

ヒューマンフィギュアの全身運動生成のための 協応構造化インタフェース

山 根 克^{*1} 中 村 仁 彦^{*2}

Pin-and-Drag Synergetic Interface for Choreography of Human Figures

Katsu Yamane^{*1} and Yoshihiko Nakamura^{*2}

This paper presents an interface for choreographing human and animal characters without reference motion. Its basic function is to enable animators to generate a natural motion clip by dragging a link to an arbitrary position with any number of links pinned in the global frame as well as other constraints such as desired joint angles and joint motion ranges. Each constraint can be switched on or off, strengthened or weakened for each joint at users' will. The interface is based on a new online inverse kinematics technique that allows more flexible attachment of pins and various types of constraints. Although the basic method can create natural motions from scratch, editing or retargeting captured motion requires only a small modification to the original method. We also demonstrate the powerfulness and usability of the proposed method by a number of example motion clips. The same methodology can be applied to flexible and intuitive manipulation of humanoid robots with large degrees of freedom in real time.

Key Words: Online inverse kinematics computation, Multiple constraints, Motion editing, Joint motion range

1. は じ め に

映画・ゲーム・インターネットなどで CG アニメーションの利用が広がりつつあるにもかかわらず、CG における人間らしい動作の生成は依然として難しく、技術のあるアニメータやモーションキャプチャに頼っているのが現状である。さらに、労力と費用をかけて一連の動作が得られたとしても、それを別の場面やキャラクタで再利用するにも高度な技術が要求される。今後、様々な分野で CG アニメーションのコンテンツが大量に必要になってくると、ヒューマンフィギュアの全身運動を手軽に制作することのできる安価で強力なツールが求められるようになると予想される。

同様の技術はヒューマノイドの動作を指示するための入力インタフェースとしても有効である。ヒューマノイドは人間と同程度の自由度を持つため、指示できる動作の柔軟性と操作の容易さを両立するのは難しい。“歩く”、“座る”など動作の種類を指示するインタフェースは操作性と直感性に優れるが、指示できる動作に柔軟性が乏しい。逆に人間の動きを計測してそのまま入力する方法は、あらゆる動作を指示できる一方、操作が難しく装置も大掛かりになるという欠点がある。

このような背景の中で、本研究ではヒューマンフィギュアに“振り付け”を行うためのインタフェースと計算アルゴリズムを開発し、CG アニメーション制作ソフトウェアの計算エンジンとして実装した。このインタフェースを以下では“ピン／ドラッグインタフェース”と呼ぶ。その基本的な機能は、Fig. 1 のように体のいくつかのリンクを空間中に固定（ピン止め）したまま別のリンクをユーザの指示（ドラッグ）どおりに動かすというものである。実際にはピン止めのほか、関節可動範囲・目標関節値の拘束が追加されているが、それ以外に人間の運動の特性を考慮したコントローラや動力学的な拘束は含まれていない。しかし、これを組み込んだ CG ソフトウェアを用いて多数

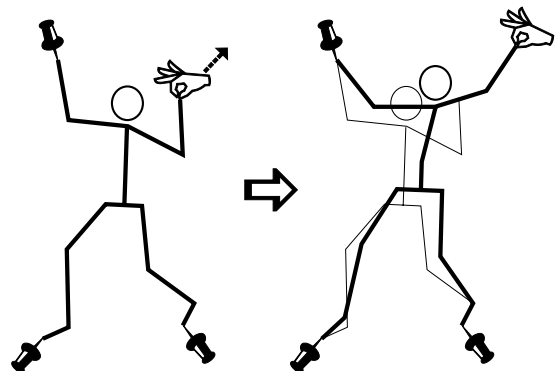


Fig. 1 The concept of pin-and-drag interface

原稿受付 2001 年 5 月 15 日

^{*1} 東京大学大学院工学系研究科機械情報工学専攻

^{*2} 東京大学大学院情報理工学系研究科知能機械情報学専攻

^{*1} Department of Mechano-Informatics, University of Tokyo

^{*2} Department of Mechano-Informatics, University of Tokyo

のアニメーションを試作した結果、CG アニメーション制作法の訓練を受けていない人でも、適切なピン止めと数回のドラッグを行うだけで人間らしい動作を生成することができることが分かった。

