 2021); Yin, Yang, Van De Panne & Yin (2021); Won, Gopinath & Hodgins (2021a); Bergamin, Clavet, Holden & Forbes (2019), nous nous sommes concentré sur les mouvements du corps impliquant des contacts pour se lever du sol. Ce type de mouvement est sous représentés par les récentes méthodes basées sur

l'apprentissage par renforcement profond. L'accent est mis sur l'apprentissage des contrôleurs pour synthétiser des mouvements de lever du sol impliquant de nombreux contacts tout en ayant un contrôle du style fournit à l'utilisateur. Ce type de mouvement est intéressant, car il demande une grande coordination dans leurs exécutions. De plus, les humains peuvent se lever du sol avec une grande variété de possibilités. Par exemple, certains mouvements peuvent être très lents, qui impliquent de créer des contacts solides et coordonnés afin de réussir. De plus, les configurations de départs au sol peuvent être très diversifiées (ex : sur le dos, le ventre, etc.), ce qui impliquent des stratégies très différentes pour accomplir les mouvements.

Motivation et objectives du mémoire

Les mouvements impliquant de nombreux contacts sont difficiles à créer. Les méthodes récentes de Peng *et al.* (2018a) Peng *et al.* (2021), Won *et al.* (2021a) et Bergamin *et al.* (2019) se basent sur des transitions de cadres d'animation de références qui sont contenus dans une base de données de mouvement. Les animations peuvent être créées par un artiste ou capturées avec un acteur. Cependant, ceci rend fastidieux leurs éditions ainsi que leurs adaptations sur des nouveaux environnements. D'autres méthodes récentes comme Peng, Guo, Halper, Levine & Fidler (2022) s'intéressent généralement à se lever pour retrouver une configuration debout. Un plus grand intérêt est alors porté sur la récupération d'une tâche ayant reçu une large perturbation qui fait tomber le personnage. Cependant, leurs exécutions omettent une grande variété de styles possibles. Les mouvements produits sont alors parfois irréalistes. Dans ce mémoire nous nous intéressons particulièrement à résoudre ce problème afin de fournir un contrôle de style supplémentaire. De plus, les travaux récents s'intéressent peu à l'adaptation sur des terrains non plats. Sachant que les méthodes existantes utilisent des cadres d'animation provenant d'un environnement de capture sur un terrain plat. Il est incertain si les méthodes actuellement proposées s'adaptent facilement sur des terrains escarpés et complexes pour des mouvements impliquant de nombreux contacts. Les auteurs s'intéressent plus particulièrement à

accomplir la tâche sans donner un contrôle sur le style de mouvement du personnage. Autrement dit, un personnage aura tendance à adopter toujours un même mode pour accomplir une tâche donnée. Dans la réalité, les humains adoptent des comportements très variés pour accomplir une même tâche. Par exemple, une personne âgée n'adoptera très certainement pas le même style de mouvement qu'un sujet jeune. De plus, des facteurs supplémentaires comme la forme du terrain ou la présence d'objets environnants influent fortement sur les stratégies employées pour accomplir cette tâche.

La synthétisation de mouvements naturels est une composante importante des applications basée sur la physique. En général deux approches dominantes permettent d'accomplir des tâches tout en respectant cette contrainte : la première est une approche basée sur l'imitation contradictoire utilisant des réseaux antagonistes (GAIL) et la deuxième par l'utilisation d'espaces latents.

Dans ce mémoire, nous nous concentrons sur l'utilisation d'un espace latent de poses naturel. Notre travail est inspiré par quelques éléments de la méthode proposée par Yin *et al.* (2021), et qui est adaptée afin d'accomplir des tâches de lever du sol. Par exemple, par rapport à la méthode initiale nous conditionnons l'espace latent à partir des caractéristiques qui décrivent le mouvement.

Contributions du mémoire

Les contributions principales du mémoire sont :

- **Mouvements naturels :** Un cadriciel basé sur l'apprentissage par renforcement profond qui est capable de produire des mouvements de lever du sol qui sont naturels et riches en contacts, tout en suivant la biomécanique et adoptant des comportements réalistes.
- **Éditable :** Une interface simple pour contrôler des caractéristiques fournies par un utilisateur afin de créer facilement de nouveaux mouvements.

- **Adaptation aux terrains :** Une méthode qui permet au personnage de s'adapter de manière robuste à des terrains non plats tout en suivant raisonnablement un style d'entrée.

L'Organisation du mémoire

Le reste du mémoire est organisé comme suit :

- **Chapitre 1 :** Nous présentons une revue de littérature qui décrit l'état de l'art pour l'animation de personnages physiques.
- **Chapitre 2 :** Nous présentons la méthodologie qui nous permet d'atteindre les objectifs du mémoire.
- **Chapitre 3 :** Nous présentons nos résultats produits par notre méthode, afin de synthétiser des mouvements de lever du sol.

CHAPITRE 1

ÉTAT DE L'ART

Dans cette revue de littérature nous passons en revue les différents travails effectués dans le domaine du contrôle de personnages animés par la physique et quelques travaux non basés sur la physique. Une étude a été effectué par Mourot, Hoyet, Le Clerc, Schnitzler & Hellier (2021) donne un aperçu des approches qui ont été développées ces dernières années pour contrôler des personnages basés sur la physique. Les premiers articles ont commencé à émerger en 2015 et 35 articles ont étaient publiés en 2020 qui démontre un certain engouement grandissant pour ce champ de recherches. Les modèles basés sur l'apprentissage par renforcement profond ont démontré d'excellents résultats pour synthétiser des mouvements complexes et dynamiques. Nous nous concentrons essentiellement sur les travaux qui ont mis l'emphase tout particulièrement sur les techniques d'apprentissage par renforcement profond dans le but d'animer des personnages simulés physiquement.

1.1 Optimisation de trajectoire

Dans notre projet nous n'utilisons pas la méthode basée sur l'optimisation de trajectoire, mais les méthodes plus récentes utilisant sur l'apprentissage profond se sont bâties en partie inspirées de cette méthode, car tous deux visent à minimiser une fonction de cout. Pour cette méthode, nous nous sommes intéressés particulièrement à la synthétisation de mouvement complexe dans un environnement physiquement simulé impliquant de nombreux contacts. Dans les méthodes impliquant l'optimisation de trajectoire, nous pouvons trouver une optimisation en temps réel et hors-ligne. Par exemple, elles sont basées sur des politiques optimisées par gradient ou par programmation dynamique.

Dans les travaux de Liu *et al.* (2010) , ils sont capables de synthétiser des mouvements riches en contacts tels que comme les roulades et les sauts à partir du sol. Les auteurs indiquent que les mouvements impliquant de nombreux contacts sont un gros défi et que les méthodes basées sur les gradients convergent souvent vers des minimums locaux pour des problèmes hautement

non linéaires. À noter, que ce travail est plus ancien et les avancements en apprentissage par renforcement profond n'étaient pas les mêmes qu'aujourd'hui. La méthode minimise le coût d'une fonction qui ressemble beaucoup aux fonctions de récompense utilisés dans l'apprentissage par renforcement. On dénote avec un terme pour la pose, un terme pour la position du joint racine, un terme pour la position des mains/pieds (end effectors) et un terme pour la balance du centre de masse par rapport au polygone de soutien. Cette méthode prend environ 25 secondes pour construire une seconde de mouvement sur un ordinateur de 80 cœurs sans avoir de problème de convergence. Cependant, leur méthode n'est pas robuste aux perturbations extérieures, qui est un désavantage sachant que les environnements physiques sont généralement perturbés.

Lin & Huang (2012) se sont occupés des problèmes de l'édition de mouvement de lever du sol. Leurs approches utilisent une exploration rapide d'un arbre (RTT) et un filtre basé sur un modèle physique permettent de générer des poses intermédiaires à partir d'une base de données de capture. Cette base de données contient des cadres d'animations entre la posture au sol et une posture clé. Notre approche nécessite uniquement un cas initial et un style désiré pour le mouvement qui est encodé dans des courbes faisant le suivi de la hauteur et de l'orientation du torse dans le temps. Ensuite, la totalité du mouvement est produite d'une manière plausible dans un environnement physique à l'aide d'un espace de poses naturelles.

Tassa, Erez & Todorov (2012) appliquent une optimisation en temps réel à un humanoïde dans un environnement simulé physiquement. Leur but est d'accomplir des tâches difficiles comme de se lever du sol à partir d'une pose aléatoire de façon robuste en répondant aux larges perturbations. Il est intéressant de voir que leurs fonctions de cout pénalisent la distance du centre de masse au sol, qui est une des solutions testées dans nos travaux. Ils utilisent une méthode d'optimisation basée sur un modèle prédictif qui évite une exploration intensive et permet de trouver des solutions seulement pour les états qui sont réellement visités. Ils atteignent des résultats qui s'approchent du temps réel sur une machine standard et répondant aux perturbations de façon acceptable.

Mordatch *et al.* (2012) s'intéressent à la planification de comportements complexes basés sur l'optimisation de contacts invariants. L'optimisation des contacts invariants est une manière de remodeler un espace de recherche de mouvements plus large, mais plus sage qui permet de trouver de bonnes solutions. Pour cela, les auteurs introduisent une variable scalaire qui indique si un contact potentiel doit être actif dans une phase donnée. Leur méthode est capable de synthétiser de nombreux comportements humains tels que se lever du sol, grimper, faire des mouvements acrobatiques et transporter des objets lourds. D'autre part leur méthode ne se limite pas aux êtres humains. Leur approche basée sur les gradients est entièrement automatique et ne demande aucune connaissance spécifique pour chacun des comportements. Être en possession d'exemples de mouvements naturels semble un prérequis pour les techniques récentes. Il est intéressant de constater qu'ils peuvent synthétiser des mouvements très complexes sans l'aide d'une base de données. L'optimisation s'effectue hors-ligne en plusieurs phases par l'activation/désactivation de termes dans la fonction de cout. Chaque animation prend de deux à 10 minutes pour être synthétisée qui reste très rapide par rapport au temps d'apprentissage des récentes méthodes utilisant des réseaux de neurones.

Une approche proposée par Al Borno, de Lasa & Hertzmann (2013) permet d'accomplir des mouvements complexes impliquant de nombreux contacts, tels que se lever du sol, se tenir en équilibre sur ses deux mains. La méthode n'est pas basée sur une base de données de mouvements et un terme d'énergie d'optimisation permet de créer de la variété dans le comportement du personnage. Dans notre projet, nous souhaitons contrôler le style de l'animation avec peu de paramètres utilisateurs et l'utilisation d'une fonction d'énergie reste une solution possible, mais ne permet pas d'assurer la production d'un certain style de mouvement en particulier. Leur méthode utilise des objectifs d'optimisation simple tels qu'avoir une seule main en contact avec le sol, ou le centre de masse au-dessus d'une certaine hauteur dans un intervalle de temps. Les paramètres sont prédéfinis sur une animation complète qui limite la généralisation à d'autres mouvements.

Les travaux de Liu, Panne & Yin (2016) présentent une méthode pour apprendre des stratégies de rétroaction robustes utilisant des cadres d'animations de capture de mouvement ainsi que

les chemins de transition entre les cadres d'animations. Le résultat est un graphe de contrôle qui prend en charge la simulation physique en temps réel de plusieurs personnages, chacun capable d'un large éventail de compétences de mouvement robustes, comme la marche, la course, les virages serrés, des roulades, les spins-kicks et les flippes. Cette méthode n'utilise pas l'apprentissage par renforcement, mais est capable de produire de très bons résultats pour combiner différents mouvements courts afin d'accomplir des tâches décrites par des objectifs de haut niveau. Combiner des mouvements fait partie d'une composante de notre recherche, car nous souhaitons pouvoir synthétiser différents mouvements de levée du sol tout en ayant des transitions lisses. Cette méthode est simple, mais permet uniquement d'activer une seule primitive à la fois dans le graphe de contrôle qui peut s'avérer potentiellement un problème pour effectuer la transition entre deux politiques de contrôle.

1.2 Biomécanique

Les mouvements de maintien debout ont été bien étudiés dans la littérature biomécanique et cinématique. En particulier, les mouvements de lever du sol sont étudiés en matière de performance physique pour des personnes âgées dans les travaux de Adams & Tyson (2000) et Klima *et al.* (2016). De notre connaissance, aucune taxinomie existe pour classifier les différents styles de mouvement de lever du sol. Cependant, dans les travaux de Bohannon & Lusardi (2004) des stratégies sont identifiées pour décrire certains mouvements tels que :

- partir d'une posture couchée sur le dos et sur le ventre pour le style poussé vers le haut quadrupède ;
- un mouvement pour s'asseoir droit pour rouler ;
- s'asseoir sur le côté et pivoter avec les genoux.

Pour notre projet, nous reproduisons les différents styles pour chacune des stratégies identifiées.

1.3 Robotique

Des contrôleurs pour maintenir des personnages debout apportent un certain intérêt depuis très longtemps dans la communauté de la robotique. Morimoto & Doya (1998) proposent un cadriciel hiérarchique basé sur l'apprentissage machine pour générer des mouvements de maintien debout pour un humanoïde simple. Kanehiro, Fujiwara, Hirukawa, Nakaoka & Morisawa (2007) sont en mesure de générer des mouvements de lever du sol en interpolant la pose courante du robot avec la pose la plus proche dans un graphe d'état. D'autres travaux de Fujiwara *et al.* (2003) ont également divisé les mouvements de lever du sol en plusieurs phases avec un graphe de contact.

1.4 Apprentissage par renforcement profond

L'apprentissage par renforcement profond connaît un très fort engouement depuis les dernières années dans le domaine de la recherche. C'est une sous-branche de l'apprentissage profond. Ce domaine a connu une grande avancé afin de résoudre des problèmes complexes. Sa particularité est d'apprendre par essai et erreur n'ayant pas accès à certaines données. Dans notre projet, nous n'avons pas accès à un modèle de l'environnement physique. Pour cela, il est nécessaire de collecter des données en interagissant dans celui-ci afin de s'améliorer. Dans cette section, nous présentons les travaux majeurs utilisant cette méthode pour des environnements simulés physiquement.

1.4.1 Espace des actions

Dans les méthodes utilisant l'apprentissage par renforcement profond afin d'apprendre un modèle physique, les travaux de Peng & van de Panne (2017) portent sur l'importance du choix de l'espace d'action. Leurs conclusions sont que certains choix de conceptions peuvent affecter le temps d'apprentissage et la performance. Ils démontrent que l'utilisation de contrôleur PD (proportionnel dérivé) utilisant un retour local permet d'obtenir les meilleures performances. Les contrôleurs PD permettent d'asservir la position d'une articulation en agissant sur le couple d'un axe. Cela permet aussi d'amener une abstraction permettant de transposer les politiques

de contrôle apprises. Avec cette étude et les excellents résultats des projets suivants effectués par Peng *et al.* (2018a), nous avons choisi d'utiliser ce même espace d'action. Tel qu'indiqué dans leurs travaux l'ajustement des gains reste cependant une opération manuelle qui doit être refaite sur chaque morphologie de personnage afin d'atteindre de bons résultats. Cependant, cette méthode limite la généralisation aux autres morphologies. L'utilisation de gains identiques entre plusieurs morphologies est sujette à faire échouer l'apprentissage des contrôleurs. Pour cela, nous avons utilisé les mêmes paramètres d'amortissement et de raideur du personnage humanoïde pour s'assurer de produire des mouvements naturels tel que présentés dans leurs résultats.

1.4.2 Approche basé sur l'imitation contradictoire

Merel *et al.* (2017) étendent la méthode d'apprentissage génératif par imitation antagoniste (GAIL) pour entraîner une politique à reproduire des mouvements proches du comportement humain. La méthode GAIL est basé sur les réseaux antagonistes génératifs (GAN). La méthode utilise des trajectoires de référence pour récupérer une fonction de coût afin d'apprendre une politique. Un discriminateur est entraîné comme un classificateur qui indique si la pose du personnage vient de la simulation ou d'un mouvement provenant de la base de données de mouvements de référence. Leurs travaux entraînent des contrôleurs de bas niveau à accomplir certaines tâches spécifiques en utilisant une collection d'animation de références. Les contrôleurs de bas niveau peuvent ensuite être utilisés pour accomplir des tâches avec un contrôleur de haut niveau. L'avantage d'utiliser un GAIL et d'éviter de définir explicitement manuellement des métriques d'imitation dans la fonction de récompenses. Pour cela, un module de discrimination est entraîné pour agir comme un classificateur afin de différencier si un mouvement produit provient de la base de données de mouvements d'exemples.

Peng *et al.* (2021) proposent d'éviter de concevoir manuellement des objectifs d'imitation et mécanismes de sélection de mouvement en utilisant une approche entièrement automatisée. Leur méthode vient améliorer les travaux précédant de Peng *et al.* (2018a) qui nécessitent la sélection manuelle d'une animation de référence à imiter. Un discriminateur permet d'indiquer

si le mouvement du personnage provient de la base de données afin de se concentrer sur une tâche de haut niveau à accomplir. Cette méthode est basée sur les travaux de Merel *et al.* (2017) qui implique l'utilisation de la méthode GAIL. Par exemple, un personnage traversant un obstacle peut utiliser une base de données qui contient plusieurs clips utiles tels que courir, sauter, faire une roulade. La méthode est capable de synthétiser des comportements novateurs qui sont similaires à ceux qui sont présents dans la base de données. Elle généralise également dynamiquement à partir de l'ensemble des données. Cette technique permet de produire des mouvements de qualité égalant les techniques de « motion tracking » tout en utilisant un large éventail de clips de mouvement non structurée. Un personnage devra inévitablement s'écarter des mouvements de référence pour exécuter efficacement une tâche donnée. Par conséquent, l'intention n'est souvent pas que le personnage suit de proche un mouvement particulier, mais d'adopter des caractéristiques comportementales générales décrites dans l'ensemble des données. Dans ce travail, un utilisateur peut sélectionner une tâche d'objectif de haut niveau à accomplir. Par la suite, il sera effectué en appliquant un style basé sur les clips de références non structurés. Cette méthode ne s'intéresse pas à créer de nouveaux mouvements, mais de composer plutôt plusieurs clips contenus dans la base de données afin d'accomplir une tâche. De ce fait pour notre projet, le résultat pourrait être biaisé fortement par les mouvements contenus dans la base de données. Dans notre recherche, la méthode GAIL peut s'appliquer à notre solution, car elle permet d'assurer un mouvement naturel qui ressemble à des postures qui sont présentes dans la base de données. Cependant, nous avons décidé d'utiliser une autre méthode. Comme indiqué par Peng *et al.* (2021), le GAIL est une méthode basée sur le GAN qui est susceptible de provoquer des effondrements de modes. C'est-à-dire que la politique de contrôle est incitée à imiter seulement un sous-ensemble de comportements présents dans une base de données très large. De plus, elle peut ignorer d'autres comportements qui peuvent s'avérer plus approprier pour accomplir une tâche particulière. Nous supposons que ceci peut devenir problématique dans notre application de création de courbe de style.

