遷移可能な状態を探索する キャラクタモーションジェネレータの実現

時崎　崇[*1]　長谷川　晶一[*1]

Motion Generation Method with Searching Transferable States based on Dynamics and Physical Characteristic Features

Takashi Tokizaki[*1] and Shoichi Hasegawa[*1]

Abstract – **Recently, more and more video games becoming to allow users design characters. In such games, it needs to generate motions of various characters in the contents. We propose a method which generates motions of characters automatically considering physical characteristic features using dynamics simulation. Our method to generate motions consists of two elements: genetic algorithm and reinforcement learning. The former narrows down the configuration space of the character body model. The latter searches the best behavior in the narrowed space. Thus the method generates natural motions of various bodies automatically.**

Keywords : **Motion, Articulated Body, Reinforcement Learning, Genetic Algorithm, Physics Simulation**

1. 背景及び目的

今後ユーザがゲーム内でキャラクタを自由にデザインするコンテンツが増えていくことが予想される．現状ではその様なコンテンツのキャラクタの動きは，動作をあらかじめ定義しておき，ユーザの作成した形状に応じてそれを再構成することでユーザが動作まで作らずに済む様に配慮されている．[1] しかしその弊害として身体の勢い，重心位置等の動力学的制約や，関節の出力，可動域，手足同士の干渉といった身体的制約を満たせない不自然な動作が生成されるという問題が起きている．

そこで本研究では上記の問題を解決するために形状を決定した後で動力学的制約や身体的制約を満たす自然な動作を生成することを目的とする．

方針として

- 物理シミュレータを使用：干渉や重心位置といった動力学的制約を満たす動作を実現
- 学習による動作の自動生成：キャラクタの形状に応じた身体的制約を満たす動作の獲得

とすることで本目的の達成を狙う．

*1: 電気通信大学大学院, {tokizaki, hase}@hi.mce.uec.ac.jp
*1: Graduate School, The University of Electro-Communications

2. 自動生成の問題と提案システム

2.1 自動生成の問題

キャラクタのアニメーションはキーフレームと呼ばれるアニメーション内の特徴的な姿勢について，その系列を指定することで生成される．アニメーションの自動生成，すなわち系列の指定をコンピュータのみで行うことになるが，キャラクタの自由度が高くなるとキャラクタの取れる姿勢の選択肢も膨大になり，解となる非常にわずかなキーフレーム系列を適切に選択することが難しくなる．この問題を解決するためには，その膨大な姿勢の中から動力学的・身体的制約を満たさない領域を排除し，残った部分からタスクが最適条件で達成される系列を選択する必要がある．

2.2 提案手法とシステム

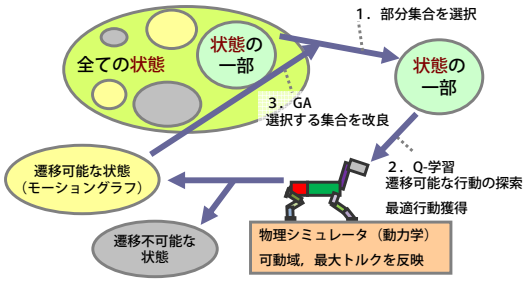

図 1　提案システム概要

そこで本研究では，動力学的・身体的制約を満たす動作の生成のために動力学シミュレーション上で可動域や最大トルクの考慮を行い，自動生成の問題を解決するために伊藤らが提案する QDSEGA [2] の動力学への拡張を行う．QDSEGA は学習で一般に問題となる状態数の爆発を冗長度に依存する計算に問題を変更する特徴を持ち，以下の手順を続ける事でより良い部分空間を抜き出し解を求めることが出来る．

1. 全ての状態から部分集合を抜き出す
2. その部分空間に対して Q 学習を行い部分空間での最適行動を学習させる
3. 学習結果からタスクの達成度を求めそれを基に部分集合を GA により選びなおす

従来，本アルゴリズムは状態を各関節の角度とすることで静力学の考えに基づいて用いられてきたが，本提案システムでは，状態を各関節の角度と角速度とした動力学を考慮したシステム構築を行う．本アルゴリズムで学習する動作は目的の運動を構成する状態の系列である．すなわち動作とは状態間の遷移であり，学習の結果選ばれた状態同士をスプライン補間した目標軌道への軌道追従制御で生成するものとする．

