

環境からの力学的な影響を考慮した リアルタイムなキャラクタアニメーションの生成手法

尾下 真樹 牧之内 顕文

九州大学大学院 システム情報科学研究院 知能システム学部門
〒812-8581 福岡市東区箱崎 6-10-1
E-mail: moshita@db.is.kyushu-u.ac.jp, akifumi@is.kyushu-u.ac.jp

本論文では、重い荷物を持ったり、他の物体と衝突したりといった、環境からの力学的な影響に応じて、リアルタイムに変化するキャラクタアニメーションの生成手法を提案する。本手法では、ユーザによって与えられた目標動作に追従するように関節の角加速度を制御し、動力学シミュレーションによってアニメーションを生成する。本手法は、キャラクタの筋力の範囲内で、関節負荷を減少し、全身のバランスを保つような制御を行う。我々は、人間の運動制御で主に活用されている関節やその動かし方に注目し、人間の体を腕、背中、足の部位に分け、動力学計算を用いてそれぞれの部位を効率的に制御するための手法を開発した。

キーワード：キャラクタアニメーション、動力学シミュレーション、運動制御アルゴリズム

A Method for Generating Real-Time Character Animations in Response to Physical Influences from the Environments

Masaki OSHITA and Akifumi MAKINOUCHI

Department of Intelligent Systems,
Graduate School of Information Science and Electrical Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-Ku Fukuoka-Shi, 812-8581, Japan
E-mail: moshita@db.is.kyushu-u.ac.jp, akifumi@is.kyushu-u.ac.jp

This paper presents a method for generating real-time character animations in response to physical influences from the environments such as holding a heavy load or colliding with other objects. Angular accelerations of the character are controlled to track a desired motion given by a user. Dynamic simulation then generates resulting animations. The character is controlled as to reduce the joint stresses and to keep its balance within the available muscle strength of the character. We analyzed human being's motions and developed control schemes specialized for arms, back, and feet.

Keywords: Character Animation, Dynamic Simulation, Motion Control Algorithm

1 はじめに

本論文では、重い荷物を持ったり、他の物体と衝突したりといった、環境からの力学的な影響に応じて、リアルタイムに変化するキャラクタアニメーションの生成手法を提案する。近年、コンピュータゲームや Virtual Reality などのオンラインアプリケーションにおいて、このように周囲の状況に応じて動的に変化する動作の生成が重要な課題となっている。例えば、現在のコンピュータゲームでは、基本的にあらかじめ作成された動作を再生することしかできないため、他の物体との衝突した時などに、キャラクタの動作が不自然になってしまうという問題がある。

本手法では、ユーザから与えられた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アニメーションを生成する。環境からの力学的な影響によって一部の関節に大きな負荷が加わる場合には、関節相互の影響を考慮しながら、その関節の負荷を減少するような制御を行う。また、バランスを失って転倒しそうな場合は、全身のバランスを保つための制御を行う。これらの制御は、キャラクタの筋力の範囲内で行われる。我々は、実際の人間の運動制御で主に活用されている関節やその動かし方に注目し、動力学計算を用いて効率的な制御を行うための手法を開発した。本手法を用いることにより、例えば、重い荷物を持ったり、他の物体と衝突したりした時に、腕を振ることによって下半身に加わる負荷を軽減したり、腰の動きによってバランスを保ったりといった、状況に応じて動的に変化する動作を実現できる。

動力学計算を用いたキャラクタアニメーションの生成手法に関しては、これまでにさまざまな研究が行われている[7][2]。しかし、それら従来の手法の多くは、手作業で与えたキーフレームや、プロシージャルに生成された単調な動作をもとに、動力学計算を用いて力学的に自然な動作を生成することを目標としている。そのため、既存のモーションデータを有効に利用することは難しく、一般のアプリケーションで十分に活用されているとは言えない。それに対して、本手法では、すでに力学的に十分自然な目標動作が与えられているも

のと仮定し、基本的には与えられた目標動作をほとんどそのまま追従する。そして、環境からの力学的な影響が加えられた時のみ、動力学を考慮した制御を行い、自然な動作を生成する。