Peng *et al.* (2022) introduisent une méthode basée sur leurs travaux précédents qui utilise la méthode GAIL. Pour cela, leur méthode utilise une base de données très larges afin d'apprendre

des compétences polyvalentes qui sont réutilisables pour animer les personnages. L'imitation contradictoire est combinée avec l'apprentissage par renforcement non supervisé afin de développer des compétences d'apparence naturelles. Une étape préliminaire consiste à apprendre un contrôleur de bas niveau avec une variable latente qui représente un comportement présent dans la base de données. Dans une deuxième étape, la politique de bas niveau peut être utilisée pour apprendre une tâche spécifique. Ils augmentent significativement leurs puissances de calcul en distribuant massivement les différentes instances de simulation physique. Pour cela, ils utilisent les plus récentes techniques de NVIDIA permettant de simuler sur les cartes graphiques de dernière génération. Leur méthode démontre qu'il est possible d'accomplir une tâche utilisant un riche répertoire de compétences tout en fournissant une fonction de récompense simple.

1.4.3 Apprentissage sans mouvements de référence

Heess *et al.* (2017) entraînent un personnage humanoïde pour accomplir des tâches de locomotions sans utiliser de mouvements de référence. Leurs travaux font évoluer un personnage dans des environnements physiques complexes comportant des murs, des terrains irréguliers et des gaps nécessitant des sauts du personnage. La fonction de récompense est très simple et consiste à récompenser l'agent lorsqu'il évolue vers l'avant dans l'environnement. Quoique le personnage soit capable d'accomplir certaines tâches de façon impressionnante, leurs résultats démontrent qu'il est difficile de produire des mouvements naturels en l'absence d'une base de données de mouvements d'exemples. Un point intéressant en relation avec notre projet est la gestion des terrains non plats. Pour cela, ils utilisent une grille de points de hauteur échantillonnée autour du personnage. Nous avons choisi la même technique afin que notre personnage soit en mesure de connaître la forme du terrain, cependant elle peut être coûteuse, car elle requiert un échantillonnage de nombreux points.

Xie, Ling, Kim & van de Panne (2020) font évoluer un personnage sur des tremplins. En effet, ils sont en mesure d'apprendre à un agent à se déplacer de tremplins en tremplins sans utiliser une animation de référence. Ils utilisent l'apprentissage par curriculum qui consiste à augmenter la difficulté pendant l'entraînement. Par exemple, dans les phases préliminaires l'agent apprend à

se déplacer sur un sol plat et ensuite sur des tremplins. La distance et hauteur sont graduellement augmentées selon l'état de l'apprentissage. Dans notre projet, nous utilisons une méthode similaire qui consiste à augmenter progressivement la difficulté du terrain. Dans d'autres travaux similaires, cette technique semble nécessaire pour réussir à apprendre des contrôleurs dans des environnements difficiles.

La méthode de Naderi, Babadi, Roohi & Hamalainen (2019) comporte l'apprentissage par renforcement et permet à un personnage simulé physiquement de grimper utilisant des prises. L'environnement est composé de blocs et le grimpeur démarre à partir du sol face au mur et ensuite on demande de commencer par placer les mains et pieds à une ou plusieurs prises prédéfinies. L'objectif du grimpeur est d'atteindre un objectif précis (généralement en haut du mur) en effectuant divers mouvements d'escalade, certains peuvent être difficiles et encombrants. La principale difficulté du problème de l'escalade est que la formation nécessite une grande quantité de simulations d'expérience afin d'explorer un éventail suffisamment large de mouvements d'escalade. Afin d'atténuer ce problème, ils utilisent des stratégies particulières d'initialisation de l'état de référence du personnage. Ils comparent deux stratégies d'initialisation en observant leurs effets sur l'exploration de divers mouvements. Ils démontrent aussi que la douceur des mouvements peut être synthétisée à l'aide de l'utilisation de différents paramètres d'action. Pour cela, ils utilisent des contrôleurs PD identiques aux travaux de Peng & van de Panne (2017), mais ils répètent l'action pendant plusieurs pas de temps afin de lisser le mouvement.

Des travaux récents de Tao, Wilson, Gou & Van de Panne (2022) permettent de synthétiser des mouvements de lever du sol sans utilisation de mouvements de références. Pour obtenir des mouvements naturels, ils s'appuient sur la modélisation d'un personnage qui suit le plus fidèlement les capacités humaines. Pour cela, les limites des couples et des articulations sont ajustées au plus proche de la biomécanique humaine. Leurs résultats démontrent qui n'est pas possible d'apprendre des mouvements de lever du sol sans l'utilisation d'une approche par curriculum. Pour cela, l'entraînement démarre avec une version plus forte du personnage, qui permet ensuite d'apprendre une version plus faible s'approchant des capacités humaines. Pendant l'entraînement, les limites de couples sont augmentées au-delà des limites humaines afin de

trouver des solutions préliminaires. Ensuite, une transition graduelle vers des couples normaux est effectuée jusqu'à la fin l'entraînement. Une stratégie finale consiste à utiliser une politique de contrôle maître qui enseigne à une politique élève à effectuer les mêmes mouvements à des vitesses différentes. Dans notre projet, nous souhaitons également moduler la vitesse. Cependant, une approche différente consiste à faire varier la base de temps d'une courbe de style d'entrée. Nos mouvements naturels sont assurés par un espace latent, qui permet d'apprendre à partir du sol tout en ayant un personnage qui respecte les capacités humaines. Leurs travaux traitent uniquement de l'apprentissage de mouvements de lever du sol sur un terrain plat. Nos travaux sont complémentaires, nous souhaitons moduler le style tout en fournissant quelques signaux à un utilisateur. De leur côté, ils souhaitent produire des mouvements naturels divers, mais ils possèdent moins de contrôles sur le style du mouvement.

1.4.4 Apprentissage par imitation

Les travaux de Peng *et al.* (2018a) visent à apprendre un contrôleur pour un personnage physique utilisant une animation de référence. Les travaux consistent à améliorer la qualité des contrôleurs appris est d'incorporer la capture de mouvement ou des données d'animations écrites à la main. Pour cela, ils adoptent une approche simple en récompensant directement le contrôleur appris pour avoir produit des mouvements qui ressemblent à des données d'animation de référence, tout en atteignant des objectifs de tâches supplémentaires. Ils peuvent produire des comportements qui sont presque indiscernables en apparence du mouvement de référence en l'absence de perturbations, évitant ainsi de nombreux artefacts présentés par les précédents algorithmes d'apprentissage. Comme nous voulons synthétiser une base de données larges de mouvements, leur technique ne peut pas s'appliquer formellement au sujet des objectifs d'imitation. Cependant, leur méthode de gestion des terrains irréguliers est intéressante pour notre projet. Pour cela, le problème est géré comme un système de vision en traitant le champ de hauteur à travers un réseau de convolution afin d'extraire les caractéristiques importantes et de réduire la taille du vecteur d'état. Nous pensons également que c'est la méthode à suivre. Cependant, nous avons

décidé d'utiliser une méthode plus facile à mettre en place qui consiste échantillonné une grille de point de hauteur autour du personnage.

Chentanez, Müller, Macklin, Makoviychuk & Jeschke (2018) s'intéressent à l'apprentissage de mouvement par imitation, mais également à la récupération de l'agent après une chute. En général, les contrôleurs entraînés ne peuvent pas toujours récupérer après l'application de fortes perturbations. Pour cela, deux contrôleurs travaillent ensemble avec le premier pour suivre une animation de référence utilisant une base de données de capture de mouvement. La deuxième apprend à récupérer à partir d'une position aléatoire vers une pose déterminée pour redémarrer la tâche de locomotion. Dans notre projet, nous souhaitons également lever le personnage du sol dans diverses situations, mais les mouvements produits par leurs contrôleurs restent non naturels et ceci demeure un critère très important pour nous.

Bergamin *et al.* (2019) proposent une nouvelle approche pour contrôler un personnage animé par la physique de façon naturelle et qui suit des mouvements de capture de référence. L'apprentissage consiste à faire évoluer un personnage dans le plus de scénarios possibles et d'utiliser la correspondance de mouvement (*Motion matching*) qui utilise une base de données de mouvement pour récupérer la meilleure animation qui correspond à l'état du personnage à un moment donné. Le contrôleur permet finalement de diriger le personnage dans toutes les directions avec un joystick tout en ayant un comportement naturel et répondant aux perturbations extérieures. Dans leurs contributions, ils utilisent un filtrage des actions qui s'avèrent intéressant pour lisser les mouvements. Nous avons également décidé d'appliquer ce principe pour pouvoir trouver des positions stables sans limiter l'exploration pendant l'entraînement. Bergamin *et al.* (2019) indiquent aussi que ce lissage permet d'obtenir de meilleures animations, mais aussi permet d'aider l'apprentissage des politiques de contrôle.

Park, Ryu, Lee, Lee & Lee (2019) proposent une méthode qui permet d'apprendre un répertoire varié des compétences pour un personnage simulé physiquement. Le personnage contrôlable par l'utilisateur est en mesure de prédire des mouvements agiles afin de s'adapter à son environnement. Leurs architectures sont basées sur les RNN et permettent de générer des mouvements dans

un environnement simulé par la physique. Le générateur de mouvement permet de guider la dynamique en fournissant une séquence de cadre clé d'animation dans le futur, ils démontrent que les prédictions de mouvements futurs permettent de faciliter l'apprentissage sur une base de données très large.

1.4.5 Espace latents

Les modèles basés sur les espaces latents entraînés sur des bases de données de capture de mouvements ont démontré d'excellents résultats. Ils permettent de produire des mouvements naturels tout en respectant un style. Les travaux de Yuan & Kitani (2020) utilisent un auto-encodeur variationnel (VAE) pour imiter de façon robuste des trajectoires de capture de mouvement à l'aide de modèles dynamiques.

Merel *et al.* (2019) mettent l'accent sur l'apprentissage d'un module moteur qui en-capsule un espace latent de mouvements humains. Ils démontrent qu'il est possible de compresser dans un modèle latent des milliers de politiques expertes. Le but n'est pas de reproduire parfaitement un mouvement en particulier, mais de synthétiser des mouvements novateurs qui sont proches des exemples dans la base de données. Merel *et al.* (2020) s'intéressent plus particulièrement par la suite à accomplir des tâches qui demandent une interaction avec des objets en utilisant la vision. Ils utilisent également le module moteur de bas niveau développé en 2019 qui est dérivé de la base de données de démonstration de mouvements utilisée dans divers scénarios. Une politique de plus haut niveau interface avec le module moteur de bas niveau afin d'accomplir des tâches basées sur une vision égocentrique. Les deux tâches difficiles sont d'interagir avec des objets larges impliquant les deux mains.

Dans la catégorie des méthodes non basé sur la physique, les travaux de Ling, Zinno, Cheng & Van De Panne (2020) proposent un modèle génératif VAE de mouvement (MVAE) qui est capable de produire des mouvements de haute qualité pour un personnage. Ils démontrent que l'apprentissage par renforcement peut produire des politiques compactes qui sont capables de reproduire une majorité des mouvements présents dans une base de données. Les espaces latents permettent

de réduire significativement la dimension d'une base de données en compressant l'information dans un encodeur. Par la suite, des mouvements présents dans la base de données peuvent être reconstruits de façon fidèle avec le décodeur. Le décodeur est un réseau de neurones qui a l'avantage d'être rapide et demande une taille de stockage très faible. Les espaces latent peuvent être couplés avec un but à atteindre. Cette méthode utilise les mouvements "kinematic" c'est-à-dire qui n'utilise pas la simulation physique pour les personnages. La technique utilise une base de données de mouvements qui sont effectués dans différentes directions. Finalement, l'utilisateur peut diriger le personnage dans différentes directions avec un joystick, il est possible aussi de spécifier une trajectoire à suivre. Il est intéressant de constater la variété de poses reconstruite à partir d'un espace latent. En effet, dans notre projet nous souhaitons reproduire une grande variété de poses à partir d'une base de données larges et dont les espaces latents semblent performant pour résoudre ce problème.

Dans les travaux de Won, Gopinath & Hodgins (2020), ils sont capables de combiner une large base de données de clip de mouvement pour animer un personnage. L'idée est de créer des agents experts dans un sous-groupe de mouvement pour ensuite les combiner dans un contrôleur plus général avec un système d'activation appelé « Gating », qui est un système de poids activant les experts qui sont spécialisés dans un sous-groupe de mouvement.

Peng, Chang, Zhang, Abbeel & Levine (2019) parlent d'une technique nommée MCP, qui est une méthode utilisant des mouvements primaires préalablement appris et qui peuvent être composés pour produire un spectre de compétences. Un agent est entraîné à partir de zéro pour une tâche spécifique telle que marcher, courir. Ensuite, ils utilisent les compétences apprises pour des tâches subséquentes. Une limitation du modèle est qu'un seul mouvement primaire peut être activé à un pas de temps particulier. Cette technique permet à un personnage d'accomplir des tâches spécifiques en combinant des mouvements primaires. Par exemple, il est possible de récupérer et de transporter un objet à une position spécifique. Une fonction de porte est entraînée pour accomplir l'objectif en produisant des poids d'influence de chaque primitive pour composer le mouvement final. Nous souhaitons combiner un contrôleur de lever du sol et de maintient debout qui sont l'équivalent nos deux primitives. L'inconvénient de cette méthode est que le

contrôleur de porte doit être entraîné à chaque fois qu'une nouvelle primitive est apprise. Dans notre projet, l'espace latent est formé une seule fois pour être ensuite utilisé pour accomplir une tâche spécifique.

Won *et al.* (2021a) ont réussi à concevoir des contrôleurs de personnages ayant la capacité de combattre entre eux. Ils se sont inspirés des méthodes d'apprentissage employées dans le sport en entraînant d'abord les compétences seules et ensemble d'apprendre à les combiner pendant des sessions de combat d'entraînement. Basés sur leurs travaux précédents, ils utilisent des contrôleurs de bas niveau formés indépendamment afin d'accomplir des tâches de contrôle continu difficiles ayant un plus haut niveau. Ce travail est novateur, car il est capable de faire interagir plusieurs agents ensemble de façon réaliste en temps réel.

Dans nos travaux nous utilisons un auto-encodeur variationnel (VAE) similaire à celui proposé par Yin *et al.* (2021) qui est entraîné sur une base de données comprenant des mouvements naturels afin d'assurer une reproduction naturelle des mouvements reproduits dans l'environnement physique. La différence est que nos travaux utilisent un C-VAE qui ajoute le vecteur conditionnel **c** qui permet de filtrer les poses possibles à partir de nos caractéristiques d'entrée. Leurs travaux consistent à apprendre à un personnage animé par la physique pour découvrir de nouvelles compétences athlétiques telles que le saut en hauteur. Une approche basée sur l'apprentissage par curriculum est utilisée afin d'augmenter la difficulté au fil de l'apprentissage. Ils proposent un algorithme permettant de produire dans une première phase des stratégies diverses autour d'un même état initial de départ et dans une deuxième phase pour d'enrichir les stratégies uniques. La première phase consiste à faire varier l'état initial, tel que la pose et la vitesse au moment du décollage pour effectuer le saut athlétique. Dans deuxième phase, ils explorent des variations potentielles de stratégie pour franchir les obstacles à partir d'états initiaux de décollage qui ont été découverts dans la première phase.

Won, Gopinath & Hodgins (2022) utilisent un C-VAE tel que proposé dans notre méthode. Ils conditionnent la distribution avec l'état courant. L'état courant au pas de temps suivant est utilisé en entrée pour l'encodeur. Pour reconstruire le prochain état de sortie avec le décodeur, un

paramètre latent **z** est échantillonné. La différence avec notre méthode est que nous conditionnons notre C-VAE avec les différentes courbes de styles au lieu d'utiliser l'état courant. Nos courbes de styles sont composées de trois valeurs scalaires encodées dans le vecteur **c** qui se compose de la position et l'orientation du torse à atteindre. Dans leurs travaux, ils indiquent que les conditions de pose dans l'état courant doivent correspondre aux proportions du personnage dans la base de données. Plus précisément, ils ont rencontré des problèmes lors de l'utilisation d'un personnage plus grand que celui utilisé dans la base de données pour apprendre le C-VAE. Par exemple, cette différence provoque des problèmes pour faire correspondre la position l'articulation racine de référence par rapport au sol. Ils utilisent une stratégie similaire à la nôtre pour s'adapter aux nouveaux environnements. Pour cela, un décalage est ajouté aux actions pour s'écarter du mouvement original afin de produire de nouveaux mouvements. Ils indiquent que le personnage échoue sur les terrains irréguliers sans cette composante. De plus, ils introduisent un paramètre supplémentaire afin de contrôler l'amplitude de celui-ci. En comparaison avec notre méthode, nous utilisons une pénalité qui limite l'utilisation des décalages. Leurs résultats démontrent qu'ils sont capables de produire des mouvements naturels tout en s'adaptant aux terrains irréguliers pour des mouvements de locomotion. Ils ont remarqué que la performance peut varier grandement en fonction la diversité de mouvements présents dans la base donnée. De plus, ils reportent que les résultats peuvent varier légèrement en fonction l'utilisation de différents moteurs physiques.

1.4.6 Apprentissage supervisé

Fussell, Bergamin & Holden (2021) proposent une méthode novatrice basée sur l'apprentissage supervisé permettant d'apprendre des mouvements de locomotion complexes pour un humanoïde. N'ayant pas accès à un modèle de la simulation physique, les méthodes plus traditionnelles des années précédentes sont basées uniquement sur l'apprentissage par renforcement profond. Dans leurs travaux, ils entraînent une approximation d'un modèle de l'environnement physique. Celui-ci est ensuite utilisé comme une fonction permettant de dériver un comportement physique au cours du temps. Deux modèles sont alors appris simultanément. Le premier modèle pour

prédire la dynamique du monde et le deuxième qui produit les meilleures actions afin de minimiser une erreur dans le suivi de la posture. Leurs résultats démontrent que la méthode accomplie des meilleurs résultats en termes de qualité tout en ayant un temps d'entraînement significativement plus court. Ce résultat vient bousculer les standards qui se sont installés durant les dernières années. Ils indiquent notamment que l'algorithme d'apprentissage Proximal Policy Optimisation (PPO) nécessite un nombre très significatif d'échantillons. Leur méthode permet d'apprendre en 20 heures une base de données contenant 6 heures de mouvements, là où l'algorithme PPO peine à produire de bons résultats sur un temps équivalent. Ils indiquent que PPO semble mieux fonctionner sur des bases de données plus petites, mais n'atteint pas les performances obtenues par leur méthode. Ils sont en mesure de réduire le temps d'entraînement significativement par rapport l'utilisation de PPO. De plus, ils résolvent certains problèmes très difficiles comme de l'adaptation aux terrains très divers et irréguliers. Ces deux derniers font partie des axes de notre recherche.

1.5 Synthèse

Les travaux présentés permettent de guider nos choix de conception. Les espaces latents semblent les plus appropriés pour créer de nouveaux styles tout en assurant de produire des mouvements naturels, de plus l'ajout de petits décalages sur une pose naturelle semble plus approprié pour s'adapter à des terrains non plats. Une technique d'apprentissage par curriculum semble nécessaire pour apprendre avec succès des tâches difficiles. Sachant qu'un agent explore un espace de solution proche de sa connaissance initiale, il devient très difficile d'apprendre une tâche qui s'écarte fortement de son champ de compétence acquis. Les travaux de Peng *et al.* (2018a) et Peng *et al.* (2021) restent au sommet de l'état de l'art. Plusieurs décisions de notre conception ont été inspirées de leurs travaux. On dénote par exemple la modélisation du personnage physique, ainsi que la conception du vecteur d'état qui est issue de leurs travaux. Finalement, l'algorithme d'apprentissage PPO semble un choix sécuritaire, celui-ci a notamment appris des politiques à accomplir des tâches très complexes dans des environnements riches.

