

# Rapport de stage

---

## Contrôle du mouvement sur support mou ou liquide

Carensac Samuel

---

## Table des matières

1	Résumé .....	3
2	Introduction .....	3
3	État de l'art .....	3
3.1	Animation de personnages humains .....	3
3.2	Contrôle dans l'espace des joints .....	4
3.3	Maîtrise de l'équilibre au cours de la marche .....	6
3.4	Contrôle de la vitesse .....	8
3.5	déplacement en milieu aquatique .....	8
4	Vue globale du système .....	9
5	Modèle physique .....	10
5.1	Modélisation du personnage .....	10
5.2	Modélisation des fluides .....	10
6	Contrôle du mouvement .....	12
6.1	Combinateur d'état .....	12
6.2	Conservation de l'équilibre .....	14
6.3	Contrôle de vitesse .....	15
6.4	Optimisation hors ligne .....	18
7	Résultats .....	20
8	Conclusion .....	20

## 1 Résumé

A rédiger en dernier contexte 1 phrase présenter le vérou 1 phrase et comment il a été résolu 1/2 phrase résultat 1 phrase

## 2 Introduction

1/ 1.5 page (à faire à la fin) abstract en écrivant +, sans les résultats et sans présenter précisément la méthode Contexte Problématique Finir par le plan du document

## 3 État de l'art

### 3.1 Animation de personnages humains

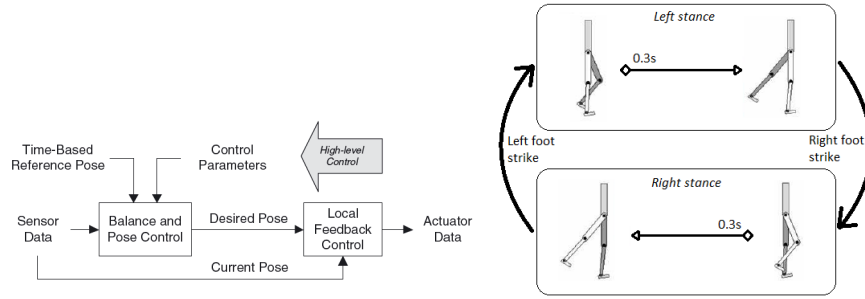
Dans le cadre de l'animation de personnages on dispose d'un squelette composé de corps solides reliés par des articulations permettant un certain nombre de degrés de libertés entre ces derniers. Il existe deux catégories de méthodes d'animation. La première, l'animation cinématique, décrit simplement les trajectoires des articulations au cours du temps. Bien que facile à réaliser cette méthode possède le défaut de rendre les interactions avec un environnement dynamique très compliquées. La seconde catégorie, l'animation basée physique, aborde le problème sous un tout autre angle. Cette fois –ci on manipulera le squelette à l'aide d'interactions physiques. On peut classer les interactions avec le squelette dans trois catégories [5] :

- L'application de moments au niveau des articulations. Le principe est d'appliquer des moments au niveau des articulations pour faire se déplacer les éléments relativement entre eux suivant les degrés de liberté.
- La seconde consiste à appliquer des forces directement sur les différents corps solides pour les animer. Cependant, un personnage réel ne possédant pas la possibilité d'effectuer ce genre d'action, les résultats obtenus peuvent sembler surnaturels. Cette technique a d'ailleurs été surnommée "hand-of-god" [14].
- La dernière méthode consiste à considérer des force comme la seconde méthode à la différence que celles-ci seront simulées à l'aide de moments sur les différentes articulations concernées. Ce type de forces est appelé forces virtuelles. On la retrouve notamment dans le cadre de contrôle de vitesse et d'équilibre [4].

Pour permettre l'animation du squelette de manière à obtenir les déplacements désirés, des systèmes haut niveau sont créés [5]. Ces systèmes prendront en compte des paramètres de haut niveau tels que la vitesse et la posture et les convertiront en interactions avec le squelette définies ci-dessus.

### 3.2 Contrôle dans l'espace des joints

Ce type de méthode se base sur un système similaire à l'animation cinématique. L'utilisateur spécifie une série de positions désirées représentant le mouvement et le système tentera de suivre ces positions dans la limite du possible. Les limites du contrôleur sont définies par des moments et des angles maximums aux articulations. La Figure 1 présente une vue d'ensemble généraliste de ce type de systèmes. Le bloc "balance and pose control" a pour rôle de convertir les paramètres choisis par l'utilisateur en une cible cinématique définissant les angles et le vitesses pour chaque articulation. Ces paramètres peuvent être de haut niveau (vitesse, hauteur des pas) ou de bas niveau (trajectoire des articulations). Cette cible cinématique permettra au bloc "local feedback control" de calculer le moment nécessaire pour chaque articulation. A ceci s'ajoute des contrôleurs de feedback qui affecteront des moments supplémentaires suivant des règles intégrées au modèle.



**FIGURE 1.** A gauche, exemple de contrôleur dans l'espace des joints [5]. A droite, exemple de machine à état pour la marche [19]

Bien qu'il existe plusieurs méthodes permettant de suivre les positions spécifiées (antagonist feedback [12], non-linear force field [11]), la méthode la plus commune est le "proportionnal-derivative control" (PD-control). Un PD-contrôleur calcule un torque pour chaque articulation linéairement proportionnel à la différence entre l'état actuel et l'état désiré. Il prend en compte la différence entre les angles mais aussi la différence entre les vitesses angulaires

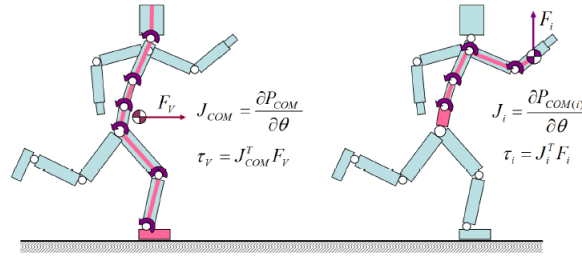
$$\tau = k_p(\theta_d - \theta) + k_v(\dot{\theta}_d - \dot{\theta})$$

Avec  $\tau$  le moment à l'articulation,  $\theta$  et  $\dot{\theta}_d$  l'angle et la vitesse désirée,  $\theta$  et  $\dot{\theta}$  l'angle et la vitesse actuelles et  $k_p$  et  $k_v$  des constantes. Parmi les systèmes basés sur un PD-contrôleur on trouve notamment le SIMBICON (SIMple Biped CONTroller) [19]. Le modèle est basé sur une machine à état fini. Chaque état est défini par une série de poses clés qui définiront les poses désirées au cours du mouvement. La transition d'un état à un autre peut s'effectuer après un

certain temps ou bien lors d'un nouveau contact entre un pied et le sol. La figure 1 illustre la composition d'un contrôleur SIMBICON pour la marche. Dans cet exemple chaque état est composé de deux poses clefs. La spécification des poses clefs possèdent quelques spécificités propres au SIMBICON. Les angles des articulations sont exprimés dans un repère local à l'exception de l'articulation entre le bassin et le dos et de celle de la hanche en phase de vol sont exprimées dans le repère du monde. Enfin la hanche d'appui ne possède pas de positions cibles. Le moment  $\tau_b$  à appliquer à la hanche d'appui est calculé de manière à ce que le moment total sur le pelvis corresponde au moment permettant d'obtenir l'orientation définie par l'utilisateur. Ce moment  $\tau$  désiré est déterminé par le PD-contrôleur en prenant pour angle cible la somme de l'angle définit dans la trajectoire du pelvis plus l'angle de heading (figure 3).

$$\tau_b = \tau - \tau_a - \tau_{torso}$$

Avec  $\tau_a$  le moment appliqué sur la hanche de balance et  $\tau_{torso}$  le moment appliqué à l'articulation entre le pelvis et le torse.