人間らしい動作をヒューマンフィギュアに実現する方法としてはモーションキャプチャが有力であり、モーションキャプチャデータを集めたライブラリも市販されている。またモーションキャプチャデータを運動学的・動力学的に編集して多彩な動作を生成する方法が研究されている [1]~[6]。しかし、モーションキャプチャを用いた動作生成には以下のような限界がある。まず、ライブラリに含まれていない動作が必要になるたびに新しい動作をキャプチャしなければならない。さらに、同じライブラリから作られた動作は類似したものになりがちであり、自分の好みに合わせて動作を修正するには高価なソフトと多大な労力が必要になる。

本研究で提案するインタフェースを実現するための計算法は、仮想リンクによる拘束表現 [7] と特異点低感度運動分解 (SR-inverse) [8] による特異点の影響の少ない逆運動学計算法をもとにしている。ピン止めの拘束条件を仮想リンクによって表現することで、現実のロボットにおけるエンドエフェクタとは無関係に、任意のリンクを固定リンクあるいはエンドエフェクタとして指定することができる。また複数の拘束条件の間に矛盾がある場合や機構的な特異点にある場合にも、SR-inverse を用いているため重みに応じて誤差が適当に分散され、通常の擬似逆行列を使った場合のように非現実的に大きな速度が出力されることはない。さらに、特異点の影響が少ないことを利用して、ピン止めだけでなく目標関節角や関節可動範囲など様々な拘束を統一的に取り扱い、それらの間の優先順位を自由に設定することができる。

この計算法を拡張すると、既存の運動データをもとにしてオンラインで新しい運動を生成することも可能になる。例えば、歩行中に障害物を避けるために頭を下げなければならないときに、足の運びを維持したまま頭リンクの位置を変更するような指示ができる。これを用いると、基本的な動作はライブラリから生成し、状況に応じた細かい修正は本論文で提案するインタフェースによって行う、といったアプリケーションが可能である。

2. 研究の背景

2.1 ピン/ドラッグ

ピン/ドラッグによる動作生成が目標とするのは、以下ののような動作の生成である。

- (1) ドラッグリンクが指示された軌道上を動く
- (2) ピン止めリンクが指示された位置に固定される
- (3) 各関節が可動範囲を逸脱しない
- (4) 目標値が与えられた関節の関節値が目標関節値になるべく近づく

これは多数の拘束条件が加わった逆運動学計算に当たり、解析的に解くのは明らかに非現実的である。また、各拘束条件間の矛盾のため、厳密解が存在しないことも十分考えられる。例えば、ピン止めされた位置で決まる可動範囲の外にリンクがドラッグされると、条件 (1) (2) を同時に満足する解は存在し

ない。

本研究では、まず逆運動学問題に対してはヤコビアンを用いた数値解法を適用する。さらに拘束条件間の矛盾に対応するため四つの拘束条件を 2 段階の優先順位に分け [9]、上位の優先順位を与えられる拘束条件を満足する解の零空間の中から下位の優先順位を与えられた拘束条件をできるだけ満足する解を探す。したがって、上位の拘束条件と下位の拘束条件の間に矛盾がある場合には、上位の拘束条件が優先され、ほかは無視される。上位の拘束条件は後述するヤコビアンが行フルランクである限り自由に選ぶことができる。

拘束条件の間に矛盾がある場合、すなわち上位の拘束条件を満たす解の零空間の中で最適解を探すときにランク落ちした行列が現れた場合に対処するため、特異点付近でも誤差を許すことによって極端に大きな解を出さない SR-inverse [8] を用いる。これによって、任意の数のピン・可動範囲・目標値を与えることができ、それらの間に矛盾がある場合にはあらかじめ与えられた重みに応じて誤差が配分されるため不自然な動作にはならない。また、矛盾が解消されたときには拘束条件を満足する姿勢に復帰するように、SR-inverse を用いることによって発生する誤差をフィードバックするコントローラを組み込む。