2 動力学シミュレーション

本節では、提案手法について述べる前に、本手法で用いるキャラクタの骨格・筋力モデルやシステム構成について説明する。

2.1 骨格・筋力モデル

動力学計算を適用するため、キャラクタの骨格モデルを、体節と関節の組み合わせによる多関節体として表現する(図1)。各体節は、質量や慣性モーメントなどの力学的な属性を持つ。各関節は、1~3自由度の回転関節によって構成され、各回転軸には角度の可動範囲が設定されている。現在の実験では、18体節、19関節、合計39自由度の骨格モデルを使用している。また、筋力モデルとして、それぞれの回転関節の出力可能な最大・最小トルクを、その関節の角度・各速度の関数によって表す[5]。現在は、手作業で適当に与えた関数を使用している。

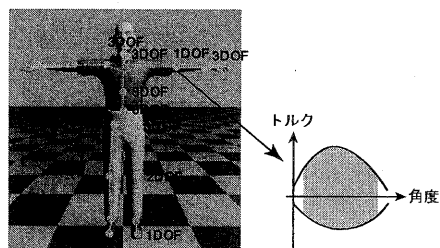


図1 骨格・筋力モデル

2.2 システム構成

図2に、提案システムの構成図を示す。従来の動力学シミュレーションの手法とは異なり、本手法では、キャラクタの関節トルクではなく関節角加速度を直接制御する。このことにより、安定した制御が実現できる。コントローラは、現在の状態と目標動作をもとに、キャラクタの各関節の出力角加速度を決定する。目標動作は、各関節角度の時間関数としてユーザから与えられる。シミュレータは、コントローラの出力した角加速度をもとに、キャラクタの運動を計算する。それぞれの関節ごとに、積分計算によって角速度・角度が更新される。また、キャラクタがバランスを失った

時の支点を中心とする回転加速度を、逆動力学計算の応用により求める[4]。

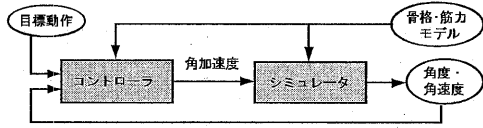


図2 システム構成図

3 関節角加速度の制御アルゴリズム

本節では、キャラクタの関節角加速度を制御するためのアルゴリズムを、その計算手順に従って順番に述べる。

3.1 初期角加速度の計算

まず、最初のステップとして、現在のキャラクタの状態と与えられた目標動作から、目標動作を実現するための出力角加速度の初期値を計算する。ここでは関節相互の影響は考慮せず、それぞれの関節ごとに、関節の現在の角度・角速度と、目標動作から出力角加速度を計算する。

従来手法では、このような制御を目的として、PD制御が一般的に用いられている[7][2]。しかし、PD手法では、動作に応じてパラメタを調整する必要があり、汎用的なコントローラを実現することは難しい。安定したフォワード制御を実現するためには、PD制御のように目標動作の次ステップの状態だけを考慮するのではなく、先の軌道を予測した制御を行う必要があると考えられる。

そこで、本手法では、補間曲線の一種である Ferguson 曲線[1]を用いて出力角加速度を計算する。B-スプライン曲線が4つの制御点を用いて1つの曲線区間を定義するのに対し、Ferguson 曲線は、2つの点とその速度によって曲線区間を定義する。本手法では、目標軌道から次の極大・極小点を探索し、その点での角度・角速度 (Q_1, \dot{Q}_1) と現在の角度・角速度 (Q_0, \dot{Q}_0) から、2点間の軌道を Ferguson 曲線で近似し、出力角加速度を決

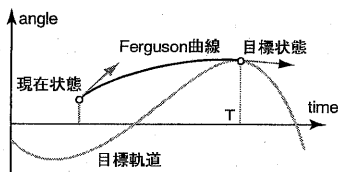


図3 Ferguson 曲線による軌道

定する(図3)。導出の詳細は省略するが、出力角加速度は、最終的に以下の式によって求まる。

$$\ddot{\theta} = \{6(Q_1 - Q_0) - 4\dot{Q}_0 - 2\dot{Q}_1\} \div T^2 \quad (1)$$