**FIGURE 2.** A gauche, jacobienne utilisée pour le contrôle de vitesse. A droite, jacobienne utilisée pour la compensation de gravité. [4]

Le défaut du PD-contrôleur est qu'il est nécessaire de connaître les bonnes valeurs pour les gains si l'on veut obtenir un résultat correct. Des gains trop faibles ne permettrait pas de suivre le mouvement définit. Des gains trop forts provoqueraient un mouvement saccadé et des possibles oscillations autour de la position désirée. On peut déterminer les bons paramètres par une série d'essais successifs mais cela ne permettrait pas d'obtenir un système robuste à des variations de géométrie dans le squelette. Pour pallier à ce problème le SIMBICON utilise un système de feedforward [19] sur les moments des articulations permettant ainsi d'obtenir une partie des moments nécessaire sans avoir à utiliser des gains élevés. Il existe d'autres méthodes permettant de calculer automatiquement une partie des moments. On trouve notamment un système de compensation de gravité [4]. Le principe est de calculer pour chaque partie du personnage une force virtuelle compensant la gravité. Donc pour chaque polygone nous appliquons une force  $F_{GC} = -mg$  au centre des masses, le signe négatif indiquant

une force vers le haut. Cette force est ensuite convertie en moment pour chaque articulation se trouvant dans la chaîne reliant le point d'application de la force et le pelvis  $\tau = J_i^T F$ .  $J_i^T$  est la transposée de la jacobienne de la chaîne reliant le pelvis et le point d'application de la force. La figure 2 illustre la chaîne affectée par la compensation de gravité d'un bras. La jacobienne d'une chaîne de  $k$  articulations contient les informations indiquant l'impact d'une rotation de chaque degré de liberté de la chaîne sur le point final de la chaîne. Chaque  $J(p)_i^T$  peut être calculé par le produit vectoriel entre l'axe de la rotation et le vecteur allant de l'articulation  $i$  et le point d'application

$$J(p)_i^T = \begin{bmatrix} \frac{\partial p_x}{\partial \alpha_i} & \frac{\partial p_y}{\partial \alpha_i} & \frac{\partial p_z}{\partial \alpha_i} \end{bmatrix} = (\alpha_i \times (p - p_i))^T$$

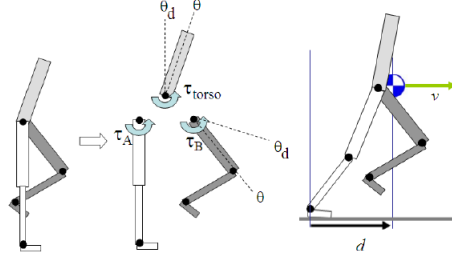
Donc la force à appliquer à chaque articulation  $i$  correspond à :

$$\tau = (\alpha_i \times (p - p_i))^T F$$

Cette méthode de compensation de gravité est appliquée à tous les membres du squelette excepté ceux composant la jambe d'appui.

### 3.3 Maîtrise de l'équilibre au cours de la marche

Définit tel quel le système ne permettrait pas de supporter des interactions avec l'environnement qui résulteraient en un déséquilibre. En effet il ne disposerait d'aucun système lui permettant de compenser les altérations de la position de personnage introduites par un sol non plat ou bien un obstacle.



**FIGURE 3.** A gauche, calcul des moments sur le pelvis [19]. A droite, représentation des variables  $d$  et  $v$  [19].

Pour pallier à ce problème le SIMBICON ajoute un système de balance feedback sur la hanche en phase de vol et le pied d'appui. Le principe est de modifier les angles cibles en fonction de la vitesse du centre des masses et de la distance entre celui-ci et le pied d'appui.

$$\theta_d = \theta_{d0} + c_d * d + c_v * v$$

Avec  $\theta_d$  l'angle qui sera utilisé par le PD-contrôleur,  $\theta_{d0}$  l'angle spécifié par les trajectoires,  $d$  la distance entre le centre des masses et le pied d'appui et  $v$  (voir figure 3) la vitesse de centre des masses. Les gains  $c_d$  et  $c_v$  sont déterminés pendant la phase d'optimisation hors-ligne. Il est important de signaler que la nécessité d'avoir des gains optimaux la rend très peu flexible.

Parmi les autres systèmes de placement intelligent du pied on trouve l'utilisation d'un modèle du pendule inversé (IPM) [7,4]. L'avantage majeur de l'IPM est qu'il est totalement autonome et ne nécessite pas de configuration manuelle offrant ainsi une grande robustesse aux variations de l'environnement. Cette version de l'IPM considère une longueur de jambe constante. Le principe de l'IPM est de partir du constat que la somme de l'énergie potentielle de pesanteur et de l'énergie cinétique reste constante peu importe la position du pendule  $\frac{1}{2}mv^2 + mgh = \frac{1}{2}mv'^2 + mgh'$  (figure ??). La supposition d'une vitesse nulle lors de la position d'équilibre (position verticale) nous permet de simplifier l'équation avec les variables suivantes  $v' = 0$  et  $h' = L = \sqrt{h^2 + d^2}$ . On peut ainsi déterminer la valeur de  $d$ .

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

Selon la vitesse du personnage il est possible que l'IPM nous donne des résultats impossible à atteindre c'est pourquoi la longueur des pas est limitée à  $d = 0.6L$  avec  $L$  la longueur de la jambe. Pour permettre d'avoir un résultat conservant une vitesse proche de la vitesse désirée  $V_d$  une altération des résultats de l'IPM est réalisé en utilisant la loi suivante  $\Delta = \alpha V_d$ . Ainsi une vitesse positive apporte une diminution de la longueur des pas provoquant une accélération et inversement une vitesse négative ralentit le personnage. La position effective du pied est calculée à l'aide d'une interpolation entre la position de départ du pied et la valeur retournée par l'IPM. La loi d'interpolation utilisée est une loi d'ordre 2 :

$$P_f = (1 - \phi^2) * pos_{init} + \phi^2 * p(x, y)$$

La hauteur du pied au cours du pas est définie par l'utilisateur à l'aide d'une trajectoire. Les angles désirés pour la hanche et le genou sont ensuite déterminés à l'aide de la cinématique inversé. Il reste un degré de liberté qui peut être contrôlé par l'utilisateur.