CHAPITRE 2

MÉTHODOLOGIE

Dans ce chapitre, nous décrivons en détails tous les sous-éléments de la méthode qui sont requis afin d'accomplir des tâches de levées du sol de façon naturelle et une variation de styles contrôlables dans les mouvements. La Figure 2.1 illustre l'architecture de notre cadriciel afin d'accomplir les objectifs.

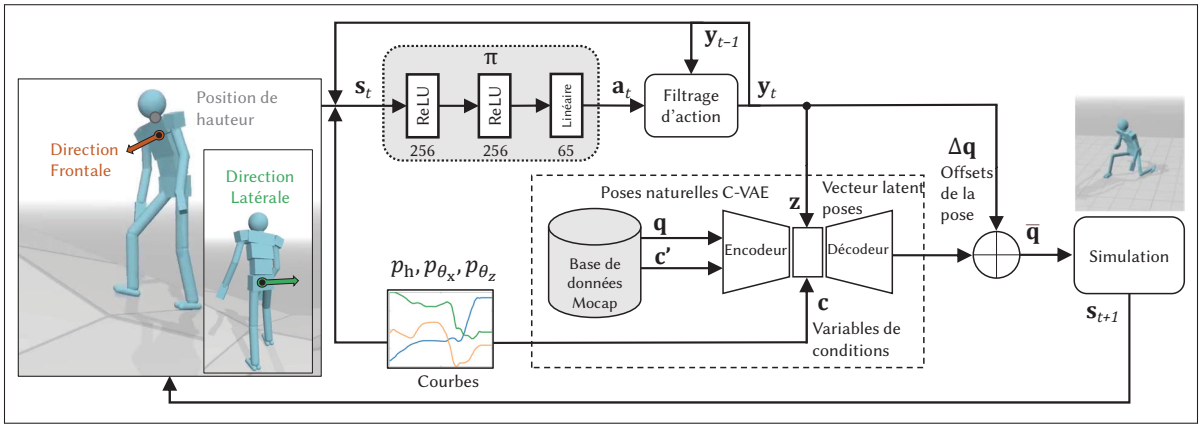

Figure 2.1 Vue générale de l'architecture

2.1 Vue générale

Une politique de contrôle π est entraînée pour la tâche de lever du sol afin de suivre des courbes définissant la hauteur et l'orientation du torse, ainsi que pour le maintien du personnage debout de façon robuste. Une politique de contrôle choisit la meilleure action $\mathbf{a}_t$ à chaque pas de temps t selon l'observation de l'environnement $\mathbf{s}_t$ qui produit une transition vers l'état suivant $\mathbf{s}_{t+1}$. Pendant l'entraînement de la politique, une récompense pour l'agent est utilisée afin d'indiquer si l'action a abouti à une bonne transition, ou une mauvaise. Dans notre cas, une bonne transition consiste à suivre les courbes de styles fournit. Plus de détails sur les courbes sont fournis dans la section 2.6. L'auto-encodeur variationnel conditionnel (C-VAE) est entraîné avec la base de données LAFAN1 d'Ubisoft Harvey, Yurick, Nowrouzezahrai & Pal (2020) contenant des

sessions de capture d'un humain effectuant des mouvements de marcher, courir, tomber et se lever du sol, plus de détails sur le C-VAE sont disponibles dans la section 2.5. Le C-VAE permet d'assurer l'utilisation d'une posture naturelle du personnage qui est appliquée sur les axes servos des articulations du personnage. La distribution des poses du décodeur est conditionnée par le vecteur $\mathbf{c}$. Ceci permet de produire des poses qui respectent la hauteur et l'orientation du torse cible. Avant l'entraînement de l'encodeur, le vecteur $\mathbf{c}'$ est pré-calculé en fonction de la posture du personnage $\mathbf{q}$. Pour cela, chaque pose $\mathbf{q}$ est appliquée au personnage afin de déduire le vecteur $\mathbf{c}'$ qui contient la hauteur, l'orientation frontale et latérale du torse. L'entraînement de l'encodeur est une étape préalable à l'entraînement de la politique de contrôle π. Une fois cette étape terminée, seul le décodeur est utilisé par la politique de contrôle. Afin de s'adapter aux contraintes de la physique et de la variation du terrain, un petit décalage $\Delta_{\mathbf{q}}$ est ajouté sur chaque joint qui nous permet de nous écarter légèrement de la posture naturelle. Les décalages permettent également de produire des mouvements qui ne font pas forcément partie de la base de données. Afin de contrôler le style, trois courbes sont utilisées afin de spécifier la hauteur p_h, l'angle frontal p_{θ_x} et latéral p_{θ_z} du torse par rapport au sol. Le filtrage d'action permet de lisser le mouvement pendant l'entraînement afin de trouver de meilleurs placements pour les effecteurs finaux afin de stabiliser le personnage en position de maintien.

2.2 Modélisation du personnage

Le personnage physique est un humanoïde possédant 19 articulations afin de contrôler l'orientation des parties du corps. La Figure 2.2 illustre notre personnage qui possède 49 dégrées de libertés (DOF) que la politique peut contrôler.

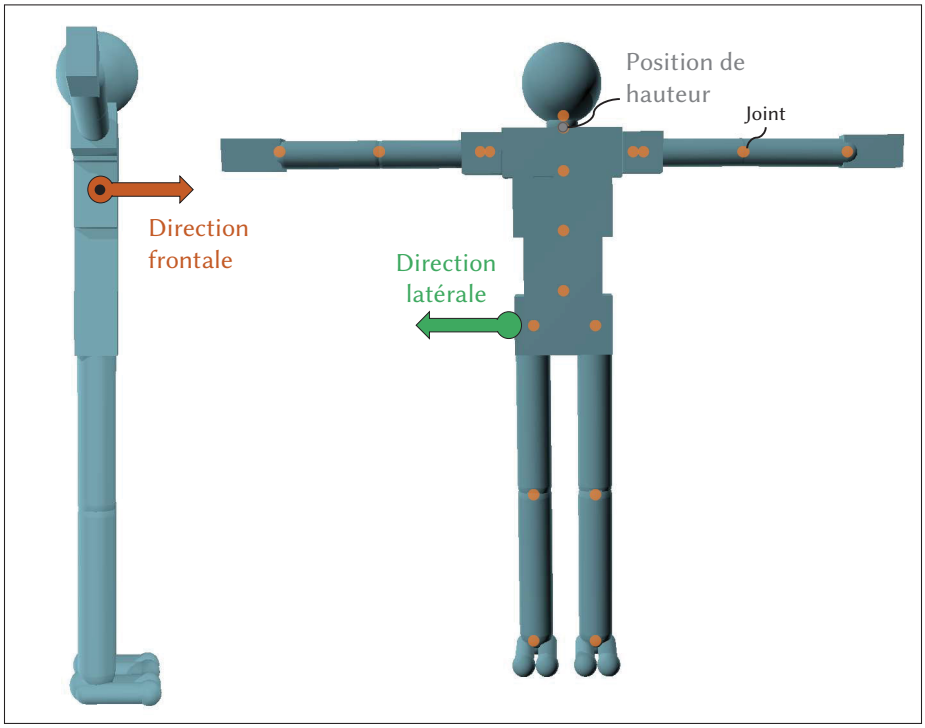

Figure 2.2 Modélisation du personnage physique 3D

Le nombre de degrés de liberté est choisie en fonction du nombre d'informations couramment retrouvées dans les animations de personnages 3D. Le logiciel Vortex Studio de CM Labs nous permet de modéliser le personnage en utilisant des formes de collisions simples (boîtes, sphères ou cylindres). Les géométries simples ont la propriété d'être rapide à calculer et sont généralement suffisantes pour simuler les collisions entre le personnage et l'environnement dans les applications interactives. Cependant, les proportions comme la taille et le poids sont importants pour simuler un comportement réel. Pour cela, nous utilisons une taille de 1.7 m et distribuons le poids des différentes parties du corps selon l'étude de Plagenhoef, Evans & Abdelnour (1983) avec un total de 65 kg. Chaque articulation du personnage est

contrôlée avec un axe asservi en position utilisant la dérivée et la proportionnelle (PD). Pour cela, une erreur de position est calculée afin de contrôler le couple de l'axe. La proportionnelle représente le gain de correction par rapport à l'erreur de position et la dérivée par rapport à la vitesse de rotation de l'articulation. Cette méthode de contrôle est particulièrement efficace et a démontrée des résultats impressionnant dans des travaux récents tels que Peng *et al.* (2018a), Peng *et al.* (2021). Afin de s'approcher du comportement humain réel, les paramètres des axes sont ajustés sur la base des travaux de Peng *et al.* (2018a) suivant le Tableau 2.1. Nous avons constaté que l'utilisation de valeurs de couples réalistes sont très importants pour synthétiser des mouvements naturels. Des expérimentations antérieures ont démontré des comportements très robotique et irréalistes en utilisant des valeurs de couples importants. Cependant, nous avons limité les angles des articulations sur les axes majeurs tout en ayant un petit jeu supplémentaire. Les limites des axes sont disponibles dans le Tableau 2.2. La direction frontale tel qu'illustré dans la Figure 2.2 est calculé utilisant le produit croisé entre deux vecteurs tels que le vecteur qui connecte le coup et les hanches et le vecteur latéral. L'articulation du coup est utilisée pour la position de suivi de hauteur.

Tableau 2.1 Paramètres de couple des axes servo PD

Nom de l'articulation	**Couple maximum** (Nm)	**Couple minimum** (Nm)	**Raideur** ($Nm.rad^{-1}$)	**Amortissement** ($kg.m^2s^{-1}$)
Colonne vertébrale (haut)	170	-170	5000	500
Colonne vertébrale (milieu)	170	-170	5000	500
Colonne vertébrale (bas)	170	-170	5000	500
Jambes supérieures	170	-170	2500	250
Jambes inférieures	125	-125	2500	250
Pieds	75	-75	2000	200
Tête	40	-40	500	50
Coup	40	-40	500	50
Épaules	85	-85	2000	200
Bras	85	-85	2000	200
Avant-bras	50	-50	1500	150
Mains	40	-40	1000	100

Tableau 2.2 Limitation des axes

Nom de l'articulation	**Axe X** $[min, max]$ **(*rad*)**	**Axe Y** $[min, max]$ **(*rad*)**	**Axe Z** $[min, max]$ **(*rad*)**
Colonne vertébrale (haut)	$[0, 0.5]$	$[-0.2, 0.2]$	$[-0.2, 0.2]$
Colonne vertébrale (milieu)	$[-0.1, 0.6]$	$[-0.3, 0.3]$	$[-0.3, 0.3]$
Colonne vertébrale (bas)	$[-0.1, 0.6]$	$[-0.3, 0.3]$	$[-0.3, 0.3]$
Jambes supérieures	$[-0.4, 2.57]$	$[-1.0, 1.0]$	$[-1.0, 1.0]$
Jambes inférieures	$[-3.14, 0.0]$	$[0.0, 0.0]$	$[0.0, 0.0]$
Pieds	$[-1.57, 1.57]$	$[-1.0, 1.0]$	$[-1.0, 1.0]$
Tête	$[-1.0, 1.0]$	$[-1.0, 1.0]$	$[-1.0, 1.0]$
Coup	$[-1.0, 1.0]$	$[-1.0, 1.0]$	$[-1.0, 1.0]$
Épaules	$[-3.14, 3.14]$	$[-3.14, 3.14]$	$[-3.14, 3.14]$
Bras	$[-3.14, 3.14]$	$[-3.14, 3.14]$	$[-3.14, 3.14]$
Avant-bras	$[-3.14, 0.0]$	$[0.0, 0.0]$	$[0.0, 0.0]$
Mains	$[-3.14, 3.14]$	$[-1.57, 1.57]$	$[-1.57, 1.57]$

2.3 Modélisation de l'environnement

La Figure 2.3 illustre l'environnement utilisé pour l'entraînement du personnage physique.

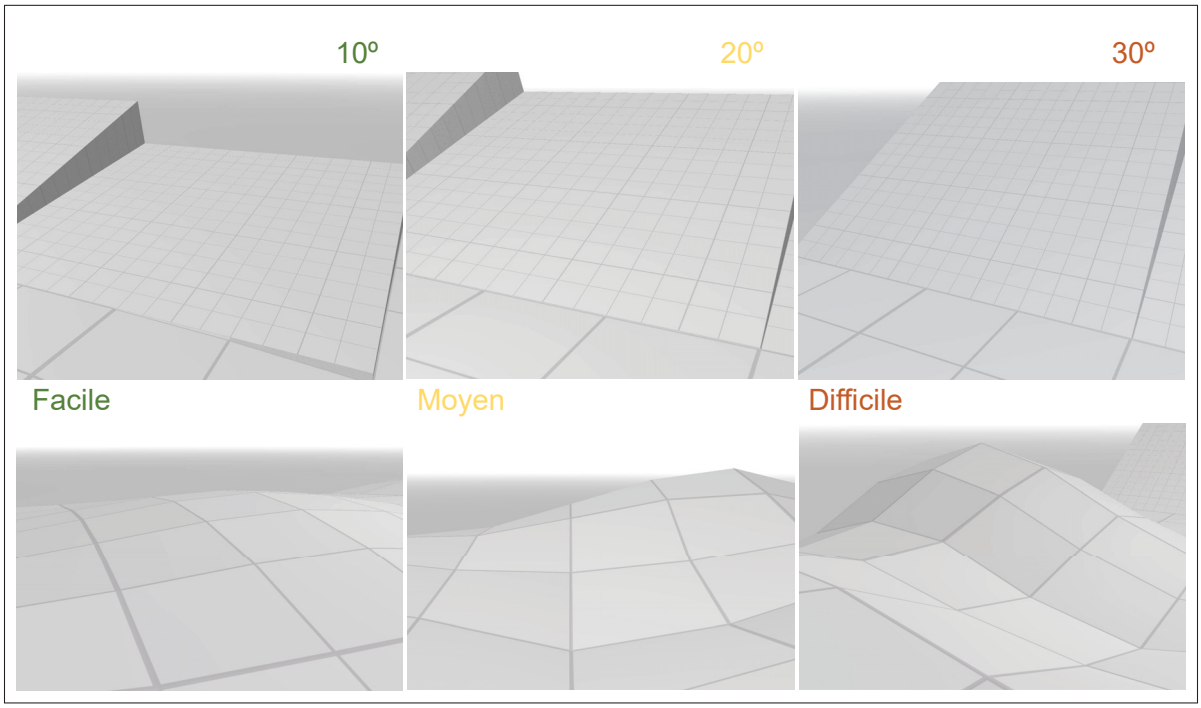

Figure 2.3 Le terrain est composé de trois niveaux de difficulté (facile, moyen, difficile). Il est composé de tuiles de 1.0 m × 1.0 m qui sont manuellement perturbées pour créer un terrain irrégulier.

Le terrain possède un emplacement plat et trois niveaux de difficulté progressive (facile, moyen, difficile). La politique est entraînée suivant l'apprentissage par curriculum qui consiste à introduire graduellement des nouveaux niveaux de difficulté tout le long de l'entraînement, plus de détails sont disponibles dans la Section 2.8. À cause de quelques limitations techniques, le terrain est pré-généré une seule fois et reste inchangé pour toutes les expérimentations. Une fois la politique entraînée, elle est exécutée sur le même environnement. C'est un procédé qui est plus simple à mettre en place et ignore l'adaptation pour de nouveaux terrains complètement inconnus. Cependant, nous considérons que notre méthode reste applicable avec une génération de terrains aléatoires et cela fait partie de nos futurs travaux. Afin de s'adapter au terrain, le personnage perçoit un champ de hauteur d'une dimension de 1.0 m × 1.0 m avec une précision de 25 cm qui

s'assure de couvrir convenablement le corps du personnage quand il est au sol. Notre méthode est inspirée par les travaux de Peng *et al.* (2018a) qui démontrent une très bonne adaptation avec une grille de 3.0 m $\times$ 3.0 m. Cependant, ils traitent principalement des mouvements de locomotion qui nécessite une plus grande prédiction du terrain environnant, qui n'est pas forcément nécessaire pour des mouvements plus statiques. Contrairement à Peng *et al.* (2018a) qui traitent le champ de hauteur avec un réseau de convolution plus communément utilisé dans les tâches basées sur la vision, notre personnage perçoit chaque point individuellement comme une hauteur par rapport au sol. Cette manière à l'avantage d'être plus simple à mettre en place. Cependant, elle agrandit la taille du vecteur d'état qui peut allonger le temps d'apprentissage et réduire la performance due aux nombreuses requêtes de calcul de distance sur la géométrie. Un coefficient de frottement d'une valeur de 1 Coulomb est utilisé pour l'environnement qui est légèrement supérieur à ce que l'on peut s'attendre en réalité, mais que nous avons trouvé suffisant pour nos expérimentations. Cette valeur conserve un comportement proche de la réalité.

2.4 Représentation de la politique de contrôle

La politique de contrôle est constituée d'un vecteur d'état $\mathbf{s}_t$ et d'un vecteur d'action $\mathbf{a}_t$. Les états représentent la perception de l'environnement par l'agent tandis que les actions des capacités pour agir dans celui-ci. Dans cette section, nous décrivons en détail la composition des deux vecteurs.

2.4.1 Vecteur d'état

Le vecteur d'état final $\mathbf{s} \in \mathbb{R}^{352}$ peut-être exprimé :

$$\mathbf{s}_t = \begin{bmatrix} \mathbf{q}_t & \mathbf{p}_t & \boldsymbol{\omega}_t & \mathbf{v}_t & \mathbf{w}_t & \mathbf{h}_t & \mathbf{t}_t & \mathbf{y}_{t-1} \end{bmatrix}$$

Les états sont les informations que les politiques de contrôle reçoivent en entrée pour produire la meilleure action en sortie en fonction de chaque situation à chaque pas de temps de la simulation. Le sous-script t correspond au pas de temps courant de simulation alors que $t-1$ le pas de

temps précèdent. Les états sont divisés en deux blocs comprenant les états du personnage qui contient des informations propos de la posture, vitesse actuelle ou encore le champ de hauteur. Le deuxième bloc correspond aux informations à propos de la tâche à effectuer.

Les états du personnage contiennent la rotation des articulations local $\mathbf{q} \in \mathbb{R}^{76}$ utilisant les quaternions. La vitesse angulaire locale $\boldsymbol{\omega} \in \mathbb{R}^{57}$, la vitesse linéaire $\mathbf{v} \in \mathbb{R}^{57}$ et position relative de chaque articulation $\mathbf{p} \in \mathbb{R}^{57}$ par rapport aux hanches qui sont utilisées comme articulation racine. On retrouve également des informations relatives aux caractéristiques qui déterminent le style du mouvement, tels que $\mathbf{w} = \begin{bmatrix} \mathbf{w}_{\mathrm{h}} & \mathbf{w}_{\theta_x} & \mathbf{w}_{\theta_y} \end{bmatrix} \in \mathbb{R}^3$ contient la hauteur courante par rapport au sol $\mathbf{w}_{\mathrm{h}}$, ainsi que l'angle frontal $\mathbf{w}_{\theta_x}$ et latéral $\mathbf{w}_{\theta_y}$ du torse courant par rapport à la direction verticale du monde $\mathbf{z}^+$. Le champ de hauteur $\mathbf{h} \in \mathbb{R}^{25}$ qui est une grille de 5×5 échantillonnée autour du personnage espacé de 25 cm entre chaque point. Cette grille informe de la distance entre l'articulation racine et le sol et permet au personnage de s'adapter aux différents types de terrains. Le champ de hauteur est orienté en fonction de la direction du personnage. Cette direction suit la direction horizontale de l'articulation racine (les hanches). Cela permet d'être invariant à la rotation globale et de généraliser les positions apprises. Les actions filtrées qui sont utilisées au pas de temps précédent $\mathbf{y}_{t-1} = \begin{bmatrix} \mathbf{z_{t-1}} & \Delta\mathbf{q}_{t-1} \end{bmatrix} \in \mathbb{R}^{65}$ sont fournis à la politique. Cette information est nécessaire pour utiliser le filtrage afin de lisser le mouvement du personnage (plus de détails dans la section 2.4.2).