本手法により、環境からの力学的な影響によって目標軌道から離れてしまった状態でも、目標動作にゆるやかに収束するような適当な角加速度を計算することができる。

3.2 環境からの力学的な影響に応じた制御

次のステップでは、環境からの力学的な影響に応じて角加速度を修正する。具体的には、関節相互の影響を考慮した負荷減少と、バランス制御の2つの制御を行う。以下に、これらの制御を実現するための動力学計算について述べる。

まず、関節相互の影響を考慮した負荷減少を行うために、慣性モーメント行列を用いる。角度 θ 、角速度 $\dot{\theta}$ の状態において、角加速度 $\ddot{\theta}$ の運動を行うための必要トルク τ は、以下の運動方程式によって表される。

$$\tau = H(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) + F(\theta) \quad (2)$$

$\theta, \dot{\theta}, \ddot{\theta}, \tau$ は n 次元のベクトルであり、各行が各回転関節に対応する。 $H(\theta)$ は $n \times n$ の慣性モーメント行列、 $C(\theta, \dot{\theta})$ は遠心力・コリオリ力による必要トルクへの影響、 $G(\theta) + F(\theta)$ は重力・外力による影響を表している。慣性モーメント行列の各列・行、及び、その他の各項は、逆動力学計算を応用することで、それぞれ $O(n)$ で計算できる[8]。ある関節において、初期角加速度を実現するための必要トルクがその関節の筋力の範囲を超えている場合は、慣性モーメント行列を考慮しながら、その必要トルクを減少するように全関節の角加速度を修正する。

次に、キャラクタのバランスを制御するために、Zero-Moment Point (ZMP) という概念を用いる[6][4][9]。ZMP とは全身から地面へのモーメントが0となる点である。ZMP が支点面(地面と足の接触面、両足で立っている場合は両足の接触面の凸包)の内部に存在しない状態ではバランスが取れておらず、そのままではキャラクタは転倒してしまう。そこで、ZMP を支点面の内部に移動するように角加速度を修正する。ZMP の位置は、

地面が水平であると仮定すると各体節の位置 (x_i, y_i, z_i) と加速度 $(\ddot{x}_i, \ddot{y}_i, \ddot{z}_i)$ 、重力加速度 g から求めることができる[6]。

$$\begin{aligned} ZMP_x &= \frac{\sum m_i x_i (\ddot{y}_i - g) - \sum m_i y_i \ddot{x}_i}{\sum m_i (\ddot{y}_i - g)} \\ ZMP_y &= \frac{\sum m_i x_i (\ddot{y}_i - g) - \sum m_i y_i \ddot{x}_i}{\sum m_i (\ddot{y}_i - g)} \end{aligned} \quad (3)$$

3.1 節の方法に従って初期角加速度を計算したら、その角加速度を出力した時の必要トルクと ZMP の位置を計算する。ここで、必要トルクが筋力の範囲内にあり、ZMP が支点面の内部にある場合には、初期角加速度をそのまま出力する。この場合は、目標動作がほぼそのまま実現される。筋力や ZMP の条件が満たれない場合は、以下のアルゴリズムに従って、関節角加速度を修正する。

3.3 関節角加速度の制御方法

角加速度の修正値を計算するための方法としては、慣性モーメント行列や関節角加速度と ZMP の関係などから方程式を立て、最適化問題として数値的に解く手法が考えられる[4]。しかし、このような手法では、冗長な関節を含むため計算に多くの時間を必要とし、また、最適化関数の選択や数値演算の安定性などの問題により、必ずしも自然な動作を実現することは難しい。そこで、我々は、実際の人間の運動制御で主に活用されている関節やその動かし方に注目し、最小限のパラメタを用いることで、目標とするような制御を実現することとした。本手法では、人間の体を腕、腰、足の3つの部位に分け、それぞれの部位ごとに制御を行う(図4)。まず、それぞれの部位の関節負荷を計算し、筋力に余裕のある部位のうち、腕、背中、足の順番で、他の関節負荷を減少したり、ZMP を支点面内に移動したりするための角加速度の修正値を計算する。それぞれの部位の修正値を計算する度に、関節負荷と ZMP の位置を再計算し、条件が満たされなければ、腕、腰、足の順番で制御を行う。