En plus de permettre l'équilibre l'IPM possède l'avantage de pouvoir complètement définir le mouvement de la marche humaine utilisant des paramètres plus haut niveau tel que la hauteur des pas et la vitesse désirée. Un avantage majeur d'un tel système est que l'on obtient une définition indépendante des caractéristiques physique du squelette offrant ainsi une grande flexibilité. Bien que très performant l'utilisation d'un IPM limite le déplacement à de la marche car il suppose un contact permanent avec le sol. De plus l'utilisation d'un IPM pour générer le mouvement complet de la jambe de balance limite grandement les styles de déplacement possibles.

### 3.4 Contrôle de la vitesse

Parmi les paramètres de haut niveau disponible à l'utilisateur on trouve le contrôle de la vitesse. Dans la version originale du SIMBICON la vitesse désirée est obtenue à l'aide d'une stratégie d'évolution. Le principe est de faire varier les positions cibles jusqu'à ce que l'on obtienne la vitesse voulue. Une amélioration du système [3] apporte la possibilité d'avoir plusieurs ensembles d'états. Le principe est d'avoir des ensembles généraux (i.e. marche avant, marche arrière, ...) et de faire des combinaisons de ces états pour obtenir des états intermédiaires. Cela permet non seulement d'avoir une vitesse variable mais aussi de pouvoir changer de vitesse sans avoir à redémarrer le contrôleur. Plus récemment, un système appliquant une force virtuelle horizontale pour accélérer ou ralentir le personnage a été présentée [4]. Le principe est d'utiliser un PD-contrôleur se basant sur la différence entre la vitesse actuelle du personnage et la vitesse désirée pour déterminer l'intensité de la force à appliquer. La force sera ensuite appliquée au centre des masses du personnage en calculant les torques nécessaires à l'aide de la méthode de la transposée jacobienne. La chaîne articulée considérée pour l'application de la force est la chaîne reliant la tête au pied d'appui (voir figure 2). Ce système permet d'avoir un contrôle fin de la vitesse. Cette méthode de contrôle a été utilisée de manière intensive dans le but de conserver l'équilibre dans le cadre de déplacement dans une position statique [6]. Cependant ce système ne permet pas de suivre correctement les vitesses si les poids du contrôleur sont incorrect. Particulièrement si l'on place le personnage dans un milieu entravant son déplacement il est nécessaire de retrouver les gains adaptés, si donné que ces gains existent. De plus l'application d'une force virtuelle reste limitée par les valeurs maximales des moments aux articulations. Ce qui veut dire que si le milieu gêne beaucoup le mouvement, cette stratégie de contrôle devient invalide.

### 3.5 déplacement en milieu aquatique

**Contrôle du mouvement en milieu aquatique** La simulation d'interactions entre un personnage et un milieu liquide a déjà été étudiée. La plupart des travaux placent le personnage en milieu aqueux [18,8,17,15]. On trouve également des approches utilisant un fluide pour simuler l'effet du vent sur le personnage [9]. Cependant ces contrôleurs sont utilisés pour simuler de la nage et non de la marche. Dans les articles discutant des simulations de vent le personnage est entièrement immergé dans le fluide [9]. De plus dans ce genre de situation l'effet de la poussée d'Archimède est ignoré. Dans le cadre de la simulation de l'eau de nombreuses approches se servent d'un modèle basé sur les équations de Navier Stokes [16] avec une représentation Eulérienne [15]. Le défaut est que ces systèmes ne permettent pas une simulation en temps réel. D'autres approches se contentent de modéliser l'eau à travers des forces extérieures calculées à l'aide d'équations simples telles que l'opposition du liquide, la poussée d'Archimède et la friction du liquide [18].



**Biomécanique de la marche en milieu aquatique** La marche en milieu aquatique a été le sujet de plusieurs études dans le milieu de la biomécanique [1,2,13,10]. Certains de ces travaux travaillent sur les différences provoquées par la présence de l'eau [1]. Cependant ces travaux utilisent des niveaux d'eau se situant au-dessus du milieu du torse (niveau du processus xiphoïde). C'est-à-dire que l'impact de l'eau sur le torse est toujours présent ce qui modifie fortement les résultats par rapport à notre situation. En effet le torse ayant une très grande surface le mouvement est extrêmement ralenti si il est plongé dans le fluide. De plus le volume du torse étant élevé la poussée d'Archimède prendrait une importance bien plus grande.

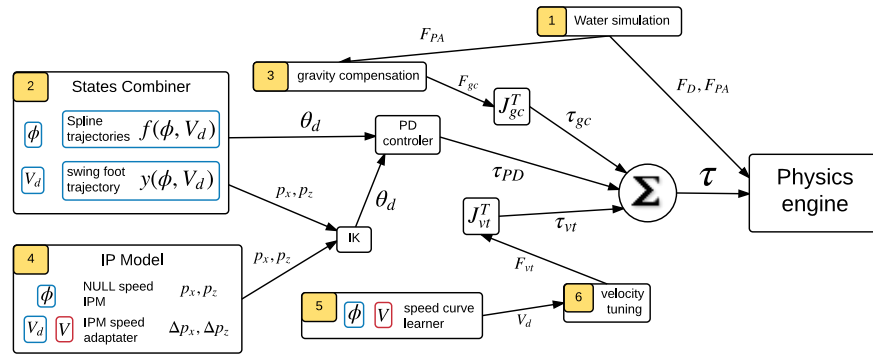


FIGURE 4. schéma récapitulatif du contrôleur

## 4 Vue globale du système

\*mettre une strip représentant la marche dans 0.25m d'eau\*

Le système de contrôle du mouvement est composé de 6 blocs principaux (figure 4). Voici une vision globale du fonctionnement du système.

Nous commençons par calculer les forces résultant de l'effet du fluide sur le personnage. Nous utilisons une modélisation simple nous permettant d'obtenir une exécution en temps réel.

Chaque articulation possède une trajectoire spécifiant les angles à suivre au cours du déplacement. Ces trajectoires sont déterminées à partir de points clés définis par l'utilisateur et les valeurs intermédiaires sont déterminées à l'aide de splines Catmull-Rom. Les trajectoires de la jambe de balance sont déterminées dynamiquement à partir de la trajectoire désirée pour le pied en phase de vol. La trajectoire du pied en phase de vol est soit spécifiée par l'utilisateur soit déterminée par les systèmes de conservations de l'équilibre lorsque c'est nécessaire. Le système de combinaison d'états permet à l'utilisateur de spécifier plusieurs trajectoires pour une articulation chacune étant associée à une vitesse. Suivant

la vitesse désirée, le système combinera plusieurs de ces trajectoires pour obtenir un résultat plus adapté pour chaque articulation. Ce système permet d’avoir un contrôle primaire sur la vitesse du personnage. Il est particulièrement utile pour les articulations possédant des comportements très différents suivant la vitesse voulue. Les angles lus depuis les trajectoires sont utilisés par le PD-contrôleur pour calculer les moments à appliquer au personnage au cours du déplacement.

