

トルク変化最小軌道に基づくキャラクターの自然な動作の自動生成

佐藤 裕仁^{1,a)} 三武 裕玄¹ 長谷川 晶一¹

概要: 物理エンジンの導入やオープンワールドゲームなどの高い自由度を持つゲーム世界において、キャラクターは多様な力学的制約を受けつつ自在に行動することが望ましい。そのために、物理シミュレータによって動作をその場で自動生成する手法が提案されているが、自然さが十分ではない。本研究では、キャラクターの姿勢変化時に、反復計算によりトルク変化最小軌道でリアルタイムに生成し適用することで、多様なキャラクター動作を自動生成する手法を提案する。

Automatic generation of character's natural movement based on minimum torque change trajectory

HIROHITO SATOH^{1,a)} HIRONORI MITAKE¹ SHOICHI HASEGAWA¹

Abstract: In the virtual world with high degree of freedom such as Open World Game or game introduces physics engines, it is desirable that virtual characters act freely while being subjected to various physical limitation. For satisfy that purpose, some automatic motion generation method using physics engine has been proposed, but they are not natural enough. In this research, we propose a method to automatically generate various character movements by generating and applying minimum torque change trajectory by iterative calculation in real time when character pose changes.

1. はじめに

物理エンジンの導入やオープンワールドのような広大な環境の構築はゲームに高い自由度を与えた。また、ユーザーからの入力形式も、単なるボタン操作から視線検知、モーショントラッキングといったものが追加され、仮想空間とのインタラクションが多様化した。このような多様に変化する環境においては、キャラクターも多種多様な行動を行うことが望ましい。しかし、キャラクターモーションの主要手段である状態遷移によるキーフレームアニメーションの再生では、あらかじめ決められた状態のみを扱うため、多様さには限界がある。そのため、機械学習や物理計算によるキャラクター動作の自動生成が注目されている。教師あり学習では、実際の人間の動作をキャプチャし、教

師データとして与えることで安定な動作生成を実現している。しかし、求められたタスクごとに教師となるデータを用意しなければならず、あらゆる動作に対応するのは難しい。強化学習や遺伝的アルゴリズムなどの、事前のデータ準備なしでの手法もあるが、長い計算時間を要する。

本研究では、人間の到達運動の軌道モデルであるトルク変化最小軌道に基づいたキャラクターの姿勢変化軌道の自動生成を提案する。実時間での導出が困難なトルク変化最小軌道を、反復計算により短時間で導出し、経由点指定によるキャラクター動作軌道生成に利用することで、多様かつ自然なキャラクター動作を実現する。

2. 関連研究

2.1 キャラクターモーションの自動生成

Thomas ら [1] は、共分散行列適応進化戦略 (CMA-ES) を用いて、筋肉ベースでの2足歩行動作の自動生成を行った。様々な身体構成のキャラクターに対して、指定した速

¹ 東京工業大学
Tokyo Institute of Technology, 4259 Nagatsuta-cho Midori-ku Yokohama-shi Kanagawa 226-8503, Japan
^{a)} satoh.h.aa@m.titech.ac.jp

度での最適化した歩行動作を事前モーションデータを用いずに学習する。また, Martin ら [2] は, 身体重心や角運動量, 末端の位置といった高次の特徴から, 歩行やジャンプ, バランス維持といった動作のリアルタイム生成を行った。物理シミュレータの各ステップにおけるキャラクターの目標姿勢を計算し, リアルタイムでの動作が可能である。キャラクターの手先運動についての研究として, Yeo らの研究 [3] があげられる。この研究では, 飛んでくるボールをキャッチするような素早い手先運動について, 実際の人間の動作を躍度最小軌道に基づいて解析し, 再現を行っている。