Les états de la tâche sont constitués des trois angles cible $\mathbf{t} = \begin{bmatrix} \mathbf{t}_{\mathrm{h}} & \mathbf{t}_{\theta_x} & \mathbf{t}_{\theta_z} \end{bmatrix} \in \mathbb{R}^{12}$ provenant des courbes de style. Ils fournissent une fenêtre de valeurs qui dérivent la hauteur, l'orientation frontale et latérale du torse respectivement. Les courbes sont fournies à des intervalles réguliers dans le futur à 0 ms, 100 ms, 200 ms et 300 ms afin d'informer de la pente et d'anticiper sur un mouvement. Les valeurs de $\mathbf{t}$ sont extraites des fonctions p_{h}, p_{θ_x}, p_{θ_z} qui définissent la hauteur cible, l'orientation frontale et latérale respectivement. Plus de détails à propos des caractéristiques des courbes de styles sont fournis à la Section 2.6.

2.4.2 Vecteur d'action

Les actions produites par la politique contiennent un paramètre latent $\mathbf{z}' \in \mathbb{R}^{16}$ et des décalages pour les articulations $\Delta\mathbf{q}' \in \mathbb{R}^{49}$, tel que

$$\mathbf{a} = \begin{bmatrix} \mathbf{z}'_t & \Delta\mathbf{q}'_t \end{bmatrix} .$$

Le filtrage d'action proposée par Bergamin *et al.* (2019) est utilisé afin de produire un lissage lors des changements des angles des articulations qui sont contrôlées par les axes servo PD. Nous avons spécifiquement observé que cela contribuait à améliorer le placement des effecteurs finaux sur des terrains difficiles. En effet, nous avons constaté que le bruit Gaussien qui est utilisé à forte amplitude pendant l'entraînement peut provoquer un glissement du personnage. Cela réduit la possibilité de trouver un positionnement stable sur le terrain. Le filtrage des actions aide tout particulièrement à atténuer ces problèmes et permet de trouver de bons positionnements sur des terrains difficiles. Les actions filtrées sont calculées par

$$\mathbf{y}_t = \eta\, \mathbf{a}_t + (1 - \eta)\, \mathbf{y}_{t-1} ,$$

où $\mathbf{a}_t$ est l'action filtrée au temps t de la politique, $\mathbf{y}_{t-1}$ est l'action filtrée à partir du pas de temps précèdent, et $\eta = 0.2$ est le coefficient de lissage. Le paramètre de pose latent $\mathbf{z}$ et les décalages $\Delta\mathbf{q}$ sont extraits à partir des actions filtrées, tels que

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{z}_t & \Delta\mathbf{q}_t \end{bmatrix} .$$

Les angles finaux des articulations pour les axes servo PD sont calculés comme

$$\bar{\mathbf{q}}_t = \textsc{Decodeur}\,(\mathbf{z}_t, \mathbf{c}_t) + \Delta\mathbf{q}_t ,$$

où le décodeur reconstruit une pose complète à partir de l'espace latent utilisant $\mathbf{z}$ et $\mathbf{c}$ avec un procédé qui est expliqué dans la Section 2.5.

Cependant, une variable auxiliaire supplémentaire à $\mathbf{z}$ est également nécessaire pour calculer la pose complète du personnage. Nous la définissions tels que

$$\mathbf{c}_t = \begin{bmatrix} p_{\text{h,t}} & p_{\theta_{x,t}} & p_{\theta_{z,t}} \end{bmatrix}.$$

Le vecteur $\mathbf{c} \in \mathbb{R}^3$ contient la hauteur, ainsi que les orientations du torse qui sont à accomplir au pas de temps courant t.

2.5 Mouvement naturels

Le mouvement humain est synergique, les articulations se déplaçant en coordination pour accomplir des tâches. Nous exploitons donc cette propriété pour apprendre une variété de poses naturelles pour les mouvements de notre personnage. Nous construisons une approche basée sur les travaux de Yin *et al.* (2021) avec des changements mineurs. La différence majeure est que nous conditionnons le modèle sur les caractéristiques suivies par notre contrôleur. Concrètement, un VAE conditionnelle (C-VAE) permet de régresser les poses qui sont également conditionnées par la hauteur cible et les angles du torse, qui sont calculées pour chaque pose dans la base de données de capture de mouvements. Nous avons constaté que l'utilisation d'une condition permet d'aider la politique à générer des poses plus adaptées à la cible hauteur et angles du torse. Le C-VAE a l'avantage de pouvoir encoder une grande dimension d'entrée contenant avec de nombreuses postures vers un espace de basses dimensions. Par la suite, le décodeur peut être utilisé avec un vecteur latent composé de $\mathbf{z} \in \mathbb{R}^{16}$ et $\mathbf{c} \in \mathbb{R}^3$ qui reconstruit une posture naturelle. La particularité des VAE sont qu'ils produisent une distribution avec une moyenne et une déviation standard. Cela implique que chaque valeur de $\mathbf{z}$ est distribuée autour d'une moyenne afin de reproduire une pose approximative. Le βeta-VAE proposé par Higgins *et al.* (2017) permet de découvrir automatiquement des variations indépendantes dans des données non supervisées. Un hyperparamètre β permet d'ajuster une contrainte d'entraînement appliqué au modèle qui limite la capacité d'une information dans l'espace latent et permet d'apprendre statistiquement des facteurs latents indépendants. Une valeur $\beta = 1$ revient à utiliser la version originale du VAE proposé par He, Gong, Marino, Mori & Lehrmann (2019). Dans nos essais,

nous avons remarqué quand utilisant une valeur de $\beta = 1 \times 10^{-5}$ identique aux travaux de Yin *et al.* (2021) a tendance à regrouper plus fortement les postures et donc limiter la variété de celle-ci. Une valeur de $\beta = 1 \times 10^{-6}$ nous a aidé à reconstruire des postures plus variées tout en gardant un regroupement suffisant dans l'espace latent et sans créer de l'instabilité en dispersant trop les facteurs latents indépendants. Le VAE est entraîné avec un taux d'apprentissage de 1×10^{-4}, le décodeur et encodeur sont modélisés par des réseaux de neurones connectés entièrement avec deux couches d'une taille de 256 unités et une fonction d'activation tanh. Le modèle est entraîné avec 100 épisodes et une taille de lot (batch size) de 128. La taille du vecteur latent $\mathbf{z}$ est déterminé avec la méthode d'analyse des composants principaux (PCA) qui est appliquée sur la base de données contenant toutes les postures. Pour cela, la taille de 16 composants est déterminée pour couvrir 85 % de la variété des postures de la base de données, qui est légèrement supérieure à la valeur de 13 utilisée par Yin *et al.* (2021). Dans la Figure 2.1, l'encodeur est entraîné hors ligne de façon supervisée et ensuite seul le décodeur est utilisé pendant l'entraînement de la politique et lors de son exécution pour observer les résultats. Celui-ci permet de reconstruire une pose qui est ensuite ajoutée aux décalages afin d'obtenir une pose naturelle $\bar{\mathbf{q}}$ pour le personnage.

Peng *et al.* (2021) propose une méthode alternative basée sur réseaux antagonistes génératifs (GAN), qui permet de créer des mouvements naturels de façon très impressionnante. Leur méthode peut être également appliquée à notre projet. Cependant, les GAN sont moins stables et plus complexes. De plus, elle nécessite d'initialiser la simulation en position et vitesse aléatoirement avec des poses de références provenant de la base de données à titre d'exploration. L'initialisation par référence a été introduite d'abord dans les travaux de Peng *et al.* (2018a). Une ablation démontre que sans cette composante, le personnage ne peut pas apprendre des mouvements très synergiques sans lui fournir des exemples pertinents. Dans nos travaux, l'initialisation par mouvement de référence est effectuée seulement pour des poses au sol et debout. Il est important que nos résultats ne soient pas influencés par les postures déjà contenues dans la base de données, mais plutôt de créer de nouveaux styles de façon naturelle. Le but de leur approche est de composer dans le temps une variété de compétences pour différents comportements, alors que notre travail se concentre sur la création de mouvements de lever

du sol en utilisant seulement un petit nombre de signaux contrôlables qui guident le style du mouvement. Nous avons trouvé qu'il est suffisant de restreindre simplement l'agent à explorer des actions dans un espace de pose naturel sans intégrer également les transitions entre les poses. De plus, comme les transitions ne font pas partie du modèle latent, la politique est capable de synthétiser de nouvelles transitions qui ne sont pas contenues dans la base de données de mouvement d'origine, ce qui peut être important pour l'application de création de courbes.

Les postures naturelles sont composées de plusieurs sessions de capture de mouvement provenant d'un sous-ensemble de la base de données LAFAN1 fournie par Harvey *et al.* (2020). Celle-ci inclut seulement des sessions de lever et de tomber au sol pour un total de 20 minutes / 36000 cadres d'animation. Les sessions de capture des mouvements étant effectuée avec un acteur toujours en mouvement, elles réduisent le nombre de cas où le personnage se lève du sol et contient des phases où le personnage coure ou marche. Nous estimons que 30 % du total des animations concernent les mouvements de lever du sol et en général les mouvements sont plutôt dynamiques.

2.6 Contrôle du style

Notre approche consiste à contrôler le style du mouvement de lever du sol sans imiter une trajectoire complète d'un mouvement existant dans la base de données. Une interface simple est utilisée pour générer trois courbes qui décrivent quelques caractéristiques de style désiré pour nos mouvements. Une courbe définit la distance verticale par rapport au sol et deux autres permettent de définir l'orientation du torse.

2.6.1 Représentation en courbe définit par morceau (spline)

Une B-spline cubique cardinale est utilisée pour représenter les trajectoires des trois caractéristiques suivies par notre contrôleur. Par expérimentation, nous avons constaté que neuf points de contrôle suffisent pour synthétiser une variété de mouvements de lever du sol. Les courbes peuvent être directement extraits d'un mouvement de référence présent dans la base de données

en sélectionnant manuellement un mouvement de lever du sol capturé dans la session de mocap. Alternativement, les courbes peuvent être créées à l'aide d'une interface d'édition.

Nous définissons une courbe de caractéristiques générales $p(s)$ comme une fonction qui retourne une trajectoire cible basée sur un paramètre s. Le paramètre $s \in [0, 1]$ représente l'écoulement du temps de façon normalisé tel que :

$$s = \min\left(\frac{t - t_0}{T}, 1\right), \tag{2.1}$$

ou, t_0 est le temps de départ du mouvement, T est la durée du mouvement de levée du sol et t le temps courant de la simulation.

2.6.2 Caractéristiques des mouvements

La hauteur du personnage est mesurée entre la distance verticale entre la base du coup et la hauteur moyenne du terrain provenant du champ de hauteur. L'orientation frontale du torse est calculée comme l'angle formé entre le vecteur frontal et la direction verticale globale du monde $\mathbf{z}^+$, le même calcule est effectué pour l'orientation latérale. Le produit scalaire est utilisé pour calculer les angles avec une valeur résultante comprise entre $[-1, 1]$. Les vecteurs d'orientation sont représentés dans la Figure 2.2.

Les positions de suivi de hauteur et les vecteurs ont été déterminés par essai-erreur avec dans le but de simplifier le processus de création tout en offrant un contrôle suffisant pour créer divers mouvements d'apparence naturels. Dans des essais préliminaires, nous avons testé de suivre le centre de masse comme point de suivi de hauteur, mais le personnage a tendance à tricher en levant les bras et les mains afin de faire monter son centre de masse. De plus, le vecteur latéral est un requis important pour reproduire des mouvements de roulis. Nous avons échoué à reproduire ce type de mouvement avec seulement l'utilisation du vecteur frontale.

2.6.3 Édition des courbes

Une interface simple pour visualiser les mouvements de lever du sol consiste à ajuster la hauteur et l'orientation du torse pour les neuf points-clés. Cet ajustement peut être fait sur un mouvement original ou pour créer un mouvement à partir de zéro. Pour éditer un mouvement, l'utilisateur ajuste la valeur d'une courbe avec les manipulateurs tel qu'illustré dans la Figure 2.4. Ensuite, l'utilisateur visualise le changement en direct tel qu'illustré dans la Figure 2.5. Finalement, il est aussi possible de simplement changer le temps T alloué pour se lever afin de créer une nouvelle variante d'un mouvement.

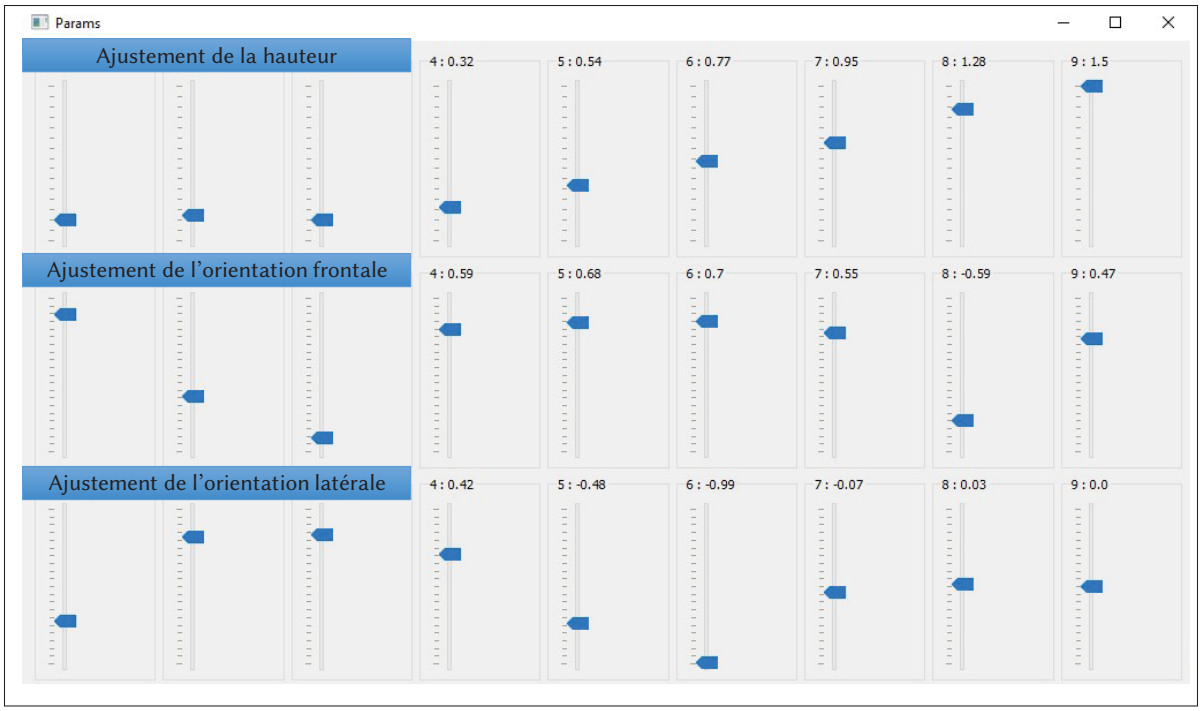

Figure 2.4 Interface d'édition des courbes

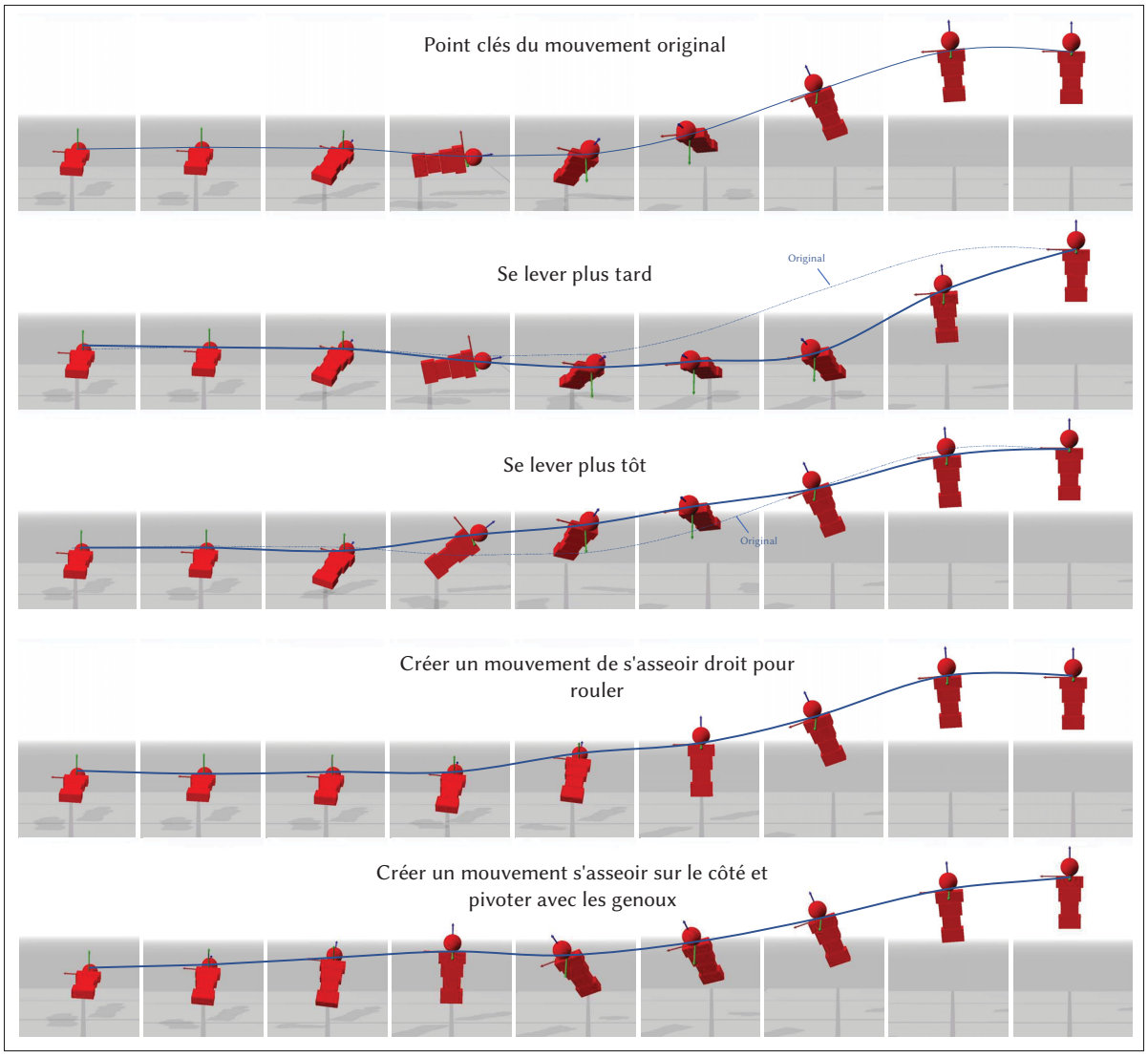

Figure 2.5 Visualisation d'une édition de courbes

2.7 Entraînement de la politique de contrôle

Cette section décrit les détails à propos de l'entraînent de notre politique de contrôle. Pour cela, nous passons en revue l'algorithme et les fonctions de récompense utilisées pour apprendre les mouvements de lever du sol.