Nous utilisons un système de compensation de gravité similaire à celui proposé par [4] modifié pour prendre en compte la présence du fluide. Ce système permet de calculer indépendamment une partie des moments nécessaire au mouvement ce qui nous autorise à utiliser des gains moins élevés dans le PD-contrôleur. L’utilisation de gains faibles est importante pour éviter d’obtenir des oscillations autour de la position désirée et obtenir un résultat s’adaptant plus facilement aux variations de l’environnement.

La conservation de l’équilibre est obtenue par un placement du pied intelligent déterminé par un modèle de pendule inversé (IPM). Pour permettre l’observation d’un plus grand nombre de styles de marche possible, nous n’utilisons l’IPM seulement lorsque le personnage est dans la phase de descente d’un pas ou lorsque nous détectons une perte de l’équilibre. Pour aider ce système l’articulation du pied d’appui possède un modèle de feedback corrigeant la position du personnage.

Le contrôle de la vitesse est obtenu à l’aide d’une force virtuelle similaire à celle utilisée par [4]. Ce système a été amélioré pour prendre en compte les variations de vitesse interne à un pas. Cette évolution nous permet d’obtenir un résultat correspondant plus à la spécification de l’utilisateur. Ce système est aidé par une altération des résultats de l’IPM permettant d’accélérer ou de ralentir le personnage. Ces deux systèmes sont contrôlés par du feedback et ne nécessitent aucun paramétrage de la part de l’utilisateur.

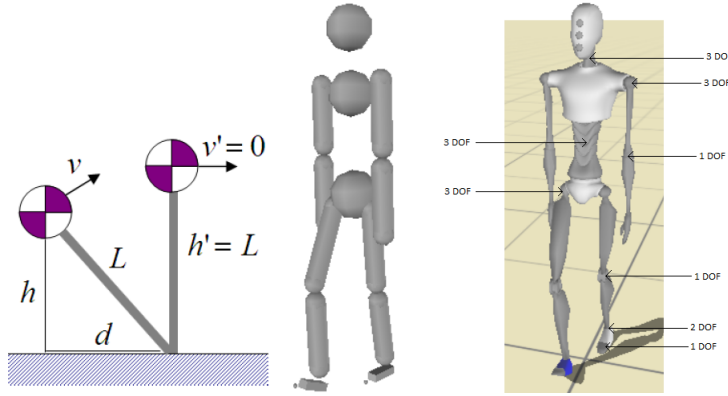
## 5 Modèle physique

### 5.1 Modélisation du personnage

Le personnage utilisé est composé de corps solides simples (visible figure 5). La tête, le torse, le pelvis et les orteils sont représentés par des sphères. Les bras et les jambes sont représentées par des capsules (cylindre avec des demi-sphères aux extrémités). Les pieds sont modélisés avec des prismes droits. Le squelette est composé des 28 degrés de liberté internes présentés dans la figure 5 plus 6 degrés de liberté externes (position et orientation). Il est intéressant de noter que le poids de chaque membre est modélisé par une valeur et un point d’application qui ne se trouve pas forcément au centre du polygone de collision.

### 5.2 Modélisation des fluides

Lorsque l’on déplace un corps solide à l’intérieur d’un fluide, ce dernier applique trois réactions principales sur le solide. La première est une force de résistance du fluide qui est produite par la nécessité de déplacer le fluide se trouvant



**FIGURE 5.** A gauche, modèle du pendule inversé [4]. Au centre, le modèle physique du personnage. A droite, degrés de liberté internes

sur la trajectoire du solide. La seconde est une force de friction entre le fluide et le solide. Ces deux réactions du fluide ont pour effet d'entraver le déplacement du solide. La dernière est la poussée d'Archimède, provenant des différences de pression sur la surface du solide, et a pour effet de repousser le solide vers la surface. Plutôt que de réaliser une modélisation réaliste du fluide (à l'aide des équation de Navier Stokes par exemple), nous allons baser notre modèle sur la définition d'une hauteur de liquide et les formules de base de dynamique des fluides pour représenter l'influence du liquide sur le personnage. Notre modèle ne considère pas les perturbations du fluide provoquées par le déplacement du personnage. Cette représentation a l'avantage d'être beaucoup moins couteuse en puissance de calcul ce qui nous permet de conserver une exécution en temps réel.

**Forces hydrodynamiques** La résistance du fluide est calculée à partir de la formule suivante :

$$F_D = \frac{1}{2} \rho v^2 A_n C_d$$

Avec  $C_d$  le coefficient de résistance,  $\rho$  la densité du fluide,  $A_n$  la surface en vue du membre du personnage et  $v$  la vitesse de ce membre. Les surfaces et les vitesses sont calculées à partir de la représentation physique du personnage. La vitesse n'étant pas constante en tout point d'un membre, nous utilisons un découpage en éléments finis de la surface des polygones pour effectuer nos calculs. La surface en vue  $A_n$  de chaque élément de surface est calculée en projetant la surface effective de l'élément dans un plan orthogonal à la vitesse  $v$ .

La viscosité du fluide est implémentée à travers un coefficient

$$F = F_D * \mu$$

Avec  $F$  la force finale à appliquer à chaque élément et  $\mu$  le coefficient de viscosité du fluide. Le coefficient de viscosité  $mu$  est obtenue en normalisant le viscosité réelle du fluide par celle de l'eau.

**Poussée d'Archimède** La poussée d'Archimède  $F_{PA}$  est calculée à l'aide de la formule suivante :

$$F_{PA} = -V_i \rho g$$

Le volume immergé  $V_i$  est calculé à l'aide du volume immergé des représentation physique des différents membres du personnage.

## 6 Contrôle du mouvement

Notre système se base sur la version du SIMBICON présentée par [4]. Les angles des trajectoires pour les chevilles, le pelvis, le dos et la tête sont spécifiées dans le repère du personnage. Ce repère est équivalent au repère du monde à la différence qu'il est orienté pour que l'axe z corresponde à la direction que face le personnage. Les angles des trajectoires pour les épaules, les coudes, le genou de la jambe d'appui et les orteils sont spécifié dans le repère local aux articulations.

Contrairement à [4] nous permettons la spécification manuelle du pied en phase de vol pour permettre l'observation d'un plus grand nombre de style de déplacement. Cette spécification manuelle est utilisée lors de la phase ascendante d'un pas si le personnage n'est pas en déséquilibre. Cette spécification s'effectue dans le repère local de la hanche en phase de vol.