2.2 人間の運動規範のモデル化

人間の運動規範を知るために, 運動軌道のモデル化が試みられてきた。単純な観察動作として, 手先を目標位置まで移動させる手先到達運動が調べられ, 躍度最小軌道, トルク変化最小軌道, 指令トルク変化最小軌道などの軌道モデルが提案された。躍度最小モデルは, 運動空間における5次元多項式で与えられ, 運動開始時の手先位置や速度といった情報を指定することで, 軌道が一意に定まる。このモデルは水平面内での手先軌道をよく近似しているが, 重力などの外部環境の影響を考慮していないため, 鉛直方向への移動などでは不自然な軌道をとることがある。トルク変化最小軌道は躍度最小軌道に比べ, 動的な軌道が計算が行われ, その軌道再現性は躍度最小軌道より優れている。トルク変化最小軌道は下記の評価関数 (1) を最小化する軌道である。

$$C_{\tau} = \frac{1}{2} \int_0^{t_f} \sum_{i=1}^n \left(\frac{d\tau_i}{dt} \right)^2 dt \quad (1)$$

ここで, t_f は運動時間, n は運動にかかわる関節数, τ_i は i 番目の関節の発揮トルクである。この最適化問題を厳密にリアルタイムで解くのは困難である。そこで川人はリアルタイムでの軌道導出モデルとして順逆緩和型神経回路モデル [4] を提案した。このモデルは, 軌道から運動指令を予測生成する逆ダイナミクスモデルと, 運動指令から制御対象の軌道を予測する順ダイナミクスモデルを組み合わせて, 相互フィードバックの繰り返しにより軌道を最適化する。本研究の手法はこの神経回路モデルをベースにしている。

3. 提案手法

提案手法の全体図を図 1 に示す。計算に用いる物理シミュレータには解析法によるシミュレータである Springhead [5] を使用し, キャラクターには剛体多関節モデルを用いる。入力からベースとなる軌道を生成し, 制御対象を実際にその軌道で目標点まで移動させる物理シミュレーションを行い, 運動に関係した関節で発揮していたトルクの時系列データを保存する。保存されたトルクデータにローパ

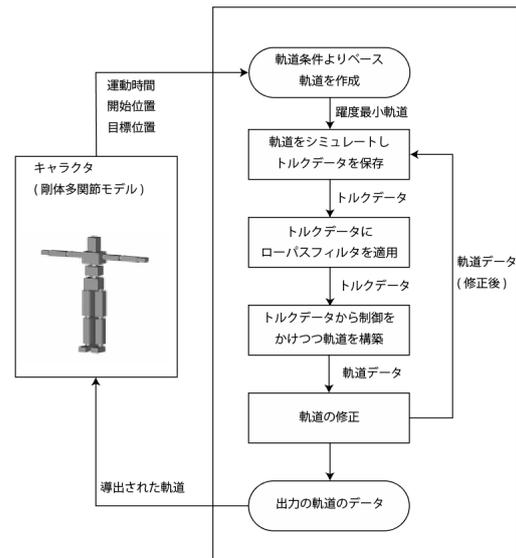


図 1: 提案手法

スフィルタを適用することで, トルク変化を滑らかにした平滑トルクデータを生成する。ベース軌道に沿う弱い PD 制御を行いつつ, 平滑トルクデータを追加の関節トルクとして与える。この時, 到達条件を無視してトルクを平滑化しているため, 操作対象が目標点に到達しない。そのため, 先ほど得られた軌道の終端点を目標点と一致するように軌道全体に修正を加える。この修正軌道を新たなベース軌道として, 上記の修正過程を繰り返す。設定した反復回数の修正過程ののち, 最後に修正された軌道データをキャラクターの到達運動軌道として出力する。出力は到達運動中の各ステップにおける制御対象部位の位置である。以下では修正の各過程を説明する。

3.1 入力

システムへの入力は計算開始時の制御対象の位置 P_s , 目標点位置 P_f , 運動時間 t_f である。これらの入力パラメータを満足する躍度最小軌道を式 2 により生成する。

$$P(t) = P_s + (P_s - P_f) (15\tau^4 - 6\tau^5 - 10\tau^3) \quad (2)$$

$\tau = \frac{t}{t_f}$ は運動時間で正規化した時間 ($0 \leq \tau \leq 1$) であり, $P(y)$ は時刻 t における制御対象の位置を表す。

3.2 軌道データからトルクデータの生成

与えられた軌道データをもとに操作対象部位を IK により動かし, その際の各関節の発揮トルクの時系列データを保存する。使用する軌道データは, 初回計算時は 3.1 の躍度最小軌道を与え, 以降は反復 i 回目における 3.4 の出力である軌道データを $i+1$ 回目の入力として与える。