以下では、上で述べた四つの拘束条件のうち、ドラッグリンクに関する拘束条件 (1) に上位の優先順位を与え、残りの三つの拘束条件に下位の優先順位を与えるものとする。これは、人間の関節の可動範囲がロボットのように明確でないこと、厳密な目標値を持って運動しているわけではないことなどを考慮したためである。ロボットのように可動範囲の制約が厳密である場合には、(3) に上位の優先順位を与えることも考えられる。

2.2 ヤコビアンによる逆運動学計算と冗長性

ロボットの関節値に関するリンク i の位置のヤコビアンは以下のように定義される。

$$\mathbf{J}_i \triangleq \frac{\partial \mathbf{r}_i}{\partial \boldsymbol{\theta}} \quad (1)$$

ここで $\mathbf{r}_i$ はリンク i の位置、 $\boldsymbol{\theta}$ はロボットの関節値を並べたベクトル、 $\mathbf{J}_i$ が $\boldsymbol{\theta}$ に関するリンク i の位置のヤコビアンである。このヤコビアンを効率的に計算する方法については [10] などを参照されたい。ヤコビアンを用いて、リンク i の速度と関節速度の関係は以下のように表される。

$$\dot{\mathbf{r}}_i = \mathbf{J}_i \dot{\boldsymbol{\theta}} \quad (2)$$

なお、ヒューマンフィギュアのようにリンク系を表す木構造のルートが慣性座標系に固定されていないとき、その並進速度・角速度も $\dot{\boldsymbol{\theta}}$ に含まれている。 $\mathbf{J}_i$ が正方正則であれば

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_i^{-1} \dot{\mathbf{r}}_i \quad (3)$$

によってリンク i の速度から関節速度を求めることができる。

しかし、ヒューマンフィギュアは通常 30 以上の自由度を持つため、 $\mathbf{J}_i$ が正方でなく冗長性を持つ。このとき式 (2) の一般解は擬似逆行列 $\mathbf{J}_i^\#$ を用いて

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_i^\# \dot{\mathbf{r}}_i + (\mathbf{E} - \mathbf{J}_i^\# \mathbf{J}_i) \mathbf{y} \quad (4)$$

と表される．ここで E は単位行列， y は任意のベクトルである．式 (4) 右辺第 2 項が冗長性による零空間を表し，これを用いてリンク i の速度を変えずにほかの拘束条件を満足する解を探すことになる [9]．

2.3 特異点低感度運動分解

特異点低感度運動分解 [8] は damped pseudoinverse [11] と呼ばれる．以下のような線形方程式を考える．

$$Ax = b \quad (5)$$

係数行列 A が正方でないときは通常その擬似逆行列 $A^\#$ を用いて解 x を求める．しかし， A が特異点に近いときには極端に大きな解となってしまう．これは擬似逆行列が誤差 $|b - Ax|$ が最小となる解の中でノルム $|x|$ が最小になるものを探そうとするためである．これに対し，SR inverse は誤差と解を合わせたベクトルのノルムを最小にしようとするため，特異点付近において解が発散するのを防ぐことができる．また，誤差と解の重みは要素ごとに設定することができるため，誤差と解の配分を調整することも可能である．

2.4 協応構造化と動作生成

本研究で提案するインタフェースは，近年様々な分野で注目を集めている協応構造化 (synergetics) [12] の一例としても興味深い．協応構造化は幅広い分野で見られる現象で，複雑な組織が外部からの干渉を受けずに自己組織化し，秩序を持った振る舞いをするをいう．人体は多数の骨・関節・筋肉から構成されているが，同時に多数の拘束があるため，全体としての運動はそれほど多くの自由度を持たない．本研究が扱う全身運動のレベルにおいても，そのままでは同時に制御するのが難しい多くの自由度を持っているが，環境から拘束を受けることによって見かけの自由度を減らし，比較的容易に制御できるようになっている．本研究では，環境からの拘束をピン止めで模擬し，ドラッグで協応構造的な効果を引き出すことにより，単純で直感的なインタフェースによる全身運動の制御を可能にしているといえる．