### 6.1 Combinateur d'état

Suivant la vitesse désirée il est possible que les trajectoires nécessaires pour obtenir un résultat optimal soit radicalement différentes. Par exemple, étudions rapidement le cas de la cheville d'appui. Lorsque le personnage vers l'avant il est nécessaire que celui-ci pose le talon en premier. Une fois que le pied est à plat au sol il est nécessaire de conserver un déséquilibre entre les forces appliquées sur l'avant du pied et celles appliquées sur l'arrière du pied pour faire pencher le personnage vers l'avant. Au contraire, pour une marche sur place ou vers l'arrière, il sera nécessaire que celui-ci pose la pointe du pied en premier puis maintienne un déséquilibre des forces le faisant pencher vers l'arrière. Ces différences peuvent varier suivant le style de marche. C'est pourquoi nous utilisons une système permettant la définition de plusieurs trajectoires pour les articulations sur lesquelles nous observons ce type de phénomène. Notre système est similaire à celui présenté par [3]. Notre système se différencie par deux caractéristiques. Premièrement, contrairement à [3] nous n'utilisons pas notre système pour tenter de trouver un état stable en combinant plusieurs état de base. Notre système à pour but de créer une base plus adaptée au suivit de la vitesse désirée. Il ne cherche pas à obtenir une état stable. La seconde différence est que

[3] spécifie des état complets (i.e. marche avant, marche arrière, ....). Notre système se contente de capturer les différences ayant une influence importante sur la vitesse. C'est pourquoi nous nous contentons d'ajouter des trajectoires sur les articulations pour lesquelles nous observons des différences importante suivant la vitesse. De plus notre système a été conçu pour permettre d'utiliser la même trajectoire pour plusieurs vitesses. Ceci permet de spécifier des fourchettes de vitesse sur lesquelles nous voulons utiliser une trajectoire similaire car il n'y a pas de différences significatives. Nous avons déterminé que les trajectoires possédant des différences significatives sont celles des chevilles, du pelvis, du pied en phase de vol et celle de l'articulation entre le pelvis et le torse (dans une moindre mesure).

Nous allons maintenant expliquer le fonctionnement du combinateur d'état plus en détail. Pour chaque articulation ou système possédant plusieurs trajectoire, le combinateur d'état va chercher les deux trajectoires associées aux deux vitesses les plus proches de la vitesse désirée. Si la vitesse désirée est inférieure (resp. supérieure) à la vitesse la plus basse (resp. plus haute) spécifiée, le système utilisera directement la trajectoire correspondant à la vitesse la plus basse (resp. haute) comme si il n'y avait qu'une trajectoire définie. Si le système trouve bien deux trajectoire encadrant la vitesse désirée, il crée une nouvelle trajectoire contenant  $k$  points clefs répartis de manière uniforme. Lors de nos test nous avons déterminé que un nombre de 10 points clefs était suffisant. Le combinateur attribue ensuite des valeurs à chacun de ces points en utilisant une combinaisons des valeurs contenue dans les deux trajectoires considérées suivant une loi d'ordre 2 :

$$f(\phi) = f_1(\phi) * \frac{(V_d - V_1)^2}{(V_d - V_1)^2 + (V_d - V_2)^2} + f_2(\phi) * \frac{(V_d - V_2)^2}{(V_d - V_1)^2 + (V_d - V_2)^2}$$

Avec  $f$  la trajectoire finale,  $(f_1, V_1)$  et  $(f_2, V_2)$  les deux trajectoires les plus proches de la vitesse désirée.

**Compensateur de gravité** Notre système de compensation de gravité se base sur celui présenté par [4]. Nous avons modifié ce système pour permettre la prise en compte d'un milieu liquide. En effet, du fait de la poussée d'Archimède le poids à supporter par le personnage est moins grand lorsqu'il est plongé dans un fluide. De ce fait au lieu de compenser le poids réel de chaque membre du personnage, il faut compenser le poids effectif (poids observable dans le fluide). Notre modification consiste donc à calculer la force résultant de la poussée d'Archimède sur chaque membre du personnage et de soustraire celle-ci à la force virtuelle à appliquer. Il paraît intéressant de noter qu'il ne suffit pas simplement de diminuer l'intensité de la force à appliquer. En effet suivant la modélisation des membres du personnage, il est possible, et même fortement probable, que la densité soit variable à l'intérieur de ceux-ci. De ce fait le point d'application de la force compensant la gravité et celui de la poussée d'Archimède seront différent, ce qui nous oblige à déterminer le point d'application de la force finale  $F$  :

$$F = F_{gc} - F_{PA}$$

Avec  $F_{gc}$  la force calculée par le compensateur de gravité originel et  $F_{PA}$  la force causée par la poussée d'Archimède sur le membre concerné.

Si le milieu est assez dense il est fortement possible que les membres du personnage disposant d'une densité faible aient un poids plus faible que la force causée par la poussée d'Archimède. Dans ce cas le système génère des moments ayant pour effet de baisser les membres du personnage au lieu de les lever. Nous ne traitons pas ce cas comme un cas particulier. En effet, comme dit précédemment le but du compensateur de gravité est de calculer une partie des moments nécessaires pour pouvoir se rapprocher de la position désirée. Or si nous avons une poussée d'Archimède plus forte que le poids des membres du personnage, ceux-ci essaieront de flotter ce qui entraverait le déplacement du personnage. Il est donc intéressant de conserver le système présenté ci-dessus même dans ce cas particulier car il aidera tout de même le personnage à suivre les poses désirées

## 6.2 Conservation de l'équilibre

L'équilibre du personnage est obtenu par le placement intelligent du pied à l'aide d'un modèle de pendule inversé (IPM) et de fonctions de feedback sur certaines articulations (balance feedback). Les systèmes de contrôle de vitesse (section 6.3) permettent également un maintien de l'équilibre du fait qu'ils maintiennent le système autour d'une vitesse stable.

**Modèle du pendule inversé** L'utilisation d'un IPM permet d'avoir un système efficace de conservation d'équilibre ne nécessitant pas de paramétrage de l'utilisateur. Dans le cadre de notre projet nous utiliserons un IPM similaire à celui présenté par [4]. Le système utilise un IPM supposant une position nulle à la position verticale et une longueur de jambe constante.

Bien que l'IPM soit intéressant il limite les possibilités de mouvement. Par exemple il rend impossible un mouvement définissant une trajectoire rectangulaire pour le pied comme l'on voit sur la figure \*mettre nbr figure de la strip avec une marche carrée\*. Ce genre de déplacement est important dans le cadre de notre étude car elles sont typiques de lieux où la progression est beaucoup entravée (par exemple de la neige). Pour permettre l'observation de tels déplacements nous avons rajouté la possibilité de manuellement définir la position du pied au cours du pas. Cette définition manuelle est considérée temps que le personnage est dans la phase ascendante du pas  $V_{COM}(y) > 0$  et qu'il n'est pas dans un état instable. Nous déterminons une instabilité du personnage si nous détectons une variation irrégulière du pattern de vitesse observé au cours d'un pas (voir la section 6.3). Si un état instable est détecté nous utilisons le résultat de l'IPM pour la totalité du pas jusqu'à ce que nous retournions dans un état stable.