今回用いた物理シミュレータ Springhead においては, 拘束力計算の一部として関節駆動トルクを求めることができる。関節駆動のモータはバネ・ダンパモデルにより駆動ト

トルクを発生させる。Springhead は、拘束条件を線形相補性問題 (LCP) に定式化し、拘束力を計算している。このとき、バネ・ダンパモデルも LCP の枠組みに組み込むことで、目標変位を実現する関節駆動トルクを拘束力として求めている [6][7]。

3.3 トルクから軌道データの生成

3.2 で記録したトルクデータをローパスフィルタの適用により変化を滑らかにする。ローパスフィルタには N 点移動平均フィルタを用いた。そして、平滑後トルクデータをもとにした軌道データの構築を行う。3.2 での軌道シミュレート時よりも小さいバネ定数及びダンパ定数を各関節のモータに設定し、弱い PD 制御を行いつつ、さらに平滑トルクデータを追加の関節駆動トルクとして付加する。この状態で物理シミュレーションを実行することにより、関節を駆動させ、動力学的に制御対象の軌道データを得る。こうして得た軌道は以前よりもトルク変化が滑らかになっていることが期待される。

3.4 軌道データの修正

3.3 で生成した軌道データは、制御を弱めていること、トルクデータにローパスフィルタを適用していることから本来の目標位置とは異なる位置で動作終了している。そのため、得られている軌道データに修正を加えることで、軌道終端点と目標点を一致させる。そのような修正方法は多数考えられるが、今回は、実装の容易さから以下のような修正方法を用いている。

- (1) 運動開始点と軌道終端点を結ぶ直線と、運動開始点と目標点を結ぶ直線が同一直線となるように軌道全体を回転させる
- (2) 軌道終端点と目標点が重なるように、軌道全体を引き延ばす

こうして修正された軌道データが 3.2 における軌道データに渡される。もしくは、設定した修正回数に達していた場合は、キャラクターの制御対象部位の軌道データとして出力される。

3.5 出力

軌道計算システムの出力は、物理シミュレーションの各ステップにおける制御対象の目標位置 $P_i (i = 0, 1, \dots, t_f)$ である。キャラクターはそれに合わせて、制御対象の目標位置を更新しシミュレーションを実行することで、動作を生成する。