以下では、それぞれの部位ごとに制御方法を述べる。なお、本論文の制御手法は、基本的にキャラクターが両足で直立している状態を対象としている。しかし、他の姿勢の場合でも、基本的には同

様の制御方法が適用できると考えている。

3.3.1 腕の制御

人間の腕の運動に注目すると、肩の回転が他の関節や ZMP への影響が最も大きく、重要であると考えられる。そこで、腕を制御するパラメタとして、両肩の x 軸と z 軸を回転軸とする回転加速度を用いる(図5)。他の関節に体の前後方向の負荷が加わっている場合、 x 軸を回転軸として両腕を同一方向に振ることにより、その負荷を軽減する。また、体に左右方向の負荷が加わっている場合は、 z 軸を回転軸として右腕が左腕を外側に振ることによって、その負荷を軽減する。一方、 y 軸を回転軸とする腕の回転は他の関節への慣性モーメントの影響が小さいため、 y 軸を回転軸とする負荷(ねじる方向の負荷)は、 x 軸を回転軸として両腕を逆方向に振ることによって軽減する。

両肩の x 軸と z 軸を回転軸とする回転加速度を計算するため、まず、負荷の大きいそれぞれの関節ごとに、その関節の各回転軸にそった必要トルクの修正量を合計し、負荷ベクトル (T_x, T_y, T_z) を計算する。そして、負荷の方向によって腕の回転軸の使い分けのため、負荷ベクトルを両腕の基準座標系(胸の座標系)に回転する。最後に、角加速度の修正量 $\Delta\ddot{\theta}_{rx}, \Delta\ddot{\theta}_{rz}, \Delta\ddot{\theta}_{lx}, \Delta\ddot{\theta}_{lz}$ は、両肩の x 軸と z 軸を回転軸として回転させた時のその関節の基準座標系での各軸への慣性モーメント $H_{rx,x}, H_{rx,y}, H_{rz,z}, H_{lx,x}, H_{lx,y}, H_{lz,z}$ から、以下の式で計算する。

$$\begin{aligned} \Delta\ddot{\theta}_{rx} &= \frac{H_{lx,y} \cdot T_x - H_{lx,x} \cdot T_y}{H_{rx,x} \cdot H_{lx,y} - H_{rx,y} \cdot H_{lx,x}} \\ \Delta\ddot{\theta}_{lx} &= \frac{H_{rx,y} \cdot T_x - H_{rx,x} \cdot T_y}{H_{rx,x} \cdot H_{lx,y} - H_{rx,y} \cdot H_{lx,x}} \\ \Delta\ddot{\theta}_{rz} &= \frac{T_x}{H_{rz,z}} (T_x > 0), \quad \Delta\ddot{\theta}_{rz} = \frac{T_x}{H_{rz,z}} (T_x < 0) \end{aligned} \quad (4)$$

次に、両肩の x 軸と z 軸を回転軸とする回転加速度の変化量に応じた ZMP の移動量を、各腕の質量と肩から腕の重心までのベクトルから求め、ZMP を支点面内に移動するための角加速度を計算する。

一般的に、環境からの力学的な影響によって、複数の関節に大きな負荷が加わったり、ZMP が支点面から外れたりする時、全ての負荷は同一方



図4 体の各部位の分類

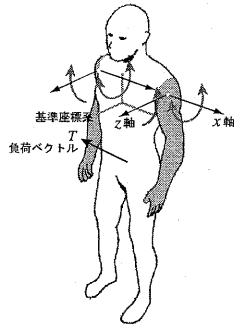


図5 両腕の制御

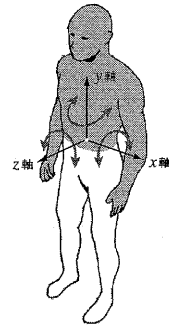


図6 背中中の制御

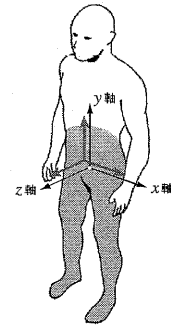


図7 両足の制御

向である場合が多いと考えられる。そこで、本手法では、それぞれを制御するための角加速度を独立に計算し、最大のものを出力角加速度とする。

また、一方の肩に大きな負荷が加わっており、片方の肩だけで制御を行う場合も、基本的には同じ方法で角加速度を計算する。ただし、片方の肩だけでは自由度が制限されるため、目標とする制御を実現できない可能性がある。その時には、背中や足の制御によって補う。

