

環境からの力学的な影響に応じたキャラクタアニメーションの生成

4 K-2

尾下 真樹

九州大学大学院システム情報科学府

牧之内 顕文

九州大学大学院システム情報科学府

1. はじめに

近年、コンピュータゲームや Virtual Reality などのアプリケーションにおいて、キャラクタの自然な動作をリアルタイムに生成する必要性が高まっている。現在、これらのアプリケーションでは、あらかじめ作成しておいた動作データを連続的に生成することによってこのような要求に対応している。しかし、このようなアプローチでは、キャラクタはあらかじめ決められた動作の繰り返ししかできないという問題点がある。特に、キャラクタに外部から力が加えられたり、他のオブジェクトと衝突したりといった、環境からの力学的な影響が加わった時には、それらの影響に応じた動作を動的に生成することが必要となる。

本稿では、このような環境からの力学的な影響に応じて、人間の自然な動作をリアルタイムに生成するための手法を提案する。本手法では、動作データを単純に再生するのではなく、与えられた動作データを目標動作として、目標動作に追従するようにキャラクタの関節角加速度を制御し、動力学シミュレーションによって実際の動作を生成する。基本的には目標動作をそのまま実行するように制御し、環境からの力学的な影響が加わる場合のみ、関節に加わる負荷を軽減したり、全身のバランスを保ったりするような制御を行う。その結果、下半身の負荷を減らすように腕を振ったり、バランスを保つように腰を動かしたり、といった動作が実現される。本アプローチによって、既存の動作データやその作成ノウハウをそのまま活用しつつ、環境との相互作用を含む自然な動作を動的に生成することが可能となる。

Generating Character Animation Considering Interaction with the Environments.

Masaki Oshita and Akifumi Makinouchi

Graduate School of Information Science and Electrical Engineering, Kyushu University

動力学シミュレーションによって人間の自然なアニメーションを生成しようという研究はこれまでも多く行われている[1]。しかし、これまでのアプローチでは、ロボット工学の制御手法を応用し、キャラクタの関節トルクを制御していた。そのため、安定な制御を行うためには目標動作や環境からの影響に応じてコントローラのパラメータを調整する必要があった。また、環境からの影響に応じた能動的な制御については考慮されていなかった。提案手法では、角加速度空間で制御を行うというアプローチによって、いかなる状況でも安定して目標動作に追従しつつ、環境からの影響に応じて動的な制御を行うことを可能とした。

2. 多関節体の運動制御アルゴリズム

本アルゴリズムは、各時刻において、与えられた目標動作と動力学計算にもとづいて、キャラクタの各関節の角加速度を計算する。目標動作は、各関節の角度の時間関数として与えられる。本アルゴリズムでは、制御全体を追従制御と動的制御の2段階に分ける。

2.1. 追従制御

追従制御では、目標動作に単純に追従するような各関節の角加速度を計算する。ここでは、関節相互の影響は考慮せず、動力学計算も使用しない。それぞれの関節ごとに、現在の関節角度・角速度と、目標動作の関節角度関数における次の極大・極小点の角度・角速度から、2点の間の軌道を3次関数で表し、その関数の2回微分により、出力角加速度を決定する。本手法により、目標動作から離れている場合でも、目標動作に対してなめらかに収束するような制御が行える。

2.2. 動的制御

動的制御では、追従制御で計算された角加速度を、動力学計算を用いて修正する。具体的には、関節負荷減少とバランス保持の2種類の制御を行う。ある関節に大きな負荷が加わる時には、全身の慣性モーメント

