

SIMBICON: Simple Biped Locomotion Control

Théo Delemazure

Janvier 2019

1 Introduction

Ce rapport présente le papier **SIMBICON : Simple Biped Locomotion Control** de *KangKang Yin, Kevin Loken et Michiel van de Panne*, datant de 2007. Dans cet article, ils présentent une méthode simple pour simuler le déplacement d'un bipède, en deux et trois dimensions.

Il est intéressant de pouvoir modéliser le mouvement des bipèdes, car il possède d'importantes implications, notamment en robotique et en animation. L'avantage de la méthode proposée et qu'en prenant en compte la physique, on obtient un vaste ensemble de déplacements possibles, par exemple pour la réaction à un coup, alors que les méthodes utilisées auparavant en animation ne capturaient qu'une seule version possible de l'action (en utilisant un déplacement humain ou en dessinant image par image).

Cependant, cela reste un problème compliqué, car le déplacement des bipèdes est souvent instable à cause de ses nombreux degrés de liberté et nombreuses dimensions. Ce problème possède également de nombreuses contraintes pour coller à la physique, et il faut réussir à obtenir un résultat à la fois robuste, fidèle, et peu coûteux en énergie.

Je vais présenter ici l'approche utilisée pour le mouvement bipède : Le point central est l'utilisation d'une *machine à états finis* dans laquelle les états sont des positions du corps (avec les angles des membres précisés) et les transitions surviennent après un certain temps ou lors d'un contact avec le sol. Comme cette machine seule n'est pas suffisante pour avoir un modèle robuste, on ajoute quelques modifications. On ajoute notamment un rétrocontrôle continu de l'équilibre sur l'angle objectif de la jambe en mouvement.

Les paramètres de cette méthode peuvent être définis manuellement ou bien à l'aide de *motion capture*. Ce modèle a pu être appliqué avec succès à des modèles bipèdes en 2D et 3D.

2 Approche pour le contrôle de l'équilibre

La stratégie proposée par SIMBICON utilise 3 outils : une machine à états finis pour l'enchaînement des poses, un contrôleur de l'orientation des membres pour aller d'une pose à l'autre et du rétrocontrôle pour ajuster l'angle souhaité des membres du corps.

2.1 Machine à états

Chaque état de la machine correspond à une pose et un angle précis pour les jambes et le torse. Un exemple pour la marche est présenté en figure 1.

Premièrement, on constate que la machine est symétrique pour les deux parties du corps (1 \rightarrow 2 fait écho à 3 \rightarrow 4). Certaines transitions ont lieu après un certain temps (transition 1 \rightarrow 2) ou après un contact avec le sol (transition 2 \rightarrow 3).

Pour obtenir la pose attendue à l'état courant, on applique un couple calculé par une formule de *contrôleur proportionnel dérivé (PD)* pour chaque articulation du robot : $\tau = k_p(\theta_d - \theta) - k_d\dot{\theta}$. Ainsi,

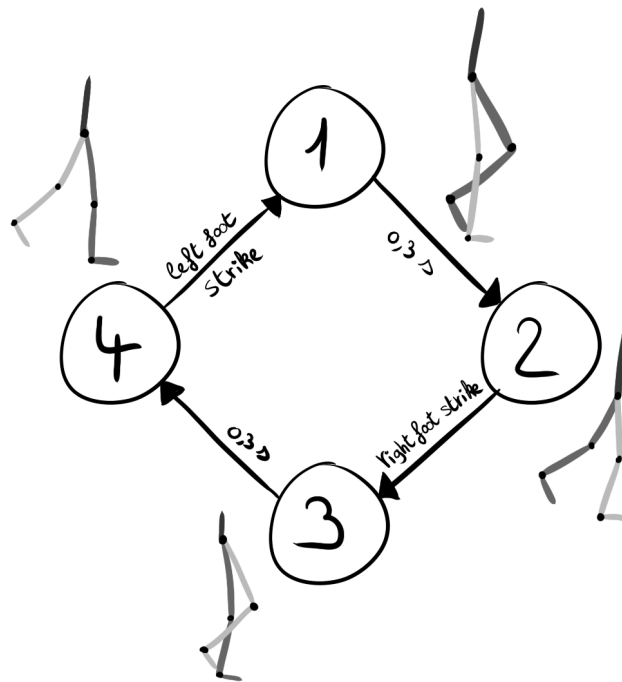


Figure 1: Machine à état finis pour un cycle de marche

la pose du robot va tendre vers l'état courant de la machine à état sans réellement l'atteindre. Même si l'état n'est pas atteint, la démarche du robot doit correspondre à celle qui est souhaitée.