2.7.1 Méthode d'apprentissage

Cette section décrie la méthode d'apprentissage utilisée pour entraîner les politiques de contrôle. Les politiques de contrôle sont entraînées utilisant l'apprentissage par renforcement profond qui consiste à apprendre par l'expérience en optimisant une fonction de récompense au fil du temps. Une politique de contrôle $\pi_\theta(a|s)$ est représentée par un réseau de neurones profonds qui déterminent les meilleures actions à prendre en fonction du vecteur d'état $\mathbf{s}$. À chaque pas de temps t, la politique de contrôle sélectionne une action $\mathbf{a}_t$ selon l'état actuel $\mathbf{s}_t$ produisant une transition dans la simulation qui génère un nouvel état $\mathbf{s}_{(t+1)}$.

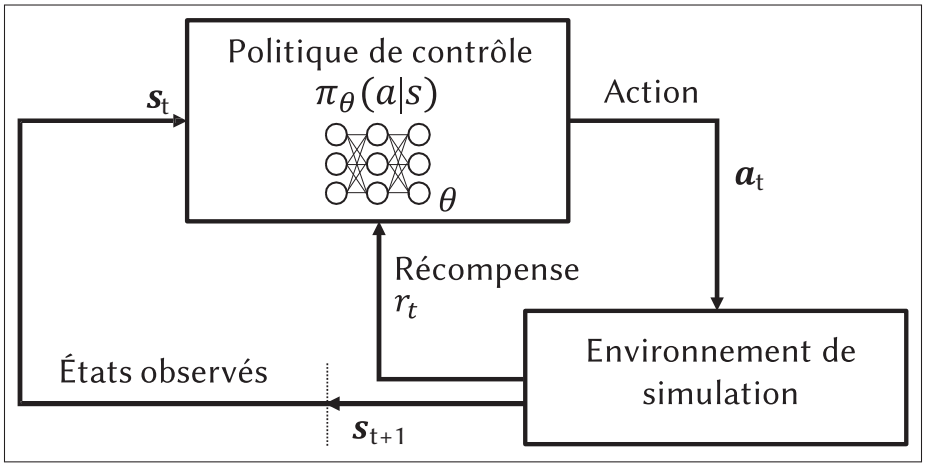

Figure 2.6 Entraînement de la politique de contrôle

Une fonction de récompense r_t est utilisée pour indiquer si la politique de contrôle applique une bonne action ou non. L'objectif d'une politique de contrôle est de trouver le paramètre optimal θ^* qui maximise la récompense :

$$J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=0}^{T} \gamma^t \, r_t \right],$$

où T est le nombre de pas de temps total, $\gamma \leq 1$ est le facteur de diminution et $p_\theta(\tau)$ représente la distribution sur toutes les trajectoires possibles $\tau = \{s_0, a_0, s_1, ..., a_{T-1}, s_T\}$ produites par la politique.

Proximal policy optimisation (PPO) proposé par Schulman, Wolski, Dhariwal, Radford & Klimov (2017) est l'algorithme d'apprentissage choisi pour apprendre notre politique de contrôle. Spécifiquement PPO-Clip avec *Generalized advantage estimator GAE(λ)* dont l'idée générale est de garder la nouvelle politique proche de l'ancienne après une mise à jour du réseau de neurones utilisant un paramètre de coupure ϵ. PPO est un algorithme composé d'un acteur et d'un critique. L'acteur est représenté par $\pi_\theta(a|s)$ et le critique par une fonction de valeur. La fonction de valeur est assurée par un autre réseau de neurones qui suit la politique lors de l'entraînement pour juger si les actions prises par l'acteur sont bonnes. La fonction de valeur $V(s_t)$ estime le retour moyen en démarrant à l'état s_t et suivant la politique pour les pas de temps suivant. L'avantage de prendre une action est alors estimé par $A_t = R_t - V(s_t)$ jugeant si la trajectoire prise améliore la politique. PPO a permis de résoudre des problèmes complexes dans les travaux récents tels que Peng *et al.* (2018a), Peng *et al.* (2021) , Won *et al.* (2021a) et Yin *et al.* (2021). L'avantage est que les implémentations de PPO sont parallélisables ce qui constitue un avantage lorsqu'un nombre très important d'échantillon est nécessaire pour apprendre les politiques. Dans notre cas, nous utilisons 16 agents qui sont entraînés dans leur propre instance de simulation et qui partagent la même politique de contrôle. Un nombre total de pas de temps est distribué entre chaque instance et la mise à jour de la politique de contrôle est effectuée une fois que chaque instance a complété leurs expériences individuelles.

Chaque politique de contrôle $\pi(\mathbf{a}|\mathbf{s}) = \mathcal{N}(\mu(s), \Sigma)$ est représenté par un réseau de neurones qui produit une distribution d'actions modélisées par un Gaussien. Pour cela, chaque état $\mathbf{s}_t$ correspond à une action moyenne $\mu(s)$ et une matrice diagonale Σ définit la déviation standard autour de la moyenne telle que $\Sigma = \text{diag}(\sigma_1, \sigma_2, ..)$. Le paramètre σ est contrôlé manuellement pendant l'entraînement afin d'explorer fortement lors du début de l'entraînement afin de créer une phase d'exploration. Ensuite, son amplitude est réduite vers une valeur finale à la fin de l'entraînement pendant la phase d'exploitation. Contrairement à la phase d'entraînement, les actions sont déterministes lors de l'utilisation de la politique finale.

Le vecteur d'état $\mathbf{s}_t$ est traité par un réseau de neurones de deux couches cachées compléments connectés de taille 256 suivis d'une couche linéaire de taille 65. La fonction d'action ReLU est

utilisée pour toutes les couches cachées. Les fonctions de valeur sont modélisées par des réseaux de neurones identiques séparés de leurs acteurs qui possèdent un même nombre et la taille de couches cachées.

2.7.2 Fonction de récompense générale

La fonction de récompense est un élément important en apprentissage par renforcement. Elle permet à l'algorithme d'apprentissage de s'améliorer en s'appuyant sur une seule valeur scalaire. Le principe est simple, mais beaucoup de comportements indésirables peuvent émerger si elle est mal conçue. Nous nous sommes inspirée de la fonction de récompense des travaux précèdent de Yin *et al.* (2021) tels que :

$$r_t = r_t^{\text{task}} r_t^{\text{natural}}, \tag{2.2}$$

où le terme r_t^{task} représente la tâche à accomplir et s'adapte à la tâche à effectuer. Pour s'assurer d'un mouvement naturel $r_t^{natural}$ est utilisé dans toutes les situations et s'exprime :

$$r_t^{\text{natural}} = 1 - \text{CLIP}\left(\left(\frac{\|\Delta \mathbf{q}\|_1}{c}\right)^2, 0, 1\right) \tag{2.3}$$

La valeur $c = 10$ radians est le décalage maximum autorisé pour toutes les articulations par rapport à une pose naturelle. Le vecteur de décalage $\mathbf{\Delta_q}$ est ajouté à la posture naturelle qui est décodée du C-VAE. De ce fait, $r_t^{natural}$ pénalise les déviations larges et encourage à rester proche des postures contenues dans la base de données.

2.7.3 Fonction de récompense de tâche de lever du sol

Dans cette section nous décrivons la fonction de récompense pour accomplir la tâche de lever du sol. La fonction de récompense peut être exprimée telle que :

$$r_t^{\text{task}} = r_t^{\text{h}}\, r_t^{\theta}\, r_t^{\text{v}}\, r_t^{\text{p}} \tag{2.4}$$

Le terme r_t^h encourage le personnage à suivre la courbe de hauteur, r_t^θ encourage le suivi de l'orientation des courbes du torse et r_t^v à réduire la vitesse des joints vers la fin du mouvement et r_t^p encourage l'agent à suivre une posture neutre définie par l'utilisateur. Nous avons trouvé plus efficace l'utilisation d'une multiplication des termes au lieu d'une forme additive. La multiplication supprime l'ajustement fastidieux des poids individuels qui sont requis dans la méthode additive. Dans la méthode utilisant des multiplications, chaque terme doit être non nul et doit atteindre une valeur raisonnable pour chaque terme. Dans les travaux de Won *et al.* (2020), ils ont démontré que l'utilisation d'une fonction de récompense basée sur des multiplications permet d'apprendre certains mouvements très dynamiques, là où ils ont échoué en suivant la méthode additive proposée par Peng *et al.* (2018a). Comme la hauteur et l'orientation du torse sont très dépendantes, une fonction de récompense utilisant des multiplications aide la politique à produire des mouvements quand la hauteur et l'orientation sont maximisées. La récompense de hauteur est exprimée telle que :

$$r_t^{\text{h}} = e^{-1.7\,|h_t - p_{\text{h}}(s)|}, \tag{2.5}$$

où $p_h(s)$ est la hauteur cible et h_t est la hauteur actuelle du personnage. Une récompense similaire est utilisée pour suivre l'orientation du torse telle que :

$$r_t^{\theta} = e^{-1.5\,\left(|\theta_{x,t} - p_{\theta_x}(s)| + |\theta_{z,t} - p_{\theta_z}(s)|\right)}, \tag{2.6}$$

avec $p_{\theta_x}(s)$ et $p_{\theta_z}(s)$ les angles qui sont à accomplir pour l'orientation frontale et latérale du torse respectif, $\theta_{x,t}$ et $\theta_{z,t}$ sont les angles actuels du torse. Une récompense qui vise à réduire la vitesse des articulations en fin de mouvement afin d'adopter une posture neutre et stable est exprimée telle que :

$$r_t^{\mathrm{v}} = (1 - \alpha_t) + \alpha_t e^{-0.05\|\omega_t\|}, \tag{2.7}$$

où ω_t représente la somme des vitesses angulaires actuelle utilisant la norme euclidienne pour chaque articulation j_i du personnage. Le terme r_t^{v} a plus d'importance en fin du mouvement, car nous ne limitons pas la vitesse du personnage pendant la phase de levée. Pour donner plus d'importance à ce terme en fin de mouvement la vitesse est modulée avec α_t calculé avec un terme exponentiel tel que :

$$\alpha_t = e^{-20\,(1-\phi)}\ , \tag{2.8}$$

avec $\phi = p_{\mathrm{h}}(s)/p_{\mathrm{h}}(s=1)$ effectuant une transition lisse vers la phase de maintien en calculant le ratio entre la hauteur cible actuelle $p_{\mathrm{h}}(s)$ et la hauteur finale $p_{\mathrm{h}}(s=1)$.

Finalement, l'agent est encouragé à suivre une posture neutre statique lors de la phase de maintien. Ce terme est nécessaire pour éviter des postures finales non désirables. Par exemple, pour éviter de se tenir debout avec les bras dans les airs. Ce terme s'exprime :

$$r_t^{\mathrm{p}} = (1 - \alpha_t) + \alpha_t\ e^{-0.1\left(\sum_j |\log(\hat{q}_t^j, q_t^j)|\right)}, \tag{2.9}$$

où $\hat{q}^j$ et q_t^j sont respectivement la posture relative à atteindre et la posture relative actuelle du personnage pour le $j$ième joint et la différence est calculée par une différence de quaternions.

2.7.4 Initialisation

L'état initial de démarrage au temps t_0 est importante, les travaux de Reda, Tao & van de Panne (2020) et Peng *et al.* (2018a) ont démontré que la façon dont la politique de contrôle explore ses résultats est fortement liée aux états initiaux. Dans nos travaux, la simulation débute l'agent au sol ou debout. Les postures au sol sont choisies à partir de la base de données dans le cas de l'entraînement sur un sol plat, car ils correspondent au même environnement de capture. De plus, utiliser des postures variées permet de rendre les politiques de mieux s'adapter, car elles sont entraînées sur des cas initiaux plus variés. Dans le cas des terrains irréguliers, la posture au sol est créée en faisant tomber le personnage sur le dos ou sur le ventre avec l'utilisation de petit torque afin de s'adapter à la forme du terrain. Pour le terrain plat, la posture neutre $\hat{q}^j$ fournie par l'utilisateur est utilisée pour initialiser la simulation, ceci aide l'agent à trouver la posture neutre final lors de la phase de maintien. Pour les terrains irréguliers nous utilisons également la posture neutre, mais adapté au terrain. Pour cela, les jambes et les pieds sont ajustés de manière à avoir les deux pieds touchants et alignés sur le sol. La posture debout est nécessaire pour explorer proche de l'état final à atteindre. Elle permet de donner un bon départ à la politique pour tenter de se stabiliser avec cette posture. Comme nous fournissons des courbes pour l'orientation du corps et la hauteur, il y a des cas où il n'est pas possible d'utiliser certaines d'entre elles en fonction de la configuration initiale du personnage. Pour cela, seules les postures initiales ayant une différence d'orientation initiale du torse de 30 $^\circ$ peuvent être choisies pour accomplir la tâche de lever du sol.

2.7.5 Fin hâtive d'un épisode

Le personnage doit suivre de proche les courbes qui définissent la hauteur et l'orientation du torse. Pour cela, un épisode est terminé quand le personnage dévie de sa trajectoire. Il a deux conditions à considérer : (i) la récompense de tâche est inférieure à 0.1, (ii) la tête du personnage touche le sol. L'une des deux conditions doit être active au moins 300 ms pour que l'épisode soit terminé de façon hâtive. Sans l'utilisation de la fin hâtive, nous avons rencontré des problèmes où l'agent n'accomplit pas le style de mouvement souhaité, cette fonctionnalité est importante

lors la phase d'entraînement initiale sur le terrain plat afin de mieux guider l'apprentissage du personnage.

2.8 Apprentissage par curriculum

L'apprentissage par curriculum consiste à augmenter la difficulté du problème progressivement pendant l'apprentissage au lieu d'essayer d'apprendre directement des comportements difficiles que le personnage ne pourra jamais réussir à accomplir. Cette méthode est inspirée de l'apprentissage humain et il a permis de résoudre des problèmes complexes dans les travaux de Xie *et al.* (2020) et de Lee, Lee, Lee & Lee (2021b). Dans notre cas, le plus bas niveau de difficulté est représenté par un mouvement de lever du sol sur un terrain plat et le dernier sur des terrains très inclinés ou avec de nombreuses crevasses, là où il est difficile de se lever et de trouver une position de maintien stable et naturelle pour un humain. La Figure 2.3 illustre les quartes niveaux de difficulté utilisée pour l'apprentissage. Le terrain reste inchangé pendant l'entraînement et pendant l'exécution finale de notre politique de contrôle. Lors de l'entraînement, la politique π le personnage démarre sur le terrain plat. Pour passer au niveau suivant, il faut que le personnage réussisse à accomplir la tâche 200 fois. Pour compter un succès le personnage doit effectuer un mouvement complet de levée à partir d'une position allonger au sol et se maintenir debout pendant trois secondes. Après 200 réussites par niveau, le compteur est remis à zéro puis, une nouvelle zone de difficulté est ajoutée à la liste des zones où le personnage apprend la tâche. Pour éviter que l'agent oubli comment effectuer les tâches sur les zones déjà apprises, le personnage visite équitablement toutes les positions déjà visitées pendant l'apprentissage. Plus l'entraînement progresse et plus de positions introduites. Afin d'homogénéiser l'entraînement, les positions sont visitées de maniée cyclique, mais elles sont mélangées dans chaque instance de simulation pour visiter le plus de cas différents sur un épisode d'entraînement.

2.9 Entraînement des courbes

Une collection de courbes est utilisée pour entraîner l'agent à se lever du sol de manière diversifiée. Les courbes sont extraites à partir des cadres d'animations des mouvements de

référence, mais proviennent aussi des courbes éditées manuellement. Spécifiquement, treize courbes pour les mouvements de référence et cinq courbes éditées manuellement pour créer différents styles de mouvement de lever du sol. L'agent est entraîné également sur des courbes identiques, mais en augmentant et diminuant de 50 % la vitesse par rapport au temps initial pour effectuer le mouvement dans la base de données. Au total, 28 courbes différentes sont utilisées pour entraîner l'agent.

2.10 Détails d'implémentation

Le cadriciel utilise PyTorch et l'implémentation de PPO-Clip est mis à disposition gratuitement par Raffin *et al.* (2021). Le personnage physique et l'environnement sont créés avec l'engin physique Vortex Studio CM Labs Simulations (2019). La simulation physique utilise un pas de temps de 60 Hz et une nouvelle action de la politique est appelée toutes les 30 Hz. Cette fréquence est une valeur standard qui est utilisée dans les travaux récents de Peng *et al.* (2021) ou Bergamin *et al.* (2019). De plus, une étude approfondie a été effectuée par Peng & van de Panne (2017) pour comparer les différents taux d'appel les politiques de contrôle. Les expérimentations sont effectuées sur un ordinateur avec un système d'exploitation Windows avec une carte graphique NVIDIA GeForce GTX 1080 Ti et possède un processeur Intel i9 de 8 cœurs. La carte graphique n'est pas utilisée pour mettre à jour les politiques de contrôle. Chaque politique est entraînée en parallèle utilisant 16 simulations œuvrant dans des processus indépendants, mais partagent la même politique de contrôle. Un entraînement complet dure environ 30 heures pour 100 millions d'appel à la politique de contrôle. Les hyperparamètres utilisés sont inspirés des travaux de Peng *et al.* (2018a) avec *generalized advantage estimate GAE(λ)* de 0.95. Contrairement à eux qui utilisent un facteur de diminution de γ de 0.95, nous utilisons une valeur de 0.99 qui est communément utilisé en apprentissage par renforcement et a fonctionné correctement pour nous. Un paramètre de clampage ϵ de 0.2, cependant un taux d'apprentissage progresse de 1×10^{-4} vers 2×10^{-5} sur 20 millions d'appels à la politique de contrôle, la déviation standard démarre à 0.6 et diminue vers 0.3 sur la même base de temps. Une taille de lot (mini-batch) de 256 est ajustée par expérimentation pour fournir les meilleurs résultats en

termes vitesse et stabilité d'apprentissage. L'acteur et le critique partage possède des réseaux de neurones séparés, les travaux de Andrychowicz *et al.* (2020) indiquent que la performance est globalement meilleure en séparant l'acteur et le critique en deux réseaux distincts. Quelques hyperparamètres supplémentaires relatif à l'implémentation de Raffin *et al.* (2021) sont ajustés. Tel que le coefficient d'entropie est 0, ce paramètre permet de diminuer l'entropie dans le système, mais n'est pas requis dans le cas où nous contrôlons manuellement la déviation standard de l'algorithme d'apprentissage. Puis finalement, un coefficient de 0.5 pour la proportion de perte (loss function) dans la fonction de valeur de l'algorithme. Cette valeur demeure l'ajustement par défaut qui est fourni par Raffin *et al.* (2021).

CHAPITRE 3

RÉSULTATS ET DISCUSSIONS

Dans ce chapitre, nous présentons quelques animations créées en utilisant notre cadriciel. Dans un premier temps nous présentons des exemples de mouvements de lever du sol en utilisant les courbes extraites des cadres d'animation de référence. Pour ce cas, nous démontrons que notre méthode est robuste à synthétiser des mouvements sur des terrains inclinés ou accidentés. Dans un deuxième temps, des résultats de mouvement de lever du sol qui sont produits à partir de l'édition de courbes. Enfin, nous effectuons une étude d'ablation sur l'efficacité de l'utilisation d'un C-VAE et de l'impact de l'utilisation de l'apprentissage par curriculum.