4. 実験

提案手法によってどのような軌道が生成され、どの程度の効果が見込めるかを知るために、簡易的なモデルを用いて、軌道生成の実験を行った。

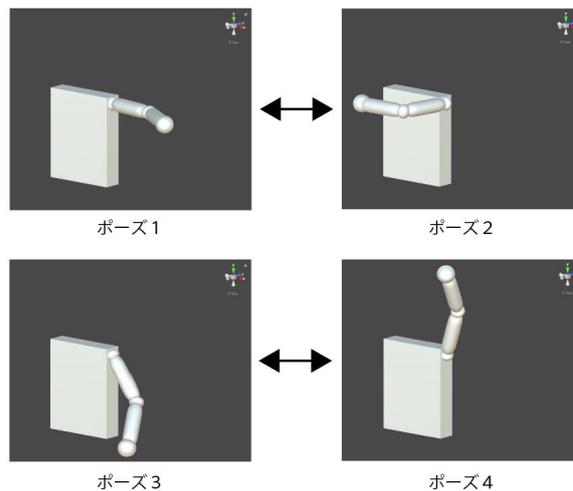


図 2: 実験に用いたポーズ

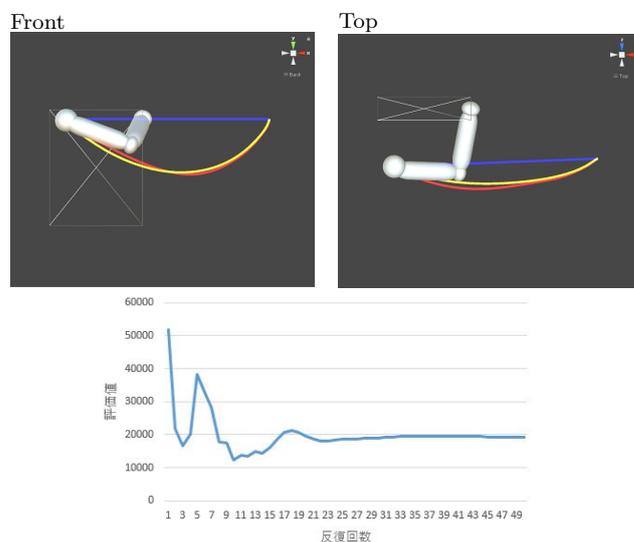


図 3: ポーズ 1 からポーズ 2 への軌道

4.1 実験方法

人間の腕と胴体を模した 4 剛体 3 関節からなる簡易的な剛体多関節モデルを用いた。各部剛体の質量は人体の構成部位質量に基づいて決定した。実験では図 2 に示す 4 種の姿勢を用い、ポーズ 1 とポーズ 2 の間での双方向の軌道、及び、ポーズ 3 とポーズ 4 の間での双方向の軌道を計算した。なお、反復回数は 50 回とし、N 点移動平均フィルタの N は 3、姿勢変化の動作時間は 1 秒と設定した。

4.2 結果

図 3 はポーズ 1 からポーズ 2 への軌道であり、図 4 はポーズ 2 からポーズ 1 への軌道である。青色が躍度最小軌道、黄色が 50 回目の反復計算により得た軌道、赤色が反復計算中に出現した軌道の中で最もトルク変化の総和が小さかった軌道である。躍度最小軌道は運動開始時の手先位置と運動の目標位置を結ぶ直線となり、図 3 と図 4 で全く同