### 6.3 Contrôle de vitesse

Dans le cadre de ce projet, la capacité à marcher à une vitesse donnée est un critère primordial. De plus nous souhaitons obtenir un système qui sera robuste à la variation du milieu. C'est à dire que nous souhaitons que notre contrôleur soit capable de marcher dans un mètre d'eau ou dans 50cm d'huile aussi bien que lorsqu'il ne se trouve pas en présence d'un fluide. Pour obtenir un tel résultat, nous utilisons une force virtuelle ainsi qu'un système d'altération des résultats de l'IPM.

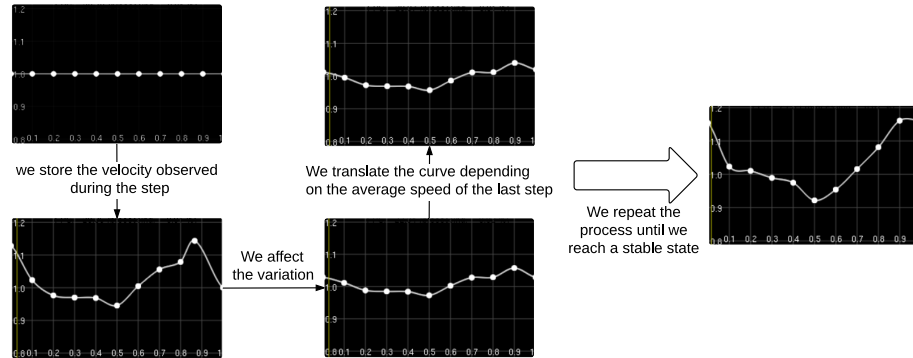
**Contrôle de vitesse par force virtuelle** Pour obtenir une vitesse désirée nous allons appliquer une force virtuelle horizontale qui accélérera ou ralentira le personnage en déplaçant le centre des masses. De manière similaire à [4] nous utilisons un PD-control prenant la l'écart entre la vitesse désirée et la vitesse actuelle du personnage pour calculer la force virtuelle nécessaire. Cependant la chaîne de membres du personnage considérés pour l'application de la force est la chaîne reliant le torse au pied d'appui (donc ne prend pas en compte la tête du personnage). La raison de ce choix est qu'il ne paraît pas naturel de pencher la tête si l'on veut aller dans une direction. Les valeurs utilisées pour la jacobienne sont différentes de celles-ci utilisées par [4]. Nous utilisons la somme des vecteurs séparant les articulations pondérés par le poids du membre présent entre ces deux articulations.

$$J(p)_n^T = \frac{1}{M} \sum_{0 < i \leq n} ((P_i(x, y, z) - P_{i-1}(x, y, z)) * M_i)$$

Avec  $M$  la somme des poids de tous les membres considérés par le système et  $P_i(x, y, z)$  la position de l'articulation  $i$  de la chaîne,  $P_0(x, y, z)$  étant le point d'application de la force. Cela évite que les articulation du bas du corps reçoivent une grande importance du fait qu'elles se trouvent loin du torse (qui représente une grande partie de la masse du personnage). Cette caractéristique est intéressante car un moment appliqué sur la cheville ayant pour but de faire pencher le torse aurait de grande chance de se faire annuler par un moment sur la hanche ou le pelvis ayant pour but de redresser le personnage. De plus il est plus simple de créer un déséquilibre régulier du personnage en appliquant un moment sur l'articulation entre le pelvis et le torse qu'en appliquant un moment sur la cheville ce qui renforce l'idée que le torse doit bien être considéré indépendamment du bas du corps.

Il est facile d'observer que la vitesse de déplacement du centre des masses n'est pas constante au cours d'un pas. De ce fait l'application d'une force virtuelle se basant sur une vitesse désirée constante aura pour effet de faire disparaître ce phénomène, voire demander localement au personnage de ralentir alors que l'on voudrait le voir accélérer de manière globale. C'est pourquoi il serait plus intéressant d'appliquer une force constante au cours d'un pas. Cependant ce genre de force constante serait dure à déterminer et diminuerait la robustesse du modèle car elle ne prendrait pas en compte les événements se passant au cours

d'un pas (i.e. ralentissements due au contact avec un objet pas exemple). On peut observer que lorsque nous sommes dans le cas d'un déplacement stable, ces variations de vitesse sont constantes à chaque pas. L'idée serait donc d'adapter la vitesse désirée que l'on communique au PD-contrôler suivant les variations de vitesse observées au cours du pas. C'est pourquoi nous avons construit un système permettant d'apprendre cette courbe de variations de vitesse de manière à appliquer une force virtuelle plus adaptée. De manière similaire aux trajectoires des articulations, cette courbe est définie comme une fonction de  $\phi$  et possède un nombre de point de référence pouvant être définis par l'utilisateur. Les valeurs entre les points de références sont également déterminées à l'aide de splines Catmull-Rom. Un nombre de 10 points de référence s'est montré être suffisant pour permettre l'obtention d'une courbe précise tout en évitant de surcharger le système. Le système utilise une courbe pour la vitesse suivant l'axe sagittal et deux courbe suivant l'axe coronal (une pour chaque phase d'appui). L'axe coronal nécessite deux courbes car dans le cas d'une vitesse non nulle selon l'axe coronal le mouvement observé devient asymétrique. Une autre différence entre les deux axes et que pour l'axe coronal nous stockons directement les vitesses en  $m.s^{-1}$  alors que pour l'axe sagittal nous stockons des facteurs multiplicateurs à appliquer à la vitesse désirée par l'utilisateur. L'utilisation des facteurs multiplicateurs nous permet une plus grande aisance à nous adapter aux variations de vitesses demandées par l'utilisateur. En effet pour une variation de vitesse de moyenne amplitude il est fortement possible que la courbe de facteurs multiplicateurs reste identique. Cependant vu que nous travaillons généralement avec une vitesse nulle suivant l'axe coronal, il nous est impossible de nous baser sur un coefficient multiplicateur.