3.1 Mouvements de référence

Les Figures 3.1 et 3.2 illustres deux mouvements différents produits à partir des courbes extraites de la base de données de référence de LAFAN1. Pour chaque mouvement présenté dans cette figure, la même courbe est appliquée sur le terrain plat et incliné. Dans les exemples, on remarque que l'agent est capable de conserver un style tout en s'adaptant à la pente du terrain. Dans les Figure 3.1 et 3.2, la ligne du haut correspond au terrain plat, la ligne du milieu correspond au terrain incliné et la dernière correspond aux courbes utilisées pour accomplir les mouvements.

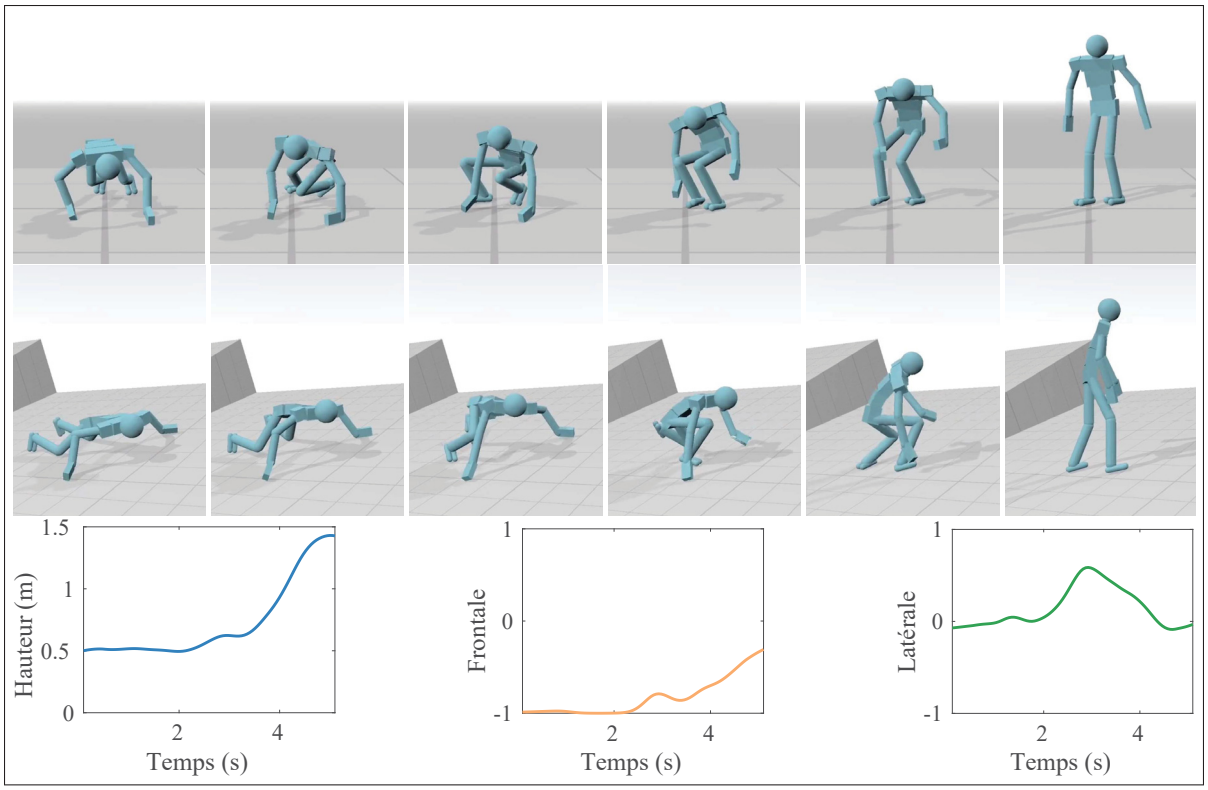

Figure 3.1 Exemples de mouvements sur terrain plat et incliné

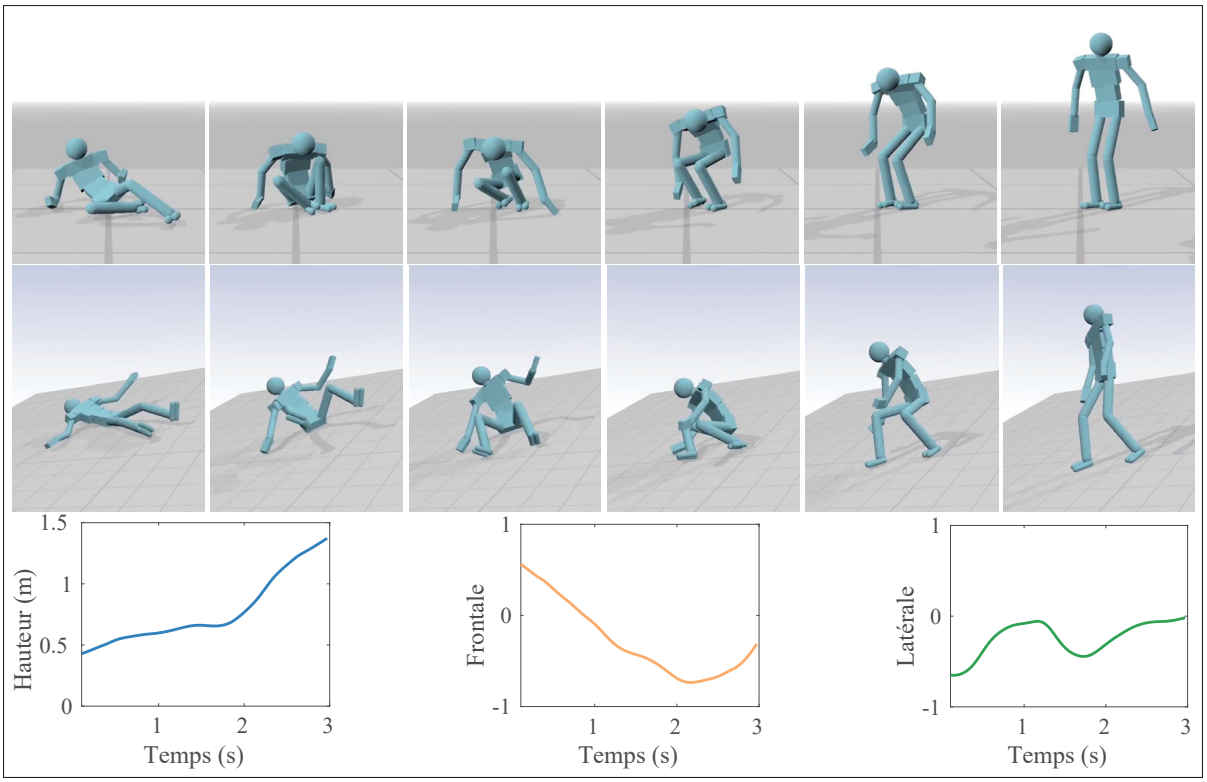

Figure 3.2 Exemples de mouvements sur terrain plat et incliné

La figure 3.3 illustres cinq mouvements produits par des courbes extraites de la base de données de référence de LAFAN1. En rouge, on distingue les cadres animations de référence au même temps de simulation que le personnage physique en bleu. On remarque que le style original du mouvement est conservé. L'orientation globale et la position des effecteurs finaux ne sont pas identiques, car nous ne considérons pas l'orientation globale comme une caractéristique de contrôle.

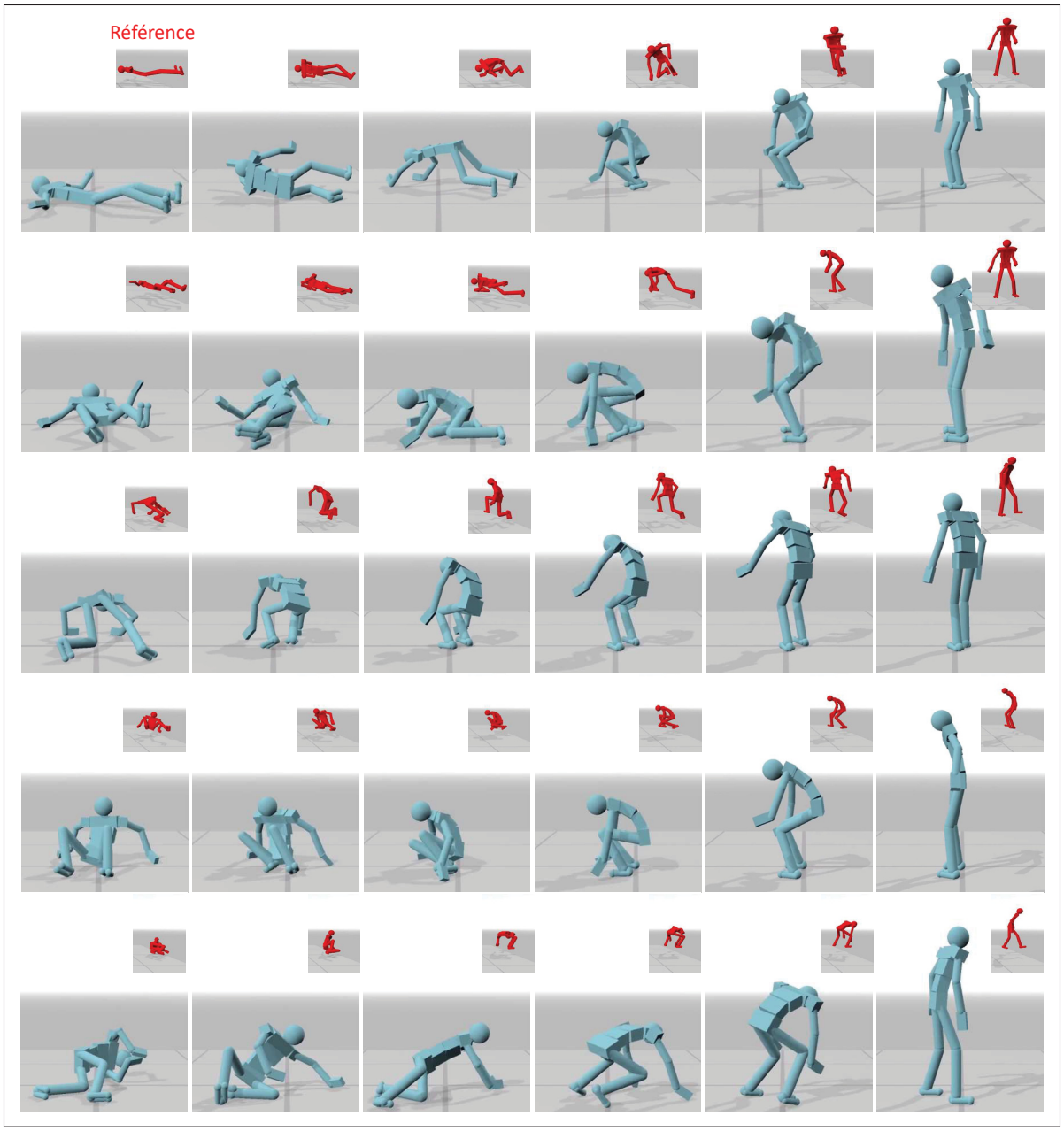

Figure 3.3 Exemples de mouvements sur terrain plat avec animation de référence. Les courbes utilisées sont extraites du mouvement de référence.

Deux exemples de mouvement sur des terrains accidentés sont présentés dans la Figure 3.4. Le personnage reproduit fidèlement les styles de mouvements extraient des clips de mouvement de

référence. De plus, il adapte sa posture à la forme du terrain en créant une flexion sur les genoux et orientant ses pieds pour les aligner avec le sol.

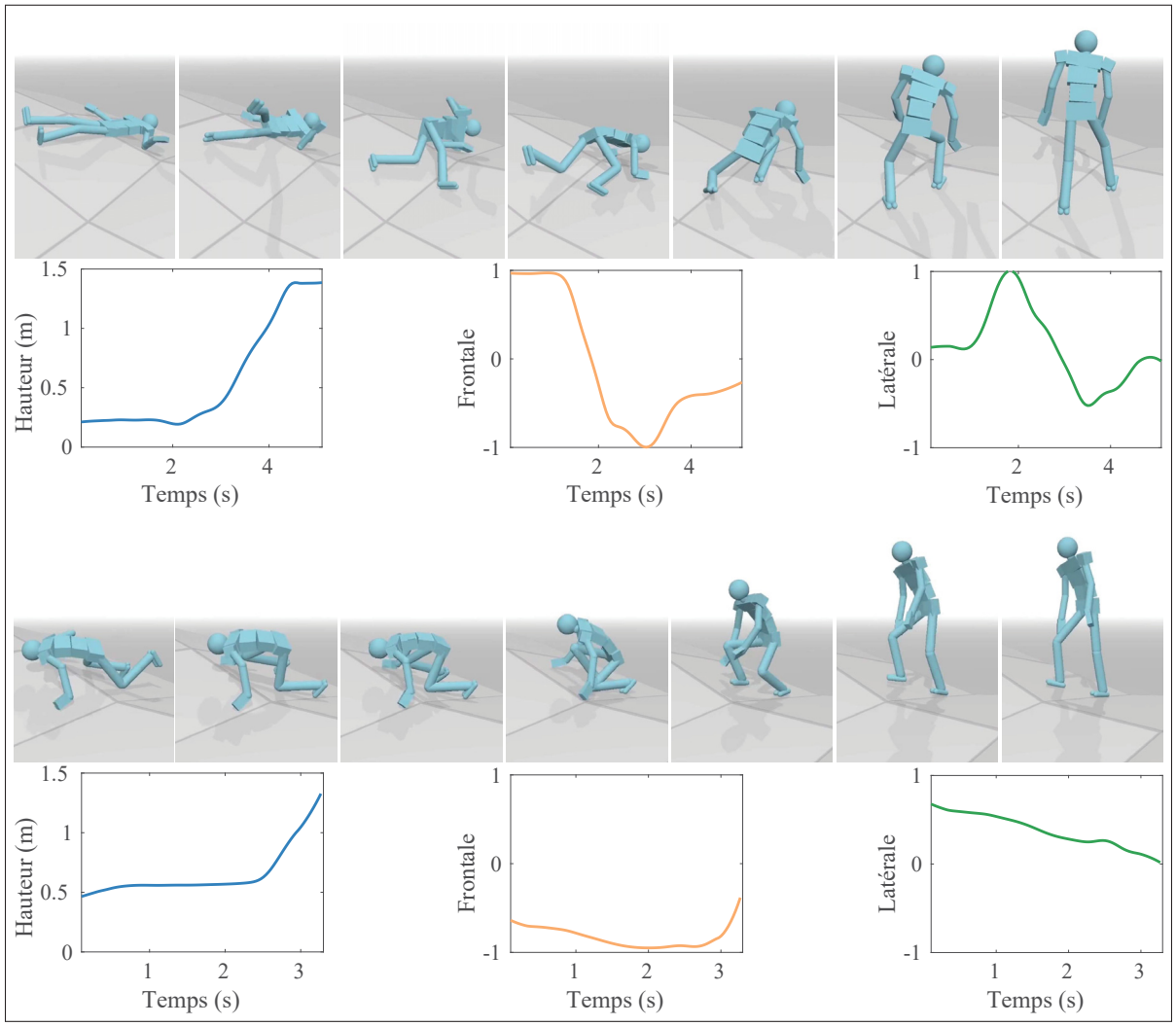

Figure 3.4 Exemples de mouvements sur terrain accidenté

3.2 Édition de courbes

Utilisant notre cadriciel, il est simple de modifier un mouvement existant en ajustement les courbes ou en changeant la durée du mouvement. Dans la Figure 3.5, un mouvement de référence est modifié pour simplement faire lever le personnage plus tôt dans le premier cas et plus tard

dans le deuxième. Cet exemple démontre qui est facile et rapide d'ajuster un style existant en ajoutant de petites variantes.

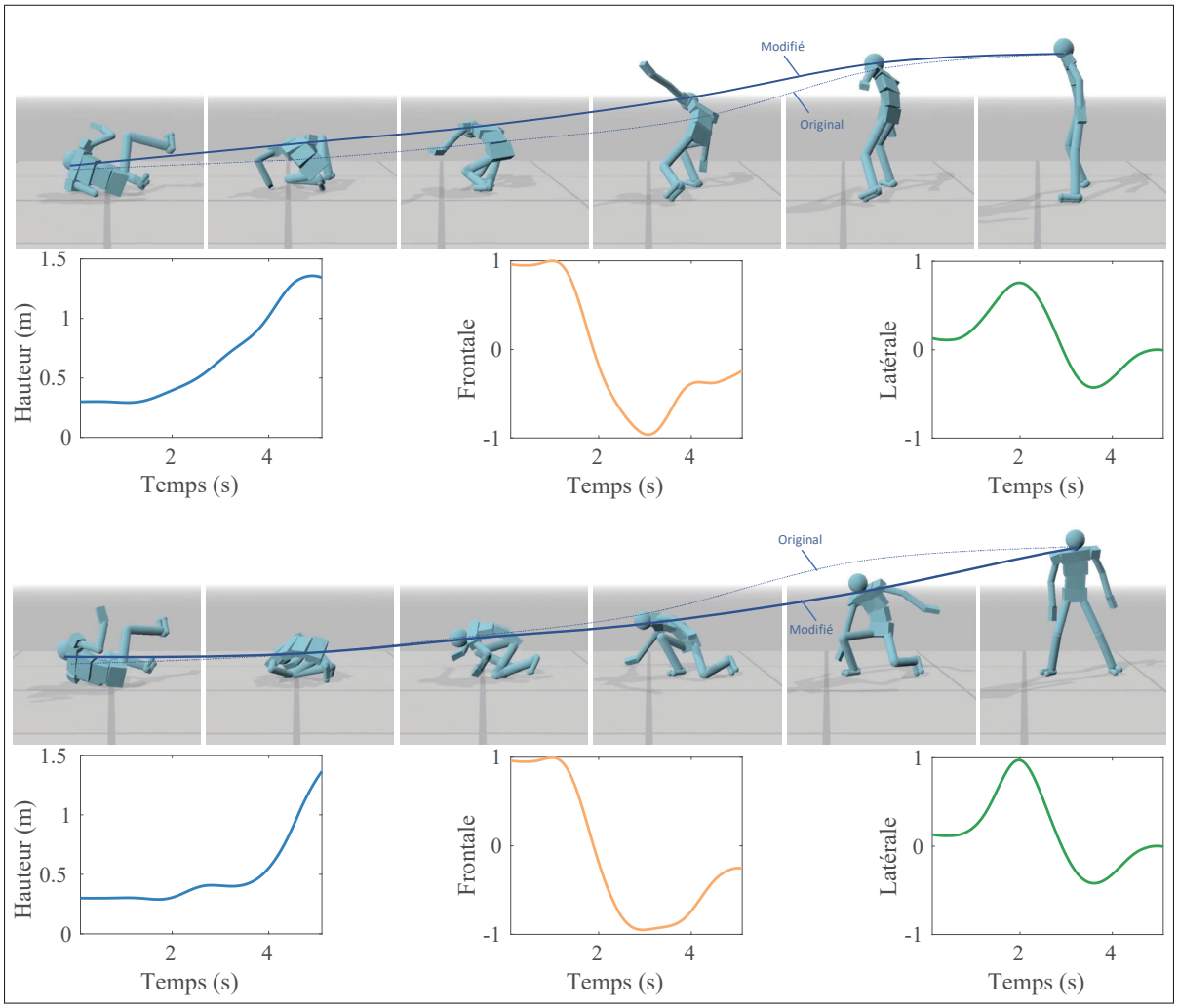

Figure 3.5 Modification d'une trajectoire pour se lever plus tôt (ligne du haut) et plus tard (ligne du bas).