3.3.2 背中中の制御

背中中の制御は、腰の3軸の回転角加速度をパラメータとして行う(図6)。腰を基準座標系として、腕の制御と同様の方法で、負荷ベクトル、慣性モーメントを計算し、各軸を回転軸とする角加速度を決定する。また、ZMP に関して、上体の質量と重心を用いて、腕と同様の方法で計算する。

3.3.3 足の制御

両足で立っている状態では、足と地面の間の拘束条件を考慮する必要があり、両足の全ての関節を同期して制御しなければならない。そこで、足の関節角加速度を直接制御するのではなく、腰の並進加速度をパラメータとして、両足の関節角加速度を間接的に制御する(図7)。足の各関節の角加速度は、腰の並進加速度から Inverse Kinematics により計算する。本手法では、計算を効率的に行うため、擬似逆行列などの数値的解法ではなく、腿の y 軸方向の回転が最小になるような動作を仮定した分析的解法を用いる[3]。

また、このように、足の全ての関節を同時に制御する場合、上体への慣性モーメントによる影響は比較的小さくなる。そこで、足の制御は ZMP の移動のみを目的として行うこととする。腕や背

中の制御時と同様に、上半身の質量と重心を用いて ZMP の位置変化を計算する。

なお、片足で立っている状態では、浮いている方の足を腕の場合と同様に動かすことで、より大きな制御が可能である。しかし、このような制御を実現するためには、角加速度空間だけでなく、関節角度空間で制御を行うことが必要になると考えられるため、本論文の範囲では扱わない。

3.4 筋力による制御の限界

前節で述べた腕・背中・足のそれぞれの制御は、各関節の有効な筋力の範囲内で行う必要がある。例えば、腰の負荷を減少するために肩の角加速度を修正する時、逆に、肩の必要トルクが筋力の範囲を超えてしまうことは許されない。そこで、3.3 節のアルゴリズムによる角加速度の計算後、あらためて必要トルクを計算し、筋力の範囲を超えていれば角加速度の修正量を減少させる。腕・背中・足による制御を有効な筋力の限界まで行っても、大きな関節負荷を補償することができない場合は、その関節の角加速度を筋力の範囲に修正する。

この結果、例えば、環境からの力学的な影響をキャラクタの筋力の範囲では補償することができない時には、筋力の範囲内でできる限りの制御を行いながらも、最終的には目標動作を実現できず、転倒してしまったりするような自然な動作が生成される。

4 実験

本節では、本論文で提案した手法を用いて実際に作成したアニメーションの例を示す。今回の実験では、目標動作として、スプライン関数によるキーフレーム動作を与えている。ステップ間隔は 10 ミリ秒に固定して、シミュレーションを実行し