2.2 Contrôle des hanches et du torse

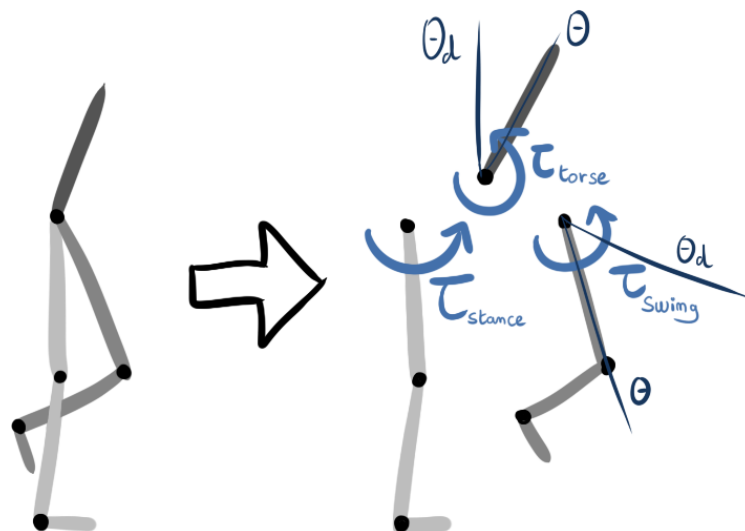


Figure 2: Contrôle séparé des hanches et du torse

Tout d'abord, on veut que le torse tende vers une position de référence. On utilise alors un contrôleur PD virtuel pour calculer un couple net τ_{torse} .

On veut également que la position du pied en mouvement soit correctement définie par rapport à l'angle courant du torse. On calcule alors τ_{swing_hip} en utilisant un contrôleur PD virtuel utilisant l'angle du torse.

Enfin, comme on ne veut que des couple internes, on souhaite $\tau_{torse} + \tau_{swip_hip} + \tau_{stance_hip} = 0$. On en déduit le couple appliqué à la hanche dont le pied est au sol : $\tau_{stance_hip} = -\tau_{torse} - \tau_{swip_hip}$.

2.3 Retrocontrôle de l'équilibre

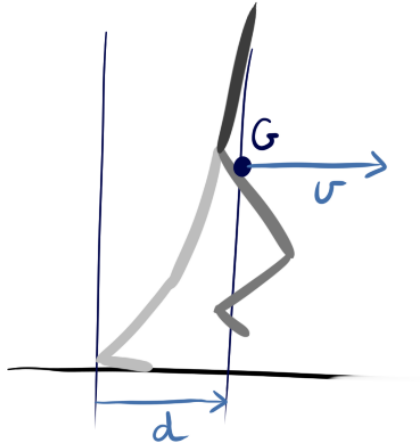


Figure 3: Paramètres du retrocontrôle pour θ_d

Pour que le robot puisse réagir correctement au terrain, on utilise un rétrocontrôle de la position cible. On va ainsi adapter l'angle cible de la hanche en mouvement à la disposition et la vitesse courante de tout le corps selon la formule suivante :

$$\theta_d = \theta_{d_0} + \alpha_d \times d + \alpha_v \times v$$

où θ_{d_0} est l'angle cible proposé dans l'état courant de la machine à l'état, d est la distance entre le centre de masse et le pied au sol et v la vitesse du centre de masse (approximé au milieu de la hanche en mouvement). Il est important que $\alpha_d \neq 0$ car la vitesse n'est pas suffisante pour connaître la phase actuelle du robot dans son cycle de marche (par exemple pour une marche sur place avec $v \simeq 0$).

Pour les robots en 3D, on utilise cette méthode de rétrocontrôle à la fois sur le plan coronal et sagittal.

3 Comment choisir les paramètres ?

La méthode présentée en partie 3 possède un grand nombre de paramètres :

1. Nombre d'états de la **machine à états** et types des transitions.
2. Les paramètres du **rétrocontrôle de l'équilibre** α_v et α_d .
3. **Les poses cibles** de chaque état de la machine à états.
4. L'état **initial** du robot.
5. Les **limites physiques** pour les angles, couples, etc... (non détaillés dans le papier)

Ces paramètres peuvent être sélectionnés manuellement ou à l'aide de la technologie de *motion capture*.

3.1 Paramétrage manuel

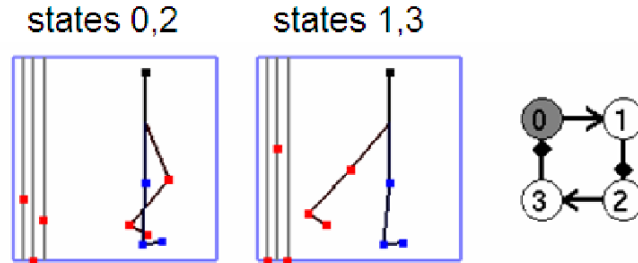


Figure 4: Interface graphique pour paramétrages manuel

Dans le cas de la marche, les auteurs ont construit une machine à 4 états qui fonctionnent par paires symétriques avec pour chaque pied une transition selon une durée et une autre après un contact avec le sol (voir Figure 1). Dans le cas de la course, on utilise seulement deux états.

On peut passer d'une machine de contrôle à une autre en passant de l'état n de l'une à l'état $(n + 1) \bmod 4$ de l'autre. On ajoute ainsi 2 états transitoire de durée nulle à la machine à état de course (afin d'avoir une machine à 4 état).