De plus, il est possible de créer des mouvements à partir de zéro. Dans le but de mieux comprendre les caractéristiques des courbes et leur effet sur le style de mouvement de lever du sol. Nous avons classifié manuellement chacun des treize mouvements de références de la base de données de LAFAN1 selon les stratégies identifiées par Bohannon & Lusardi (2004) basées sur la moyenne et la déviation standard. Cette classification est illustrée dans la Figure 3.6. Nous distinguons à partir d'une posture couchée sur le dos et sur le ventre pour le style poussé vers le haut quadrupède, un mouvement pour s'asseoir droit pour rouler et s'asseoir sur le côté et pivoter avec les genoux. Ces tracés fournissent un guide utile pour la création de styles spécifiques de lever du sol.

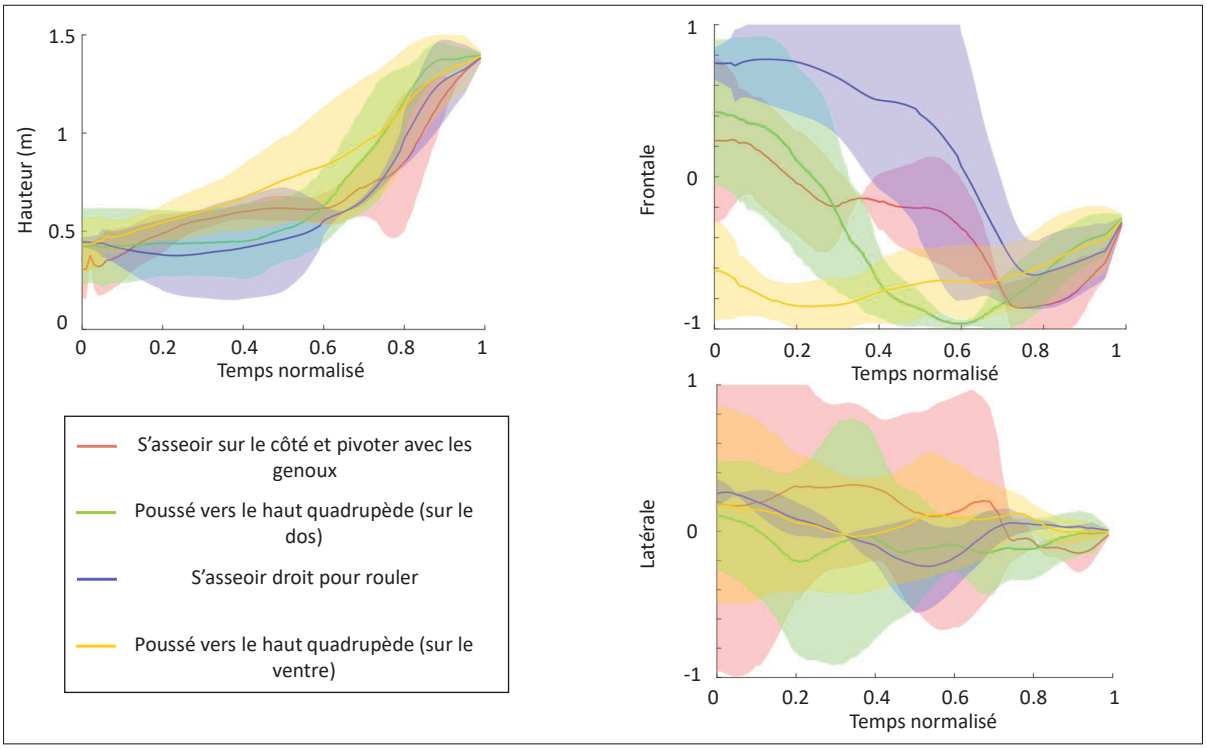

Figure 3.6 Classification des stratégies de mouvement de lever du sol

Avec l'aide des stratégies identifiées, nous avons édité deux stratégies telles qu'illustré dans les deux dernières lignes dans la Figure 2.5. Le premier mouvement en haut de la figure concerne la création d'un mouvement de s'asseoir droit pour rouler et le second pour s'asseoir sur le côté et pivoter avec les genoux.

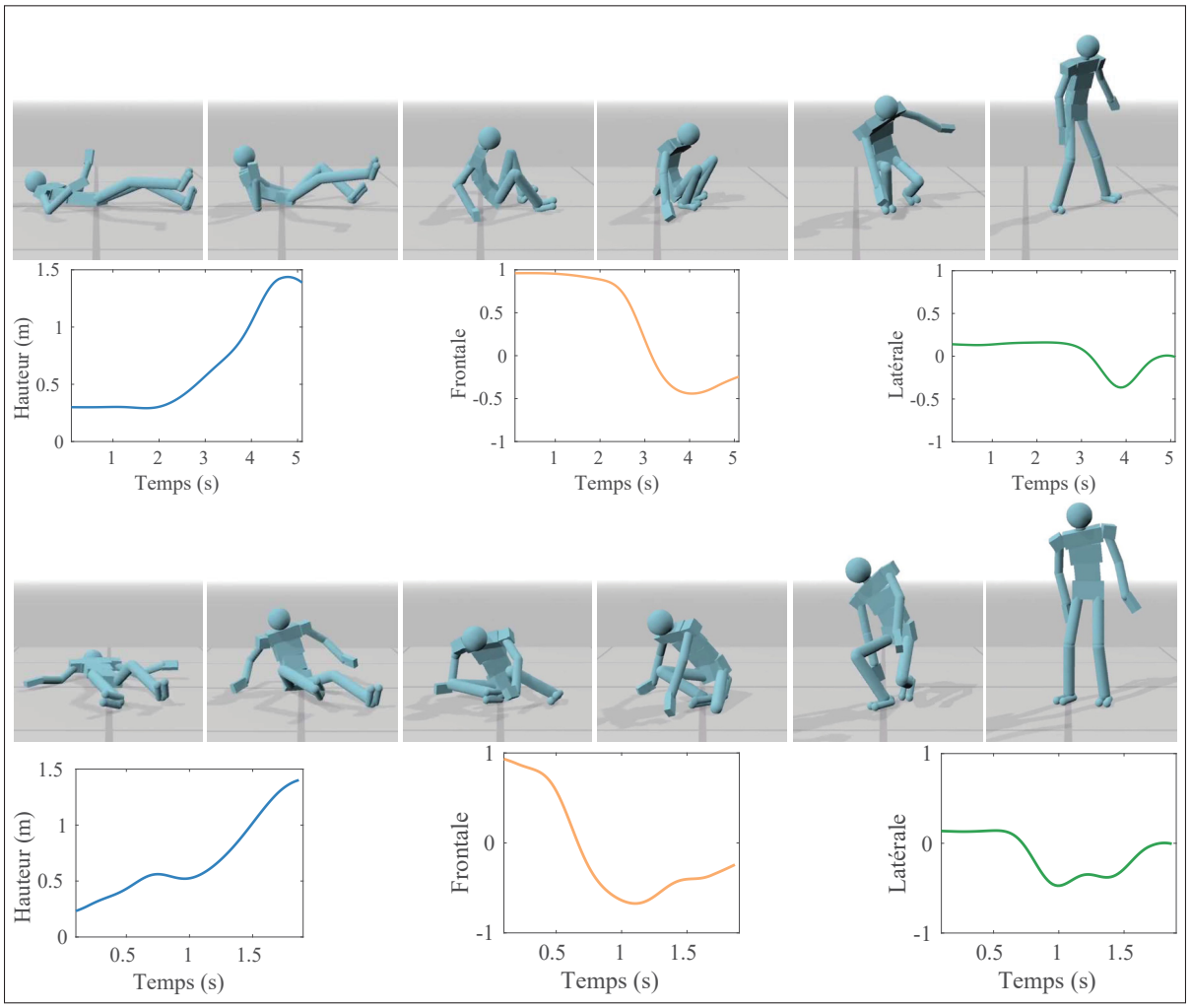

Figure 3.7 Trajectoires éditées manuellement pour créer deux stratégies distinctes

3.3 Étude d'ablation

Plusieurs études d'ablation ont été menés pour comparer l'efficacité de notre méthode d'entraînement. Les études ont été effectuées sur une même base de temps pour pouvoir les comparer. Pour cela, les expérimentations durent environ 30 heures pour 100 millions de pas de temps de simulation. Nous augmentons la difficulté tous les 20 millions de pas de temps de simulation, car certains cas peuvent faire échouer le passage vers les niveaux supérieurs de l'apprentissage par curriculum. Ceci correspond à peu après la fréquence de passage des niveaux dans les résultats sans ablation. Les chutes drastiques du taux de retour moyen qui sont observées dans la Figure 3.8 indiquent le passage sur des nouveaux terrains ayant une difficulté supplémentaire. Pour cela, l'agent échoue plus souvent que précédemment, car il n'a pas encore appris les nouvelles positions. Une valeur correspond à la somme cumulative des récompenses de tâche une fois ayant visité toutes les courbes et positions d'entraînement. Cette valeur est ensuite normalisée en divisant par le nombre de pas de temps effectués pour visiter tous les cas. Cette base de temps prend en considération le temps complet du mouvement de levée et de maintien pour toutes les courbes visitées. La fin hâtive des épisodes induit que chaque ablation ne complète pas en même temps toutes les courbes. En effet, une politique avec une performance moindre visitera plus de courbes sur la totalité de l'entraînement. Dans la Figure 3.8, chaque expérimentation partage la même base de temps totale, mais les politiques plus performantes comptabilisent moins de valeurs de taux de retour que les politiques ayant une performance plus faible.

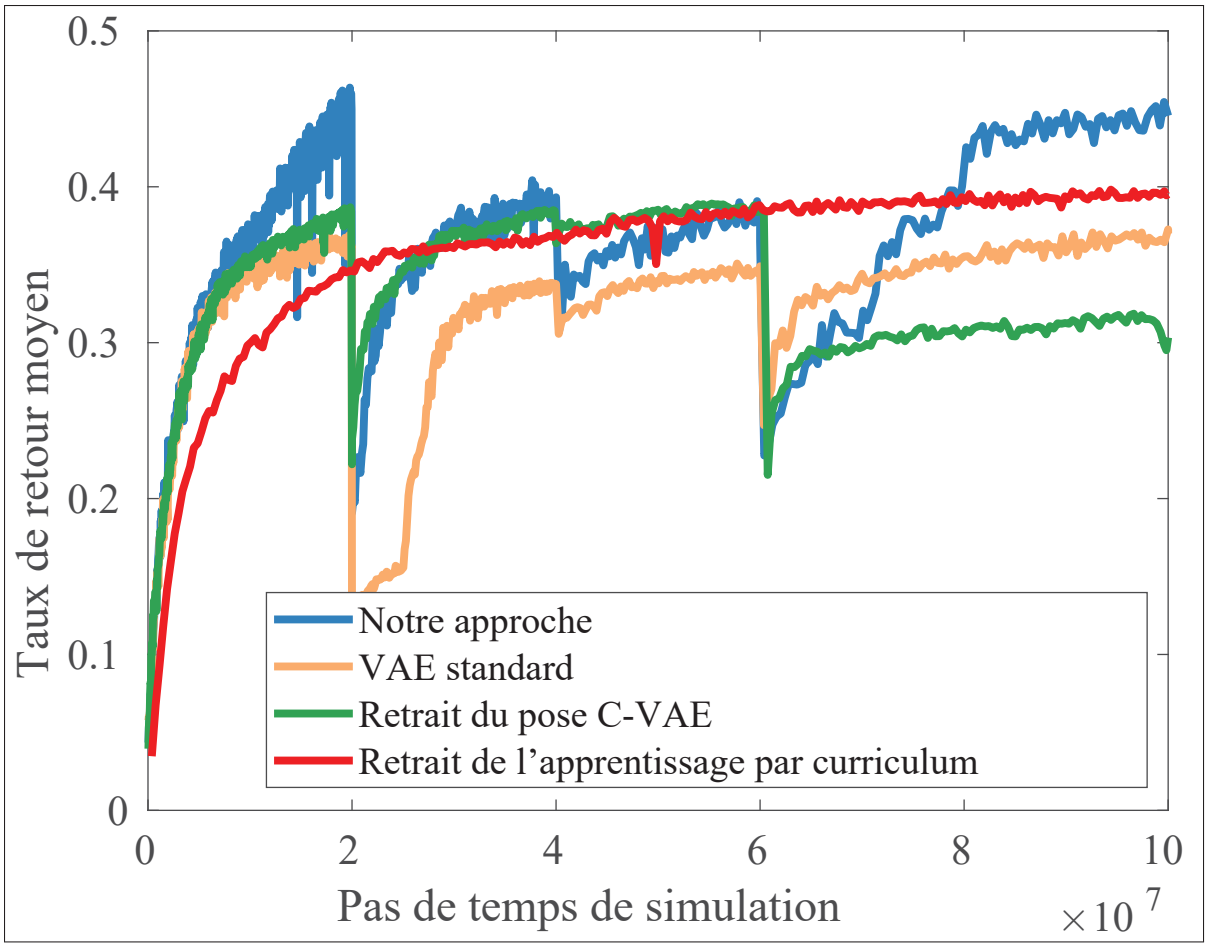

Figure 3.8 Courbe de progression pour l'étude d'ablation

3.3.1 VAE Standard

Nous avons entraîné le VAE standard proposé par Yin *et al.* (2021) au lieu de notre C-VAE. Le personnage est capable de suivre souvent la trajectoire et de produire différents styles de mouvements. Cependant, le personnage échoue souvent à atteindre la posture debout de façon stable. Cependant, nous dénotons quelques succès à effectuer un mouvement de complet tout en atteignant une position de maintien stable. L'agent démontre un bon progrès au début de l'expérimentation, le résultat se dégrade lorsque les terrains non plats sont introduits comme nous pouvons le voir dans la Figure 3.8. En fin d'expérimentation, l'agent arrive plus souvent à se lever, mais il peine à s'améliorer. Nous avons remarqué également que certains mouvements

peuvent être moins naturels dans leurs exécutions. Contrairement au C-VAE, les postures choisies peuvent être moins appropriées dans certaines phases du mouvement qui peuvent rendre leurs exécutions moins naturelles.

3.3.2 Retrait du pose C-VAE

La politique est entraînée pour produire directement les angles des joints plutôt que le vecteur latent $\mathbf{z}$ et les décalages $\mathbf{\Delta_q}$. De plus, le terme de naturalité r_t^{natural} est retiré de l'Équation 2.2 et seule la récompense de tâche r_t^{task} est utilisée. Nous observons que l'agent échoue complément à se lever du sol. De plus, il n'est pas capable d'atteindre la posture debout. Il essaie également d'atteindre la position debout en levant la tête vers le haut. Cette stratégie échoue par un manque de transitions plausibles dans les postures pour effectuer le mouvement de lever du sol. En fin d'expérimentation, on observe dans la Figure 3.8 que le taux de retour moyen reste loin des résultats atteints par notre méthode.

3.3.3 Retrait de l'apprentissage par curriculum

Dans cette étude, l'agent visite directement toutes les positions de difficultés sans être entraîné préalablement sur le terrain plat. Le personnage est capable parfois de se lever et rester debout. Cependant, il échoue fréquemment à atteindre la position debout pour certains mouvements. Plus spécifiquement, pour des mouvements qui démarrent étendu sur le ventre. L'agent a des difficultés à amener les pieds du personnage dans la zone de support. Ceci est une condition nécessaire pour se lever du sol. L'agent progresse d'une façon constante pendant l'entraînement. Il progresse toujours en fin d'entraînement et il est possible que l'agent atteigne des résultats similaires à notre méthode, mais avec un temps d'entraînement significativement plus long. Le taux de retour moyen final est inférieur au résultat obtenu par notre méthode. Nous expliquons ceci par l'échec à effectuer certains mouvements. Nous interprétons que le nombre de solutions possibles sont réduites par l'omission de l'entraînement préalable sur le terrain plat.

3.3.4 Discussion

Édition de courbes : La création des courbes implique d'abord d'analyser les données de mouvement réel et obtenir des informations sur les courbes pour des trajectoires typiques. Pour cela, l'analyse de classification présentée dans la Figure 3.6 est nécessaire pour ensuite reproduire certains mouvements qui appartiennent à la même classe. Pendant des expérimentations préliminaires, nous avons essayé de produire des mouvements seulement avec les courbes et sans outils. Cette stratégie implique un procédé par essai-erreur qui peut prendre plusieurs jours. Il existe un couplage entre les courbes qui peut être difficile à visualiser et peuvent échouer s'il n'existe pas un minimum de cohérence spatial. Au début, nous avons essayé seulement de suivre la trajectoire de la hauteur, qui est la plus facile à visualiser. Cependant, nous avons rapidement constaté qu'il est nécessaire d'avoir un suivi du torse pour créer de la variété dans nos mouvements. Dans un premier temps, nous avons utilisé uniquement le vecteur frontal, mais nous avons échoué à reproduire certains styles de mouvement, plus spécifiquement des mouvements de roulis. La direction globale du personnage est ignorée, cependant les vecteurs du torse fournissent un petit contrôle sur cette direction, mais n'assure pas celle-ci. Nous avons fait ce choix pour que les courbes fonctionnent quelque soi l'orientation initiale du personnage dans l'environnement. Si la direction globale et inclut, alors il est nécessaire d'ajouter des courbes de suivi pour cette caractéristique et des cas initiaux correspondants.

Orientation initiale du torse : Le contrôleur est moins robuste aux états initiaux ou l'orientation du torse ne correspond pas à celle des courbes. Nous avons constaté que l'agent réussit parfois à se corriger son orientation afin d'accomplir le mouvement à lever du sol. Cependant, le personnage peine souvent à accomplir le style souhaité.

Terrain escarpés : Se lever sur des terrains escarpés et complexes est un challenge important en animation basée sur des personnages physiques. Dans nos résultats, nous avons constaté que le personnage réussit très souvent dans un éventail large de type de terrain. Ceci inclut des terrains ayant des pentes jusqu'à 30° d'inclinaison.

Standard VAE contre le C-VAE : Dans des expérimentations préliminaires, le modèle du personnage n'avait aucune limitation de couples. Dans ce cas, le standard VAE est capable d'accomplir des mouvements de lever du sol. Cependant, les mouvements produits apparaissent très robotique et non naturel. Par la suite, les couples ont été réduit provoquant un échec d'apprentissage avec le VAE Standard. L'utilisation du C-VAE qui est conditionnée par la hauteur et l'orientation du torse a permis de réussir à accomplir les tâches, mais également de produire des mouvements ayant une meilleure qualité.

CONCLUSION

Nous proposons une méthode pour entraîner un contrôleur qui permet un personnage animé par la physique à se lever du sol sur des terrains variés adoptant des styles différents et contrôlables. L'adaptation est robuste par rapport à des terrains variés ce qui est accompli grâce à un apprentissage par curriculum. Le style du mouvement est contrôlable avec quelques caractéristiques clés qui sont facilement éditables par un utilisateur ou extraites d'un clip de mouvement de référence. Les mouvements produits sont naturels grâce à l'utilisation d'un espace latent représenté par un C-VAE. Celui-ci est entraîné sur une base de données larges de mouvements variés et conditionné par les signaux qui guident nos mouvements de lever du sol. Cette stratégie augmente la qualité des mouvements, mais aussi la rapidité d'entraînement du modèle.