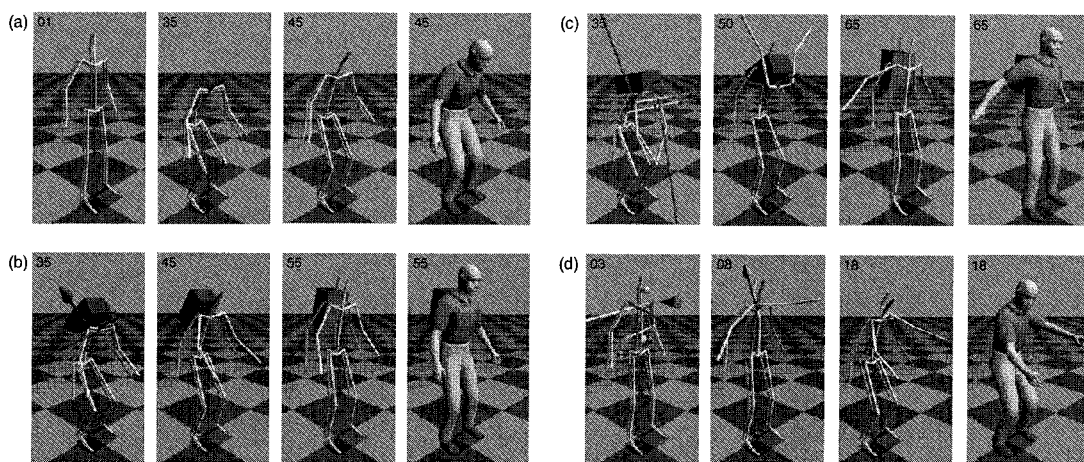


図1: スクワット動作への適用例。(a) 環境からの影響を追加しない場合の動作。(b) 重さ 5kg の荷物を背負った場合の動作。(c) 重さ 8kg の荷物を背負った場合の動作。(d) 3 フレーム目に左肩に衝撃を与えた場合の動作。

行列を考慮し、その関節負荷を減少するように他の関節の角加速度を制御する。また、バランス制御のため、全身から地面へのゼロモーメント点 (ZMP) [2] が支点面の外部にある場合には、同じく、ZMP を支点面の内部に移動するような制御を行う。

ある関節の角加速度は、他の関節の負荷や全身のバランスに影響を及ぼすため、動的制御では、関節相互の影響を考慮する必要がある。しかし、全関節の影響を考慮して最適化計算を行うと、多くの計算時間が必要となる。そこで、本手法では、人間の体を腕・背中・下半身の3つの部位に分類し、それぞれの部位を重要な最小限の自由度によって順番に制御することとした。

3. 実験結果

図1は、提案手法をスクワット動作に対して適用した結果である。今回の実験では、36自由度の骨格モデルを使用し、秒間30フレームのアニメーションを生成した。各画像の左上の数字はフレーム番号を表す。各画像において、灰色のフィギュアが目標動作、白のフィギュアは制御結果の動作を表している。また、関節から出ている矢印は、その関節の動的制御の回転方向と大きさを表している。(a)環境からの特別な影響がない場合には、追従制御のみが行われ、目標動作がほぼそのまま実現された。また、(b)背中に5kgの荷物を背負った状態では、腰に加わる負荷を減少するために

腕を前方に振るような制御が行われた。同じく、(c)8kgの荷物を背負った状態では、より大きく腕を振りながら、さらに、腕や腰の動きによってバランスを保つような制御が行われた。そして、(d)動作の途中で衝撃を与えられた時には、腕を振ることで全身のバランスを保ちつつ、筋力の許す最大限の加速度でもとの動作に追従するような制御が行われた。

1フレームの制御に必要なとした計算時間は、平均で1.2ミリ秒、追従制御の場合のみで0.7ミリ秒、最も負荷が大きい場合で3.1ミリ秒であった。結果として、レンダリングやシミュレーションの時間を含め、アニメーションをリアルタイムに生成することができた。

4. まとめ

本稿では、あらかじめ作成された動作データをもとに、環境からの力学的な影響に応じてキャラクタを制御する手法を提案した。本手法は、コンピュータゲームなどへの応用に非常に有効であると考えられる。

参考文献

- [1] Jessica K. Hodgins, Wayne L. Wooten, David. C. Brogan, James F. O'Brien, "Animating Human Athletes", SIGGRAPH '95 Proceedings, pp. 71-78, 1995.
- [2] Seyoon Tak, Oh-young Song, Hyeong-Seok Ko, "Motion Balance Filtering", Eurographics 2000, Computer Graphics Forum, Vol. 19, No. 3, pp. 437-446, 2000.