Pour le reste des paramètres, les auteurs ont simplement fait une interface graphique permettant à l'utilisateur de choisir les poses liées aux états et les paramètres du rétrocontrôle (voir Figure 4). On constate que le temps de transition entre les poses sans contact a une grosse influence sur la démarche finale.

3.2 Utilisation de la *Motion capture*

On peut également utiliser la *motion capture* pour définir les paramètres du robot qui imitera la démarche du sujet. L'utilité de la méthode SIMBICON avec la motion capture et qu'en imitant la démarche, le robot pourra également réagir à des modifications du terrain ou des coups.

Pour calculer les paramètres à partir des enregistrement de motion capture, on procède ainsi :

1. On récupère tout d'abord environ 6 cycles de la démarche que l'on veut imiter.
2. On fait une décomposition de Fourier sur un des angles, par exemple celui de la hanche droite afin de trouver la période T du mouvement. On prend également en compte les fréquences de plus grandes amplitudes dans la décomposition.
3. On peut alors apprendre un mouvement périodique Θ dont on détecte facilement les extremums. Le maximum de l'angle de la hanche droite correspondra à une phase de $\phi(t_{max}) = 1/4$. On calcule également les instants correspondant à un contact auxquels on associe les phases $\phi = 0.5$ et 1 .
4. On utilise alors les contrôleurs PD vus en partie 2 avec $\theta_{d_0} = \Theta(\phi(t))$.
5. On peut calculer de la même façon la vitesse périodique $\dot{\Theta}$ et on réécrit la formule du contrôleur PD :

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

Cependant, cette méthode possède quelques défauts : Il y a souvent du bruit indésirable qui est capté par la motion capture et les paramètres physiques de l'acteur ne correspondent pas forcément à ceux du robot.

4 Optimisation

Dans les simulations, le torse du robot suit un mouvement oscillant peu naturel. En effet, il réagit au mouvement des hanches au lieu de l'anticiper, ce qui donne l'impression que le torse n'est pas synchronisé avec les jambes. Pour corriger cela, les auteurs utilisent du *Feedback error learning* sur la fonction de calcul du couple τ .

5 Résultats

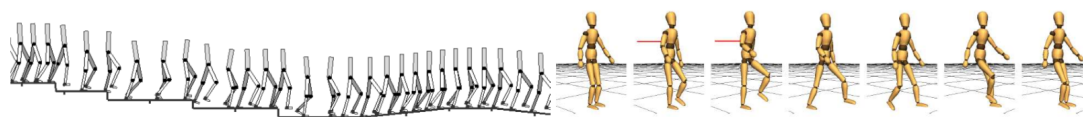


Figure 5: Résultats de la méthode SIMBICON sur des terrains avec obstacles

Un grand nombre de démarche ont été simulées avec succès grâce à SIMBICON (marche simple, avec genoux levés, recroquevillé, course rapide, à l'envers, sauter au-dessus d'un obstacle, ...) et la simulation peut passer aisément d'une démarche à une autre en allant de l'état n d'une machine à l'état $n + 1$ d'une autre. On constate que certaines démarches sont très sensible à l'état initial.

La simulation a été soumise à des variations de terrains (voir figure 5) et des coups séparés de 5s, laissant le temps au robot de retrouver son cycle normal. Chaque démarche a une limite à partir de laquelle elle ne se remet pas d'un coup, donné par $\sigma_{lim} = \Delta t F$ où Δt est la durée du coup et F sa force.

6 Conclusion

Le défi du mouvement bipède est très important dans l'animation et la robotique et ce papier propose une manière simple de programmer ce mouvement et de réagir aux éléments imprévus comme des coups ou des marches d'escalier inattendues.

De part sa simplicité et sa robustesse, SIMBICON est resté longtemps *l'état de l'art* en matière de simulation du mouvement bipède et est encore utilisé aujourd'hui.

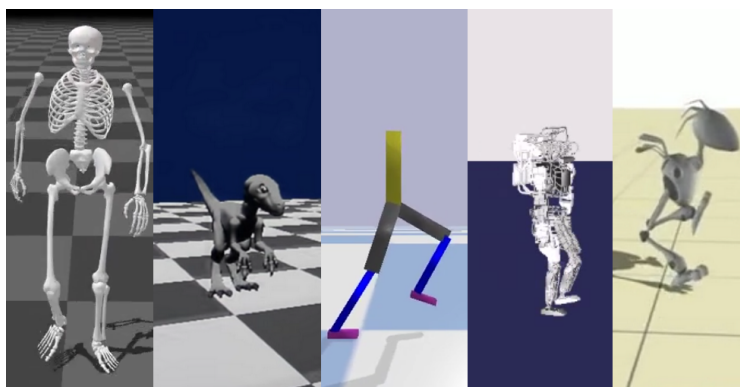


Figure 6: Différents exemples de simulations utilisant SIMBICON pour de l'animation