**FIGURE 6.** Schéma présentant le processus d'apprentissage de la courbe de vitesse

Ce paragraphe explique de manière plus précise l'apprentissage des courbe de vitesse. Comme présenté sur le schéma 6 le système d'apprentissage des courbe de vitesse s'effectue par une boucle de 3 étapes répétées jusqu'à l'obtention d'une



courbe stable. La première étape est de relever, aux cours d'un pas, la vitesse réellement observée pour chaque point de référence. Une fois le pas terminé nous calculons la variations moyenne entre la courbe que nous avons et la courbe observée. La deuxième étape consiste à, si la variation moyenne observée est inférieur à une heuristique, adapter les valeurs stockées pour les rapprocher des valeurs observées. Nos test nous ont montré qu'une heuristique de 0,4 donne des résultat satisfaisant. Nous notons tout de même qu'il peut être intéressant d'utiliser une heuristique plus élevée au début de la phase d'apprentissage car il est possible que la trajectoire initialement utilisée et très différente de la trajectoire observée. Pour éviter d'obtenir d'oscillations autour de la courbe ver laquelle on cherche à converger, la variation de chaque point est limitée à une variation d'au maximum la valeur de la variation moyenne constatée. Pour finir nous modifions les points de la courbe de façon à ce que la nouvelle courbe définisse les valeurs à demander pour obtenir la vitesse voulue. Pour obtenir ce résultat nous déterminons le ratio entre la vitesse moyenne demandée et la vitesse moyenne observées lors du dernier pas et nous multiplions tout les points de la courbe par ce ratio. Dans le cas de l'axe coronal nous utilisons le ratio entre la vitesse demandée et la moyenne des vitesses observées lors des deux dernier pas.

La présence de l'heuristique nous permet de détecter les variations irrégulières de la courbe de vitesse. Une fois la situation d'équilibre atteinte nous obtenons une courbe stable qui est caractéristique au déplacement observé. Si l'on constate une variation de cette courbe cela veut dire qu'un élément est venu perturber le déplacement. Dans ce cas, nous déclarons au contrôleur que nous ne sommes plus dans un état stable ce qui signale au contrôleur qu'il doit utiliser l'IPM durant toute la durée du pas jusqu'à ce que le mouvement redevienne stable. Une fois le mouvement redevenue stable, c'est à dire que la variation moyenne de la vitesse est redevenue inférieure à l'heuristique, nous recommençons à utiliser le système décrit précédemment pour adapter la courbe de vitesse. Si le système détecte plus de 5 pas consécutif pour lesquels le mouvement est déclaré instable, alors nous réinitialisons tous les points des courbes de vitesses à la valeur de la vitesse désirée spécifiée par l'utilisateur. Cela nous permet de restaurer le système dans le cas ou les courbes auraient subit une variation très forte les éloignant trop de la courbe résultant de l'IPM ce qui bloque le système.

**Altération des résultats de l'IPM** Bien que le système utilisant une force virtuelle pour contrôler la vitesse permette un réglage fin de la vitesse, il ne permet pas d'effectuer de grandes variations de vitesses. En effet, pour réaliser de telles variations nous aurions besoin d'une force virtuelle très importante ce qui aurait des effets indésirables. Pour régler ce problème [4] propose d'altérer les résultats de l'IPM pour aider à atteindre la vitesse voulue. Cependant, cette modification est basée sur une relation linéaire dépendant de la vitesse désirée. Cette relation linéaire rend le système trop peu efficace notamment pour des mouvement dans un fluide qui ralentira le personnage.

Vue qu'il semble impossible d'utiliser une simple loi dépendant de la vitesse désirée, nous avons décidé d'utiliser un système qui stocke un  $\Delta(x, z)$  qui sera

ajouté au résultats de l'IPM. Pour pouvoir déterminer la valeur de ce  $\Delta$  nous utilisons une méthode incrémentale. A la fin de chaque pas nous utilisons un PD-contrôleur se basant sur la différence entre la vitesse désirée et la vitesse réellement observée au cours du pas pour déterminer comment doit évoluer le  $\Delta(x, z)$  pour permettre au personnage de se déplacer à la vitesse désirée. Nous n'utilisons ce système que sur l'axe sagittal. Les vitesses demandées sur l'axe coronal étant plus faibles le personnage n'a pas besoin de ce système pour les atteindre.

Ce système de contrôle de vitesse est extrêmement efficace et robuste. Cependant, si l'on utilise un  $\Delta(x, z)$  trop élevé dans le but d'accélérer le personnage cela risque d'empêcher l'IPM de maintenir l'équilibre. C'est pourquoi nous avons mis une limite aux valeur du  $\Delta(x, z)$ . Nous fixons cette valeur à 0,09m, ce qui correspond à la valeur maximale permettant de conserver l'équilibre avec un personnage immergé jusqu'à la taille dans de l'eau, les autres situations de test supportant des valeurs plus élevées cela parait être une bonne sécurité.

#### 6.4 Optimisation hors ligne

Dans le cadre de notre projet nous voulons être capable de simplement prendre les paramètres d'un contrôleur ne considérant pas les interactions avec les fluide pour les utiliser dans notre système (i.e. les gains des différents PD-contrôleurs). C'est pourquoi notre stratégie d'optimisation hors ligne ne considèrera qu'une nombre limité de paramètres présents dans le système. Les paramètres que nous optimiserons seront les trajectoires associées aux éléments suivants :

- le pelvis;
- l'articulation du dos;
- la cheville d'appui;
- la cheville en phase de vol;
- la trajectoire du pied en phase de vol;
- le genou d'appui;

Nous n'optimisons pas les articulations des bras et du cou de manière à obtenir un modèle à partir duquel un utilisateur peut spécifier des actions spécifiques utilisant ces membres.

**Fonction objectif** Nous allons construire notre fonction d'optimisation en utilisant d'une part une combinaison pondérée de fonctions d'évaluation élémentaire représentant un but physique et d'autre part des règles représentant des objectifs à atteindre.

- *Minimisation de l'énergie* ( $f_{energ}$ ). Le but de cette fonction est de faire tendre le résultat vers une simulation utilisant le minimum d'énergie possible. Cela nous permettra d'éviter les mouvement superflus. Cette fonction sera déterminé par le calcul de la somme des moments aux articulations;