Produire des mouvements naturels est un défi pour les personnages simulés par la physique. Un humain possède de nombreux degrés de liberté et se déplace en coordination. Notre approche utilise un espace latent de pose qui assure de produire des mouvements naturels. Pour cela, notre espace latent est entraîné avec une base de données larges de mouvements. N'ayant pas de métrique précise pour évaluer la qualité d'un mouvement, nous l'évaluons de façon visuelle. Nous estimons produire des mouvements naturels pour les phases de levée. Cependant, certaines postures finales n'adoptent pas une posture où les bras restent proches du corps. Par exemple, l'agent écarte les bras pour assurer une position de stabilité. Nous utilisons un terme qui incite le personnage à suivre une posture finale. Cependant, ce terme est ajusté de façon pour ne pas être trop restrictif afin ne pas contraindre trop les solutions sur les terrains irréguliers. Une piste de solution afin de régler ce problème serait de moduler la posture de référence afin qu'elle s'ajuste plus convenablement aux terrains non plats. D'autres solutions restent possibles en ajoutant des termes dans la fonction de récompense. Cependant, les méthodes récentes évitent la création de tâches exprimées par des fonctions de récompense complexes.

Nos mouvements produits suivent des configurations de contacts réalistes afin d'accomplir les mouvements de lever du sol. Cependant, il reste difficile de trouver de fortes variations de mouvement au niveau des effecteurs finaux (mains et pieds). L'agent a tendance à prendre la solution facile par rapport à la configuration initiale. Sachant que nous suivons un nombre limité de caractéristiques, nous avons tendance à perdre beaucoup de variation par rapport aux mouvements originaux. C'est un problème d'exploration dans l'espace latent. Une piste de solution serait d'ajouter un terme d'exploration afin de trouver des mouvements plus diversifiés. Par exemple dans les travaux de Peng *et al.* (2022), ils ajoutent un terme d'objectif afin de diversifier les comportements.

Nous suivons des courbes fournies par un utilisateur afin de créer facilement de nouveaux mouvements. La classe de mouvements que nous visons est difficile à créer avec les méthodes traditionnelles proposées par Peng *et al.* (2022, 2021, 2018a). Nous sommes également capables de suivre les caractéristiques d'un mouvement de référence. Cependant, l'omission de la direction globale peut rendre difficile l'édition des mouvements quand on souhaite contrôler celle-ci. Une piste de solution serait d'encoder une transformation composée d'une rotation et translation. La rotation peut décrire l'orientation du torse et le suivi de la hauteur avec la translation dans un cadre local. Par la suite, l'utilisateur pourrait obtenir un contrôle plus précis sur l'animation produite en sortie.

Le personnage physique s'adapte de manière robuste à des terrains non plats et ajuste le positionnement des contacts au sol tout en suivant convenablement les courbes de style. Cependant, le positionnement des effecteurs finaux reste un paramètre non contrôlable. Dans cette recherche, nous avons trouvé difficile de créer des mouvements très riches en contacts qui sont très séquentiels. Une piste de solution est d'introduire un graphe de contacts. Pour cela, l'agent pourrait parcourir un graphe d'état afin de reproduire une suite temporelle de configuration de contact. Par exemple, il nous a été difficile de créer des contacts francs avec

les genoux au sol sur des mouvements très lents. C'est un comportement plus communément adopté par les êtres humains afin de faire une transition plus naturelle du centre de masse.

Dans cette recherche, nous nous sommes intéressé à valider notre méthode sur des terrains irréguliers. Cependant, dans notre méthode la politique de contrôle reste dépendante du type de terrain. Nous avons constaté que l'agent généralise plus rapidement sur des terrains faciles, mais reste moins robuste sur des terrains très escarpés. Trouver la stabilité reste un défi majeur dans l'animation par la physique. Une solution serait d'apprendre des terrains plus diversifiés. Par exemple, une approche intéressante telle que proposée par Kumar, Fu, Pathak & Malik (2021) dans la robotique pourrait être une solution viable pour ce type de problème. Leurs travaux portent sur l'adaptation d'un robot quadrupède sur des terrains inconnus en temps réel. Pour cela, le module d'adaptation est entraîné sur les terrains rocheux, des surfaces glissantes, des surfaces qui se déforment, des environnements avec de la végétation, du béton, des cailloux et même des escaliers. Par la suite, le robot est capable de s'adapter de façon novatrice en une fraction de seconde sur des terrains inconnus.

Dans les travaux récents, récupérer après une perturbation large devient un pré-requis important. Dans nos travaux, nous nous sommes intéressé à produire des animations qui démarrent à partir d'une configuration statique du personnage au sol. Nous avons remarqué que notre méthode peut gérer convenablement la récupération après une chute, mais peu souvent échouer ou créer des transitions irréalistes n'ayant pas appris assez de configurations diverses au sol. Nous avons constaté dans des essais antérieurs que notre méthode fonctionne convenablement si des scénarios de chute sont inclus pendant l'entraînement. Par exemple, en ayant le personnage debout et en appliquant une force aléatoire sur la tête du personnage au début de l'épisode. Cependant, nous nous sommes concentré sûr d'autres point plus important dans notre recherche.

Dans l'avenir, il serait intéressant d'explorer plus profondément les caractéristiques qui peuvent guider d'autres types de mouvements. Les êtres humains sont capables d'accomplir un large d'éventails de mode ou de style pour une même tâche. Nous sommes intéressés tout particulièrement par l'accomplissement de mouvements riches en contacts. Notre outil interactif pourrait être utilisé pour créer des comportements novateurs pour ces cas particuliers.

Nous pensons également que notre méthode n'est pas seulement applicable aux mouvements de lever du sol. Sachant que nos caractéristiques suivent l'orientation du torse et une position de suivi de hauteur. Il pourrait être intéressant de valider que notre méthode fonctionne pour des mouvements de locomotion. D'ailleurs dans les travaux de Won *et al.* (2022), ils ont démontré que les contrôleurs appris avec des C-VAE s'adaptent convenablement sur des terrains irréguliers pour des mouvements de locomotion. Un contrôle de style pourrait être alors ajouté dans l'exécution de ce type de mouvement.

Notre méthode de suivi du contrôle de style n'est pas exclusif au C-VAE. Il pourrait être intéressant d'utiliser la méthode dernière de pointe de Peng *et al.* (2022) pour ajouter un contrôle de style sur les mouvements de lever du sol. Pour cela, la fonction de récompense proposé dans notre recherche peut être utilisée dans leurs cadriciel afin d'atteindre les mêmes objectifs. Sachant que leur méthode intègre un terme qui évite les effets d'effondrement de mode. Dans leurs résultats, le personnage récupère la position debout de façon très dynamique et peu réaliste. Notre méthode pourrait aider à créer des scénarios intéressants pour enrichir les résultats.

BIBLIOGRAPHIE

Adams, J. M. & Tyson, S. (2000). The Effectiveness of Physiotherapy to Enable an Elderly Person to Get up from the Floor : A single case study. *Physiotherapy*, 86(4), 185-189. doi : 10.1016/S0031-9406(05)60962-5.

Al Borno, M., de Lasa, M. & Hertzmann, A. (2013). Trajectory Optimization for Full-Body Movements with Complex Contacts. *IEEE Trans. on Visualization and Computer Graphics*, 19(8), 1405-1414. doi : 10.1109/TVCG.2012.325.

Andrychowicz, M., Raichuk, A., Stanczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., Gelly, S. & Bachem, O. (2020). What Matters In On-Policy Reinforcement Learning ? A Large-Scale Empirical Study. *CoRR*, abs/2006.05990. doi : 10.48550/arXiv.2006.05990.

Bergamin, K., Clavet, S., Holden, D. & Forbes, J. R. (2019). DReCon : Data-Driven Responsive Control of Physics-Based Characters. *ACM Trans. Graph.*, 38(6). doi : 10.1145/3355089.3356536.

Bohannon, R. W. & Lusardi, M. M. (2004). Getting up from the floor. Determinants and techniques among healthy older adults. *Physiotherapy Theory and Practice*, 20(4), 233-241. doi : 10.1080/09593980490887993.

Chemin, J. & Lee, J. (2018). A Physics-Based Juggling Simulation Using Reinforcement Learning. *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, (MIG '18). doi : 10.1145/3274247.3274516.

Chentanez, N., Müller, M., Macklin, M., Makoviychuk, V. & Jeschke, S. (2018). Physics-Based Motion Capture Imitation with Deep Reinforcement Learning. *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, (MIG '18). doi : 10.1145/3274247.3274506.

CM Labs Simulations. [[Online]]. (2019). Vortex Studio 2019c. Repéré à http://www.cm-labs.com/vortex-studio/.

Faloutsos, P., van de Panne, M. & Terzopoulos, D. (2001). Composable Controllers for Physics-Based Character Animation. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, (SIGGRAPH '01), 251–260. doi : 10.1145/383259.383287.

Fujiwara, K., Kanehiro, F., Kajita, S., Yokoi, K., Saito, H., Harada, K., Kaneko, K. & Hirukawa, H. (2003). The first human-size humanoid that can fall over safely and stand-up again. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2, 1920-1926 vol.2. doi : 10.1109/IROS.2003.1248925.

Fussell, L., Bergamin, K. & Holden, D. (2021). SuperTrack : Motion Tracking for Physically Simulated Characters Using Supervised Learning. *ACM Trans. Graph.*, 40(6). doi : 10.1145/3478513.3480527.

Geijtenbeek, T. & Pronost, N. (2012). Interactive Character Animation Using Simulated Physics : A State-of-the-Art Review. *Computer Graphics Forum*, 31(8), 2492–2515. doi : 10.1111/j.1467-8659.2012.03189.x.

Harvey, F. G., Yurick, M., Nowrouzezahrai, D. & Pal, C. (2020). Robust Motion In-Betweening. *ACM Trans. Graph.*, 39(4). doi : 10.1145/3386569.3392480.

He, J., Gong, Y., Marino, J., Mori, G. & Lehrmann, A. (2019). Variational Autoencoders with Jointly Optimized Latent Dependency Structure. *International Conference on Learning Representations*.

Heess, N., TB, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S. M. A., Riedmiller, M. A. & Silver, D. (2017). Emergence of Locomotion Behaviours in Rich Environments. *CoRR*, abs/1707.02286. doi : 10.48550/arXiv.1707.02286.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S. & Lerchner, A. (2017). beta-VAE : Learning basic visual concepts with a constrained variational framework. *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*.

Kanehiro, F., Fujiwara, K., Hirukawa, H., Nakaoka, S. & Morisawa, M. (2007). Getting up Motion Planning using Mahalanobis Distance. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2540-2545. doi : 10.1109/ROBOT.2007.363847.

Klima, D. W., Anderson, C., Samrah, D., Patel, D., Chui, K. & Newton, R. (2016). Standing from the floor in community-dwelling older adults. *Journal of aging and physical activity*, 24(2), 207–213. doi : 10.1123/japa.2015-0081.

Kumar, A., Fu, Z., Pathak, D. & Malik, J. (2021). RMA : Rapid Motor Adaptation for Legged Robots. doi : 10.48550/arXiv.2107.04034.

Lee, S., Lee, S., Lee, Y. & Lee, J. (2021a). Learning a Family of Motor Skills from a Single Motion Clip. *ACM Trans. Graph.*, 40(4). doi : 10.1145/3450626.3459774.

Lee, S., Lee, S., Lee, Y. & Lee, J. (2021b). Learning a Family of Motor Skills from a Single Motion Clip. *ACM Trans. Graph.*, 40(4). doi : 10.1145/3450626.3459774.

Lin, W.-C. & Huang, Y.-J. (2012). Animating rising up from various lying postures and environments. *The Visual Computer*, 28(4), 413-424. doi : 10.1007/s00371-011-0648-x.

Ling, H. Y., Zinno, F., Cheng, G. & Van De Panne, M. (2020). Character Controllers Using Motion VAEs. *ACM Trans. Graph.*, 39(4). doi : 10.1145/3386569.3392422.

Liu, L., Yin, K., van de Panne, M., Shao, T. & Xu, W. (2010). Sampling-Based Contact-Rich Motion Control. *ACM Trans. Graph*. doi : 10.1145/1833349.1778865.

Liu, L., Panne, M. V. D. & Yin, K. (2016). Guided Learning of Control Graphs for Physics-Based Characters. *ACM Trans. Graph.*, 35(3). doi : 10.1145/2893476.

Luo, Y., Xie, K., Andrews, S. & Kry, P. (2021). Catching and Throwing Control of a Physically Simulated Hand. *Motion, Interaction and Games*, (MIG '21). doi : 10.1145/3487983.3488300.

Merel, J., Tassa, Y., TB, D., Srinivasan, S., Lemmon, J., Wang, Z., Wayne, G. & Heess, N. (2017). Learning human behaviors from motion capture by adversarial imitation. doi : 10.48550/arXiv.1707.02201.

Merel, J., Hasenclever, L., Galashov, A., Ahuja, A., Pham, V., Wayne, G., Teh, Y. W. & Heess, N. (2019). Neural Probabilistic Motor Primitives for Humanoid Control. *International Conference on Learning Representations*. doi : 10.48550/arXiv.1811.11711.

Merel, J., Tunyasuvunakool, S., Ahuja, A., Tassa, Y., Hasenclever, L., Pham, V., Erez, T., Wayne, G. & Heess, N. (2020). Catch & Carry : Reusable Neural Controllers for Vision-Guided Whole-Body Tasks. *ACM Trans. Graph.*, 39(4). doi : 10.1145/3386569.3392474.

Mordatch, I., Todorov, E. & Popović, Z. (2012). Discovery of Complex Behaviors through Contact-Invariant Optimization. *ACM Trans. Graph.*, 31(4). doi : 10.1145/2185520.2185539.

Morimoto, J. & Doya, K. (1998). Reinforcement learning of dynamic motor sequence : learning to stand up. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems.*, 3, 1721-1726 vol.3. doi : 10.1109/IROS.1998.724846.

Mourot, L., Hoyet, L., Le Clerc, F., Schnitzler, F. & Hellier, P. (2021). A Survey on Deep Learning for Skeleton-Based Human Animation. *Computer Graphics Forum*. doi : https://doi.org/10.1111/cgf.14426.

Naderi, K., Babadi, A., Roohi, S. & Hamalainen, P. (2019, Aug). A reinforcement learning approach to synthesizing climbing movements. *IEEE Conference on Games 2019, CoG 2019*, pp. 1-7. doi : 10.1109/CIG.2019.8848127.

Park, S., Ryu, H., Lee, S., Lee, S. & Lee, J. (2019). Learning Predict-and-Simulate Policies from Unorganized Human Motion Data. *ACM Trans. Graph.*, 38(6). doi : 10.1145/3355089.3356501.

Peng, X. B. & van de Panne, M. (2017). Learning Locomotion Skills Using DeepRL : Does the Choice of Action Space Matter ? *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, (SCA '17). doi : 10.1145/3099564.3099567.

Peng, X. B., Abbeel, P., Levine, S. & van de Panne, M. (2018a). DeepMimic : Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills. *ACM Trans. Graph.*, 37(4). doi : 10.1145/3197517.3201311.

Peng, X. B., Kanazawa, A., Malik, J., Abbeel, P. & Levine, S. (2018b). SFV : Reinforcement Learning of Physical Skills from Videos. *ACM Trans. Graph.*, 37(6). doi : 10.1145/3272127.3275014.

Peng, X. B., Chang, M., Zhang, G., Abbeel, P. & Levine, S. (2019). MCP : Learning Composable Hierarchical Control with Multiplicative Compositional Policies. Dans Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. & Garnett, R. (Éds.), *Advances in Neural Information Processing Systems 32* (pp. 3681–3692). doi : 10.48550/arXiv.1905.09808.

Peng, X. B., Ma, Z., Abbeel, P., Levine, S. & Kanazawa, A. (2021). AMP : Adversarial Motion Priors for Stylized Physics-Based Character Control. *ACM Trans. Graph.*, 40(4). doi : 10.1145/3450626.3459670.

Peng, X. B., Guo, Y., Halper, L., Levine, S. & Fidler, S. (2022). ASE : Large-scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *ACM Trans. Graph.*, 41(4). doi : 10.1145/3528223.3530110.

Plagenhoef, S., Evans, F. G. & Abdelnour, T. (1983). Anatomical Data for Analyzing Human Motion. *Research Quarterly for Exercise and Sport*, 54(2), 169-178. doi : 10.1080/02701367.1983.10605290.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M. & Dormann, N. (2021). Stable-Baselines3 : Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268), 1-8. Repéré à http://jmlr.org/papers/v22/20-1364.html.

Reda, D., Tao, T. & van de Panne, M. (2020). Learning to Locomote : Understanding How Environment Design Matters for Deep Reinforcement Learning. *Motion, Interaction and Games*, (MIG '20). doi : 10.1145/3424636.3426907.

Schulman, J., Moritz, P., Levine, S., Jordan, M. & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. doi : 10.48550/arXiv.1506.02438.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. (2017). Proximal policy optimization algorithms. doi : 10.48550/arXiv.1707.06347.

Tao, T., Wilson, M., Gou, R. & Van de Panne, M. (2022). Learning to Get Up. *ACM Trans. Graph.* doi : 10.1145/3528233.3530697.

Tassa, Y., Erez, T. & Todorov, E. (2012). Synthesis and stabilization of complex behaviors through online trajectory optimization. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4906-4913. doi : 10.1109/IROS.2012.6386025.

Wang, Z., Merel, J. S., Reed, S. E., de Freitas, N., Wayne, G. & Heess, N. (2017). Robust imitation of diverse behaviors. *Advances in Neural Information Processing Systems*, 30. doi : 10.48550/arXiv.1707.02747.

Won, J., Gopinath, D. & Hodgins, J. (2020). A Scalable Approach to Control Diverse Behaviors for Physically Simulated Characters. *ACM Trans. Graph.*, 39(4). doi : 10.1145/3386569.3392381.

Won, J., Gopinath, D. & Hodgins, J. (2021a). Control Strategies for Physically Simulated Characters Performing Two-Player Competitive Sports. *ACM Trans. Graph.*, 40(4). doi : 10.1145/3450626.3459761.

Won, J., Gopinath, D. & Hodgins, J. (2021b). Control Strategies for Physically Simulated Characters Performing Two-Player Competitive Sports. *ACM Trans. Graph.*, 40(4). doi : 10.1145/3450626.3459761.

Won, J., Gopinath, D. & Hodgins, J. (2022). Physics-Based Character Controllers Using Conditional VAEs. *ACM Trans. Graph.*, 41(4). doi : 10.1145/3528223.3530067.

Xie, Z., Ling, H. Y., Kim, N. H. & van de Panne, M. (2020). ALLSTEPS : Curriculum-Driven Learning of Stepping Stone Skills. *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, (SCA '20). doi : 10.1111/cgf.14115.

Yin, Z., Yang, Z., Van De Panne, M. & Yin, K. (2021). Discovering Diverse Athletic Jumping Strategies. *ACM Trans. Graph.*, 40(4). doi : 10.1145/3450626.3459817.

Yuan, Y. & Kitani, K. (2020). Residual force control for agile human behavior imitation and extended motion synthesis. *Advances in Neural Information Processing Systems*, 33, 21763–21774. doi : 10.48550/arXiv.2006.07364.