以下，軌道追従制御の物理シミュレータへの組み込みについて説明の後，学習部の説明を行う．

3. 軌道追従制御について

本提案システムでは，身体的制約を考慮するために軌道追従制御によりアニメーションを生成する．軌道追従制御は質量 M, バネ係数 K, ダンパ係数 D のシステムの目標値と実際の値との偏差 $e = x_{ideal} - x_{real}$ に対して，

$$M\ddot{e} + D\dot{e} + Ke = 0 \tag{1}$$

として制御する方法である．

本システムでは，動作の再生に物理シミュレータを用いている．今回はその中で，LCP(Linear Complementary Problem：線形相補性問題) を用いたソルバーへの帰着を行った．

まず，節 3.1 において剛体の拘束運動を LCP に帰着する方法について簡単に述べ，次に節 3.2 でこの枠組みに速度拘束を組み入れる方法を，節 3.3 で指定のトルクとバネ・ダンパの効果を組み入れる方法を述べる．

3.1 LCP への帰着

時刻 t における剛体の位置，速度を表わす変数をそれぞれ $p(t)$，$v(t)$ とおく．剛体に作用する拘束は等式拘束と相補性拘束に分けられる．

等式拘束は，主に関節による剛体の連結を表現するために用い，相補性拘束は，主に剛体同士の接触を表現するために用いる．

ここで等式拘束および相補性拘束によって拘束される速度変数をそれぞれ w_e，w_c とおく．また，対応する拘束力を λ_e, λ_c とおく．すると，これらは適切なヤコビ行列 J_e 及び J_c を用いて

$$w_e(t) = J_e(t)\, v(t) \tag{2}$$

$$w_c(t) = J_c(t)\, v(t) \tag{3}$$

と表わせる．また，これらをまとめて

$$w = \begin{bmatrix} w_e \\ w_c \end{bmatrix},\ J = \begin{bmatrix} J_e \\ J_c \end{bmatrix},\ \lambda = \begin{bmatrix} \lambda_e \\ \lambda_c \end{bmatrix} \tag{4}$$

と表記する．

まず，剛体の運動は Newton-Euler の運動方程式に基づき以下のように記述される．

$$M\dot{v}(t) = f(t) + J(t)^T\, \lambda(t) \tag{5}$$

ここで M は質量行列，$f(t)$ は外力とコリオリ項を含むベクトルである．

一方，拘束条件は次のように記述される．

$$w_e(t) = \mathbf{0} \tag{6}$$

$$w_c(t) \geqq \mathbf{0},\ \lambda_c(t) \geqq \mathbf{0},\ w_c(t)^T\, \lambda_c(t) = 0 \tag{7}$$

ただしベクトルに関する不等号は，各成分について不等号が成り立つことを意味する．

さて，動力学シミュレーションでは，常微分方程式 (5) を元に何らかの離散時間上の更新則を導く必要がある．

以下では t は離散化された時刻を表わすものとし，時刻 t の値を $[t]$ で表記する．本手法では各時刻 t において以下の手順で更新を行う．

1. 拘束力 $\lambda[t]$ の計算
2. 剛体の速度の更新

$$v[t+1] = v[t] + M^{-1}(f[t] + J[t]^T\lambda[t])h \tag{8}$$

3. 剛体の位置の更新

$$p[t+1] = p[t] + v[t+1]h \tag{9}$$

ここで h は積分ステップである．手順 1 において計算する拘束力 $\lambda[t]$ は，ステップ 2(速度の更新) の直後，ステップ 3（位置の更新）の直前において $v[t+1]$ が拘束条件を満たすように求める．

すなわち，拘束条件は

$$w_e[t+1] = J_e[t]v[t+1] \tag{10}$$

$$w_c[t+1] = J_c[t]v[t+1] \tag{11}$$

と書ける．式 (8) 式 (9) より，拘束力を求める問題は次の線形相補性問題 (LCP) に帰着される．

$$\begin{bmatrix} w_e[t+1] \\ w_c[t+1] \end{bmatrix} = A \begin{bmatrix} \lambda_e[t] \\ \lambda_c[t] \end{bmatrix} + b \tag{12}$$

$$A = J[t]M^{-1}J[t]^T h \tag{13}$$

$$b = J[t]\{v[t] + M^{-1}f[t]h\} \tag{14}$$

$$w_e[t+1] = \mathbf{0} \tag{15}$$

$$w_c[t+1] \geqq \mathbf{0},\ \lambda_c[t] \geqq \mathbf{0},\ w_c[t+1]^T \lambda_c[t] = 0 \tag{16}$$