- *Minimisation de la résistance de l'eau.* Cette fonction aura pour effet de pousser le personnage à tenter de sortir de l'eau autant que possible et à chercher des trajectoires évitant les pics de vitesses en dessous de l'eau. Nous calculons cette fonction à travers la somme des moments produits sur chacune des articulations par la force de résistance de l'eau sur le membre manipulé par l'articulation ;
- *Minimisation des accélérations angulaires ( $f_{acc}$ ).* Notre but est de parvenir à obtenir non seulement un mouvement lisse mais aussi des trajectoires désirées lisses. Cette fonction vient du fait que nous avons observé des fortes variations dans les positions désirées pour certaines articulations (i.e. la cheville d'appui) nous faisant penser que la méthode d'évolution avait appris des détails optimisant le mouvement mais le rendant moins robuste. Nous espérons donc qu'en pénalisant les variations au sein des trajectoires nous obtiendrons des trajectoires représentant les caractéristiques principales du mouvement. Nous calculons cette fonction en faisant simplement la somme des accélérations angulaires des poses désirées et sur mouvement observé. Nous signalons que nous avons attribué une pondération de 0.25 au accélération des poses désirées et une pondération de 0.75 au accélération sur mouvement observé ;
- *vérification de la vitesse ( $f_{speed}$ ).* Bien que nous disposons de systèmes nous assurant que la vitesse spécifiée sera la vitesse respectée, il est possible que l'optimisation arrive à trouver des combinaisons de paramètres qui rendraient la convergence vers la vitesse désirée très longue voire impossible (par exemple si l'on a une vitesse positive avec des paramètres pour aller vers l'arrière). Cette condition est représenté par une pénalisation très élevé si une précision de 1% n'est pas atteinte pour la vitesse demandée ;
- *stabilité du mouvement.* Ce critère sert à nous assurer que nous nous trouvons bien dans un état stable, c'est à dire un état utilisant la spécification manuelle du pied en phase de vol pour le début du pas. Cette condition est représentée par une pénalisation très élevé si l'on détecte un pas pour lequel le mouvement est déclaré instable ;
- *Minimisation de l'altérateur d'IPM.* Comme nous l'avons expliqué lors de la présentation de ce système, l'utilisation de valeurs élevées pour l'altérateur d'IPM peut rendre le mouvement instable. De plus ce système entraine une modification majeure des résultats de la simulation (en général il raccourci les pas du personnage). C'est pourquoi nous avons ajouté une fonction pénalisant l'utilisation de ce système. Cette pénalisation est représenté par une augmentation de la somme des autres fonctions d'évaluation par un facteur multiplicateur de  $0.1 * \frac{\Delta(x,z)}{\max(\Delta(x,z))}$ , avec  $\Delta(x,z)$  la valeur de l'alterateur pour obtenir l'état d'équilibre et  $\max(\Delta(x,z))$  la valeur maximale autorisée (0.09).

Donc notre fonction d'évaluation aura la forme suivante :

$$f_{eval}() = \begin{cases} max_{eval} & \text{if not stable} \\ max_{eval}/1000 & \text{if } ||V - V_d|| > 0.01 * V_d \\ (\alpha f_{energ} + \beta f_{acc} + \gamma f_{speed}) * 0.1 * \frac{\Delta(x,z)}{max(\Delta(x,z))} & \text{else} \end{cases}$$

**Méthode d'évolution des paramètres** Nous nous trouvons dans le cas d'un système espace complexe de variables avec de nombreux minimum locaux. De manière similaire à de nombreux articles concernant l'animation basée physique (\*citer 3/4 articles\*) nous utiliserons la méthode de "covariance matrix adaptation" (CMA) [?] pour explorer l'espace des solutions. CMA est une méthode d'évolution cherchant à apprendre la matrice de covariance de l'espace de recherche à l'aide d'un échantillonnage aléatoire.

## 7 Résultats

\*donner paramètres, méthodologies utilisée pr les expériences, capacités système\*  
 \*faire un comparatif visuel des mouvement obtenus suivant le niv d'eau pr les diff combinaisons d'évaluations\*  
 \*faire une comparaison en désactivant les systèmes créés\*  
 \*si il y en a présenter lestruc qui peuvent être utilisé en tant que feed back\*  
 \*faire une partie limitation\*

## 8 Conclusion

\*faire une conclusion qui contient la partie travaux futur\*

## Références

1. Barela, A.M., Stolf, S.F., Duarte, M. : Biomechanical characteristics of adults walking in shallow water and on land. *Journal of Electromyography and Kinesiology* 16(3), 250–256 (2006)
2. Chevutshi, A., Alberty, M., Lensel, G., Pardessus, V., Thevenon, A. : Comparison of maximal and spontaneous speeds during walking on dry land and water. *Gait & posture* 29(3), 403–407 (2009)
3. Coros, S., Beaudoin, P., van de Panne, M. : Robust task-based control policies for physics-based characters. In : *ACM Transactions on Graphics (TOG)*. vol. 28, p. 170. ACM (2009)
4. Coros, S., Beaudoin, P., van de Panne, M. : Generalized biped walking control. In : *ACM Transactions on Graphics (TOG)*. vol. 29, p. 130. ACM (2010)
5. Geijtenbeek, T., Pronost, N. : Interactive character animation using simulated physics : A state-of-the-art review. In : *Computer Graphics Forum*. vol. 31, pp. 2492–2515. Wiley Online Library (2012)

6. Geijtenbeek, T., Pronost, N., van der Stappen, A.F. : Simple data-driven control for simulated bipeds. In : Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. pp. 211–219. Eurographics Association (2012)
7. Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K., Hirukawa, H. : The 3d linear inverted pendulum mode : A simple modeling for a biped walking pattern generation. In : Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on. vol. 1, pp. 239–246. IEEE (2001)
8. Kwatra, N., Wojtan, C., Carlson, M., Essa, I., Mucha, P.J., Turk, G. : Fluid simulation with articulated bodies. Visualization and Computer Graphics, IEEE Transactions on 16(1), 70–80 (2010)
9. Lentine, M., Gretarsson, J.T., Schroeder, C., Robinson-Mosher, A., Fedkiw, R. : Creature control in a fluid environment. Visualization and Computer Graphics, IEEE Transactions on 17(5), 682–693 (2011)
10. Miyoshi, T., Shirota, T., Yamamoto, S.I., Nakazawa, K., Akai, M. : Functional roles of lower-limb joint moments while walking in water. Clinical Biomechanics 20(2), 194–201 (2005)
11. Mussa-Ivaldi, F.A. : Nonlinear force fields : a distributed system of control primitives for representing and learning movements. In : Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on. pp. 84–90. IEEE (1997)
12. Neff, M., Fiume, E. : Modeling tension and relaxation for computer animation. In : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation. pp. 81–88. ACM (2002)
13. Orselli, M.I.V., Duarte, M. : Joint forces and torques when walking in shallow water. Journal of biomechanics 44(6), 1170–1175 (2011)
14. Van de Panne, M., Lamouret, A. : Guided optimization for balanced locomotion. Springer (1995)
15. Si, W., Lee, S.H., Sifakis, E., Terzopoulos, D. : Realistic biomechanical simulation and control of human swimming. ACM Transactions on Graphics (TOG) 34(1), 10 (2014)
16. Stam, J. : Stable fluids. In : Proceedings of the 26th annual conference on Computer graphics and interactive techniques. pp. 121–128. ACM Press/Addison-Wesley Publishing Co. (1999)
17. Tan, J., Gu, Y., Turk, G., Liu, C.K. : Articulated swimming creatures. In : ACM Transactions on Graphics (TOG). vol. 30, p. 58. ACM (2011)
18. Yang, P.F., Laszlo, J., Singh, K. : Layered dynamic control for interactive character swimming. In : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation. pp. 39–47. Eurographics Association (2004)
19. Yin, K., Loken, K., van de Panne, M. : Simbicon : Simple biped locomotion control. In : ACM Transactions on Graphics (TOG). vol. 26, p. 105. ACM (2007)