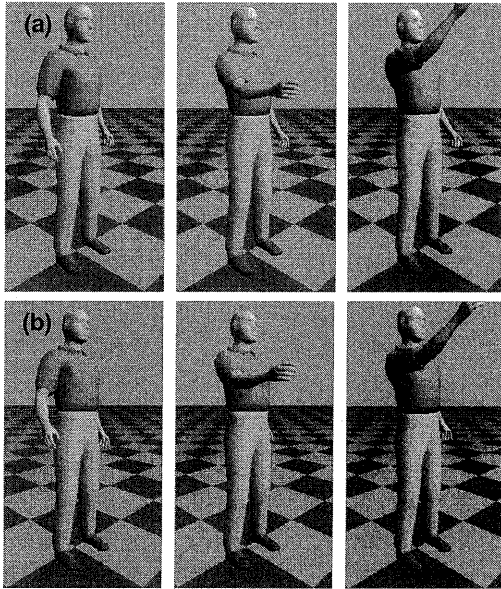


図8 持ち上げる動作 (a)荷重 1kg (b)荷重 3kg

た。1ステップの制御に必要なとする時間は、PentiumIII 800MHz のパソコンで3ミリ秒程度であり、動作はリアルタイムに生成される。

図8は、荷物を持ち上げる動作を、重さを変えて実行した適用例である。荷重が1kgの時は、関節負荷は筋力の範囲内であるため、もとの動作がほぼそのまま追従されている。一方、荷重が3kgの時は、持ち上げ始める時点で、右肩に加わる負荷を軽減するため左肩と腰が後方に振られている。

また、図9は、スクワット動作への適用例である。5kgの荷重を背中に背負った状態では、腰を持ち上げ始める時点で、背骨に加わる負荷を減少するために両腕が同時に後ろに振られ、バランスを保つために腰が前方に動かされている。

5 まとめ

本論文では、動力学計算を用いることによって、環境からの力学的な影響に応じた、リアルなキャラクターアニメーションを生成するための手法を提案した。本手法は、既存の動作データを有効に利用しつつ、動的に変化するアニメーションを生成できるため、コンピュータゲーム等への応用が期待できる。今後の課題としては、衝突を含むような動作への適用や、3.3.3節で述べたような関節角度空間での下半身の制御、より高度な筋力モデルの導入などが挙げられる。

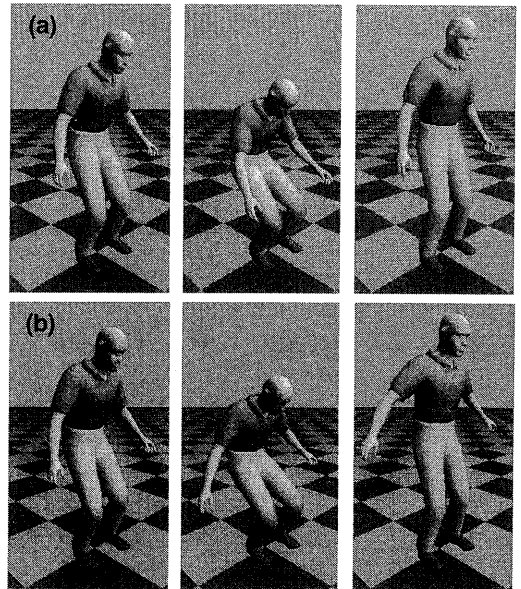


図9 スクワット動作 (a)荷重なし (b)荷重 5kg

謝辞 本研究の一部は、文部省科学研究補助金(課題番号 10308012)の援助を受けている。

参考文献

- [1] 黒瀬 能幸, “3次元図形処理工学”, 共立出版,
- [2] Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien, "Animating Human Athletes", SIGGRAPH '95 Proceedings, pp.71-78, 1995.
- [3] Jehee Lee, and Sung Yong Shin, "A Hierarchical Approach to Interactive Motion Editing for Human-like Figures", SIGGRAPH '99 Proceedings, pp.39-48, 1999.
- [4] Masaki Oshita, and Akifumi Makinouchi, "Motion Tracking with Dynamic Simulation", Computer Animation and Simulation 2000 (Proceedings of the Eurographics Workshop on Animation and Simulation 2000), pp.59-71, 2000.
- [5] Abhilash K. Pandya, James C. Maida, Ann M. Aldridge, Scott M. Hasson, and Barbara J. Woodford, "The Validation of a Human Force Model To Predict Dynamic Forces Resulting From Multi-Joint Motions", Technical Report 3206, NASA, Houston, Texas, 1992.
- [6] Seyoon Tak, Oh young Song, and Hyeong-Seok Ko, "Motion Balance Filtering", Eurographics 2000, Computer Graphics Forum, Vol.19, No.3, pp.437-446, 2000.
- [7] Michiel van de Panne, "Control for Simulated Human and Animal Motion", plenary lecture and paper, Atelier IFAC Motion Control, 1998.
- [8] M. W. Walker, and D. E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms", Journal of Dynamic Systems, Measurement, and Control, Vol.104, pp.205-211, September 1982.