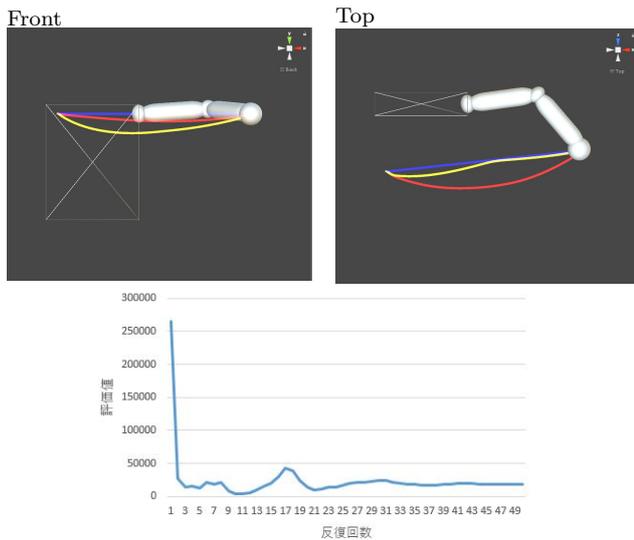


図 4: ポーズ 2 からポーズ 1 への軌道

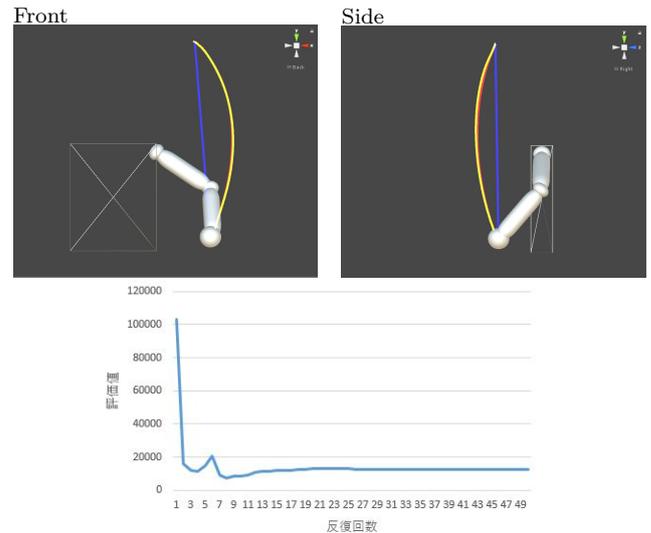


図 6: ポーズ 4 からポーズ 3 への軌道

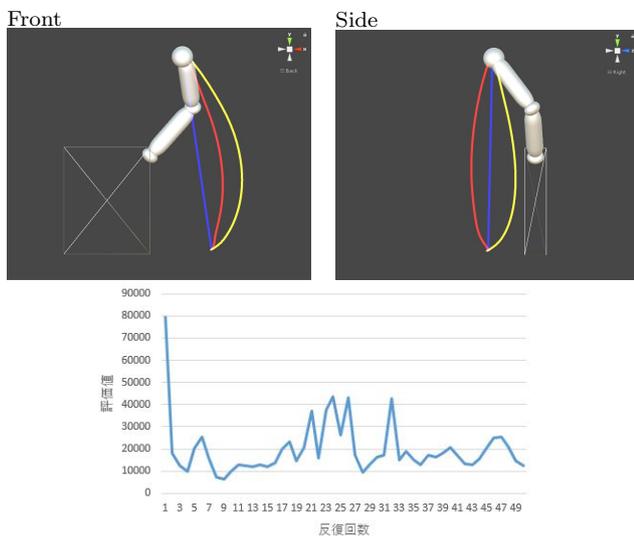


図 5: ポーズ 3 からポーズ 4 への軌道

じ軌道である。それに対し、ほかの 2 種の軌道は、滑らかな曲線を描いており、かつ遷移の方向により違う軌道が得られている。これは、運動開始時に関節で発揮されていたトルクという動的な条件が軌道計算に考慮された結果であると考えられる。また、軌道の下にグラフで示したものは、各反復回数において生成された軌道の評価関数の値を表示したものである。30 回程度で収束し、初期値である躍度最小軌道での値に比べ低い値をとっていることがわかる。

同様にポーズ 3 からポーズ 4 への軌道を示したものが図 5、ポーズ 4 からポーズ 3 へのものが図 6 である。こちらも、反復計算中に出現した軌道は遷移方向により異なっている。しかし、ポーズ 3 からポーズ 4 へ向かう軌道では、軌道が収束しなかった。今回の実験に用いた簡易的なモデルは、関節の可動域を考慮していない。そのため、実際の人間に比べ軌道の制約が少なく、これが軌道が収束しない一因と考えられる。

各軌道計算において 50 回の反復に要した計算時間は 500msec~519msec であった (CPU: Core i7-6700 3.40GHz, OS: Windows10 Enterprise 64bit, メモリ: 8.00Gbyte)。計算時間は反復回数と動作時間に比例し、今回の実験を鑑みるに反復回数は今回の設定値より小さくできそうである。リアルタイムとはいえないが、計算時間としては十分に短いため、バックグラウンドで計算を行うことで、動作目標決定から実際の動作までをスムーズに見せることができると考えている。

5. まとめと今後

反復計算によりトルク変化を小さくする軌道の生成手法を提案し、実際に提案した手法により軌道のトルク変化が減少することを示した。今後は、可動域制限のほか、関節の最大発揮可能トルクによる重みづけや人体関節の有する抵抗特性などを考慮に入れ、実際のキャラクターへの適用を行いたいと考えている。

参考文献

- [1] Thomas Geijtenbeek, Michiel van de Panne, A. Frank van der Stappen : Flexible Muscle-Based Locomotion for Bipedal Creatures, ACM Translation on Graphics Vol32 No6 Article 206, 2013
- [2] Martin de Lasa, Igor Mordatch, Aaron Hertzmann : Feature-Based Locomotion Controllers, ACM Transactions on Graphics(TOG) Vol29 Issue4 Article No.131, 2010
- [3] Sang Hoon Yeo, Martin Lesmana, Debanga R. Neog, Dinesh K. Pai : Eyecatch: Simulating Visuomotor Coordination for Object Interception, ACM Transactions on Graphics(TOG) Vol31 Issue4 Article No.42, 2012
- [4] 川人光男: 脳の計算理論, 産業図書 (1996)
- [5] 長谷川晶一, 三武裕玄, 田崎勇一: 動作行動開発のための物理エンジン Springhead, 日本ロボット学会誌 Vol.30 No.9, pp.841~848, 2012
- [6] Takashi Tokizaki, Yuuichi Tazaki, Hironori Mitake,

Shouichi Hasagawa : Pliant Motion: Integration of Virtual Trajectory Control into LCP Based Physics Engines, 2009

- [7] 田崎勇一, 長谷川晶一: 拘束法の動力学シミュレータのための安定なバネダンパモデル, IPSJ SIG Technical Report, 2006