次節では，上述の枠組みにバネ・ダンパの効果を組み入れる方法を述べる．

3.2 速度制御の LCP ソルバーへの組み込み

速度制御は前節の等式拘束に組み込むことができる．速度制御の目標速度を $\boldsymbol{w}_0$ とすると，前節の 式 (2) を，

$$w_e[t+1] = \boldsymbol{w_0} \tag{17}$$

とすることで，速度制御を LCP の拘束条件に組み込むことができる．

3.3 LCP ソルバーへのバネダンパ及び補正トルクの導入

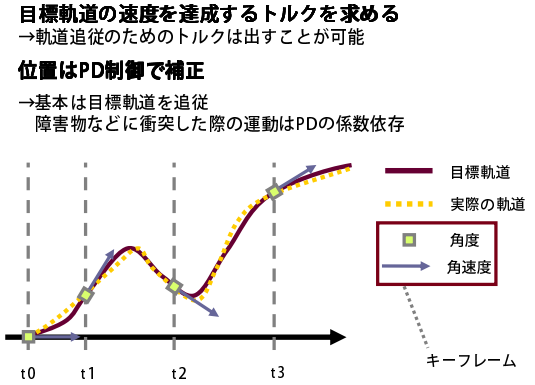

図 2　軌道追従制御の表現

軌道追従制御を実現するために，目標軌道の角速度を達成するためのトルク f_0 を節 3.2 で述べた速度制御を用いて算出する．これにより目標軌道を追従する力が求まるので，角度の補正にバネ・ダンパを用いることで軌道追従制御を実現する．以下に LCP への拘束条件の組み込み方を述べる．

前節で定義した w_e, w_c に加えて，バネ・ダンパが作用する速度変数として w_s を考える．w_s について他と同様に適切なヤコビ行列を用いて

$$w_s(t) = J_s(t)\, v(t) \tag{18}$$

と表わせる．さて，線形バネ・ダンパにおける拘束力と速度との関係は

$$\begin{aligned} \lambda_s[t] &= -Kq_s[t+1] - Dw_s[t+1] + f_0 \\ &\approx -K(q_s[t] + w_s[t+1]h) - Dw_s[t+1] + f_0 \end{aligned} \tag{19}$$

と書ける．ここで $q_s[t]$ は時刻 t におけるバネの変位を表わす．さらに，上式は次のように変形される．

$$\begin{aligned} w_s[t+1] = &-(D+Kh)^{-1}\lambda_s[t] \\ &-(D+Kh)^{-1}(Kq_s[t] - f_0) \end{aligned} \tag{20}$$

ここで改めて

$$w = \begin{bmatrix} w_e \\ w_c \\ w_s \end{bmatrix},\ J = \begin{bmatrix} J_e \\ J_c \\ J_s \end{bmatrix},\ \lambda = \begin{bmatrix} \lambda_e \\ \lambda_c \\ \lambda_s \end{bmatrix} \tag{21}$$

と再定義すると，前節と同様にして

$$\begin{bmatrix} w_e \\ w_c \\ w_s \end{bmatrix} = A \begin{bmatrix} \lambda_e \\ \lambda_c \\ \lambda_s \end{bmatrix} + b \tag{22}$$

$$A = J[t]M^{-1}J[t]^T h \tag{23}$$

$$b = J[t]\{v[t] + M^{-1}f[t]h\} \tag{24}$$

となる．ここで式 (22) に式 (20) を代入し，これを右辺に移項すると

$$\begin{bmatrix} w_e[t+1] \\ w_c[t+1] \\ \mathbf{0} \end{bmatrix} = \tilde{A} \begin{bmatrix} \lambda_e[t] \\ \lambda_c[t] \\ \lambda_s[t] \end{bmatrix} + \tilde{b} \tag{25}$$

$$\tilde{A} = J[t]M^{-1}J[t]^T h + \begin{bmatrix} O & O & O \\ O & O & O \\ O & O & (D+Kh)^{-1} \end{bmatrix} \tag{26}$$

$$\tilde{b} = J[t]\{v[t] + M^{-1}f[t]h\} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ (D+Kh)^{-1}(Kq_s - f_0) \end{bmatrix} \tag{27}$$

を得る．上式より，バネ・ダンパ拘束とオフセットトルクは，修正された係数行列・係数ベクトル $\tilde{A}, \tilde{b}$ によって定義される LCP における等式拘束として等価的に表現できる．

3.4　学習による動作の獲得機能の実装

学習によって獲得される動作は状態の系列であるが，スプライン補間される目標軌道は，目標値によっては設定した関節の可動域の外へ軌道が侵入してしまう可能性がある．そこで軌道追従制御の目標軌道が可動域外を通ることを避けるために，学習する離散状態を以下のように設定した．まず，状態を構成する角度と角速度のうち，角度の状態分割を可動域内で行った．次に，分割された角度の場所に応じて角速度の分割数を変更した．角速度の分割は可動域限界近くでは，可動域の外へ向かう向きの角速度を出さないように区切り，可動域中央ではどちらの方向へも角速度が出ると仮定して分割を行った．

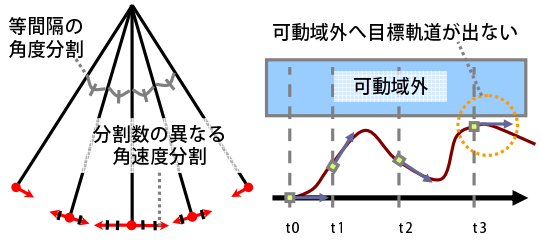

図 3　角度と角速度の可動域内での分割

また，この時強化学習で一意な状態を指定するために体幹に対して回転運動しないという拘束を掛けた．

4.　動作検証

小節 3.4 のように分割された状態で，第 2.2 節の処理を行った．今回は簡単な四足多関節剛体モデルに対して，紙面左側方向へ進む歩行動作が得られるか動作検証を行った．その結果得られた動作は後ろ足のみを交互に動かすものとなった．

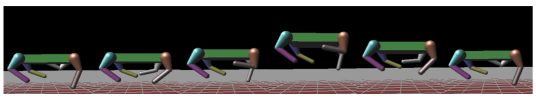

図 4　回転拘束を切った状態での動作生成

5.　問題点とその対応

四足多関節剛体モデルから想像される Walk, Trot, Canter, Gallop などの歩行とは図 4 の動作は異なる．この原因は現時点で，

1. GA の世代数の不足
2. 回転軸の拘束による影響

が考えられる．

まず，GA の世代数に対する問題だが，物理シミュレータを用いると強化学習に掛かる計算時間が長くなり，現実的な時間で十分に GA の世代を進めることができない．そのため，GA を繰り返せば動物の歩行と似た動作が出てくるのかそれともこの枠組みでは出てこないのかという判断が現時点ではつかない．

これに対して，一度行った状態間の遷移についてその時の強化学習の報酬と物理シミュレーション後の状態を保存しておき，再度その状態間の遷移が選択された時には保存した情報を用いてシミュレーションを省略することで強化学習の高速化を図りたいと考えている．

二つ目の回転軸の拘束による影響であるが，本システムの状態は各関節の角度および角速度であるため，身体全体の向きを制御することができない．そのため，体幹を回転しないように拘束することで学習を行っている．何が起きても体幹の回転しない環境は現実の環境とは異なるため，異なる運動が獲得されていると考えられる．

これに対して，状態空間を各関節の角度と角速度にすることを止め，体幹に対する状態空間を導入することで体幹の回転拘束を解いた空間で学習することにより解決したいと考えている．

6.　まとめ

今回までにシステム全体を構築し，簡単な多関節剛体モデルに対して動作検証を行った．その結果，体幹の回転拘束が存在する環境下で前進運動を獲得したが，現実世界の生物の運動とは異なる結果となった．これに対して，強化学習の高速化と状態定義の改良により解決を図りたいと考えている．

参考文献

[1] Chris Hecker, Bernd Raabe, Ryan W. Enslow, John DeWeese, Jordan Maynard, and Kees van Prooijen. Real-time motion retargeting to highly varied user-created morphologies. *ACM Trans. Graph.*, Vol. 27, No. 3, pp. 1–11, 2008.

[2] 伊藤一之, 松野文敏, 五福明夫. 強化学習による冗長ボットの自律制御に関する研究　—身体像を考慮した強化学習—. 日本ロボット学会誌, Vol. 22, No. 5, 2004.