2.5 関連研究

ヒューマンフィギュアの自然な動作の生成は CG 分野で盛んに研究されているが，その多くはモーションキャプチャなどで得られた動作を編集したり，ほかのキャラクタに適用したりすることを目的としており，参照運動なしに動作を生成するのは難しいとされる．

逆運動学問題に対しては，拘束を伴う動作生成に関する様々な計算方法が提案されている．多くの研究では運動全体にわたってグローバルな最適化を行うアプローチがとられている [1] [3] [13] [14] が，リアルタイムでの動作生成には向かない．一方，将来の運動データを必要としないオンラインの逆運動学計算法も提案されている [2]．この方法は本研究と同様にフィードバック制御と零空間を利用しているが，擬似逆行列を用いているためピン配置の自由度が少ない．

Badler ら [15] と Phillips ら [16] は *Jack* [17] と呼ばれるシステムにおいてピン／ドラッグと同様のインタフェースを実装している．しかし，以下のような大きな相違がある．

- Badler のシステムではドラッグするリンクが常に木構造の

ルートになるように構造を組み替えているのに対し，本研究では仮想リンクを用いた機構表現法 [7] を用いているため，木構造を組み替える必要はない．これによりドラッグリンク切り替え時のオーバーヘッドがなくなり，ユーザにとってより使いやすいインタフェースが実現できる．

- 関節可動範囲については，Badler のシステムでは回転関節のみ考え，可動範囲への射影を行っているため，可動範囲の表現が複雑になる球面関節への拡張は難しいと思われる．
- 擬似逆行列を用いているため，ピンを自由に設定することはできない．

球面関節の可動範囲の表現と実現も本研究の新しい点である．人間の体を考えると，オイラー角などの要素ごとの上限・下限を指定するだけでは不十分であることは明らかである [18]．人間の肩の構造を考えた精密なモデルも提案されている [19] が，計算時間の問題がある．本研究では可動範囲を比較的簡単に表現できる姿勢表現法を採用し，実時間で計算することを可能にした．

3. 計算法の詳細

3.1 計算法の概要

動作の生成のための計算は以下の 5 段階で行う．

- (1) ドラッグリンクを指定された位置に動かすことのできる関節速度の一般解を求める (3.2 節)
- (2) 優先度の低い拘束条件について，現在の状態と理想状態からのずれに基づいて拘束条件を表す変数の目標速度を計算する (3.5 節)
- (3) 優先度の低い拘束条件をそれぞれ表す変数の関節値に関するヤコビアンを計算する (3.4 節)
- (4) (1) で得られた一般解のうち (2) で計算された目標速度に近い速度を実現することのできる関節速度を計算する (3.3 節)
- (5) 得られた関節速度を積分して関節角データを得る

本計算法の特徴は以下のとおりである．

- シリアル機構の逆運動学計算とは異なり，1 回のドラッグで全身を動かすことができる
- どのリンクでもドラッグ可能
- ピンは好きなところに，好きな数だけ設定できる
- 拘束条件は任意に追加したり除いたりできる
- 拘束条件の相対的な重みが調整可能

3.2 ドラッグリンクの拘束

ドラッグリンクが指示された軌道の位置 r_P^{ref} ，速度 $\dot{r}_P^{ref}$ に追従するような $\dot{\theta}$ の一般解を求める． r_P をドラッグされるリンクの現在位置とすると，フィードバックを考慮した目標速度 $\dot{r}_P^d$ は次式で計算される．

$$\dot{r}_P^d = \dot{r}_P^{ref} + K_P(r_P^{ref} - r_P) \quad (6)$$

ここで K_P は正定値のゲイン行列である． $\dot{\theta}$ と $\dot{r}_P$ の関係は，関節値に関するドラッグされるリンクの位置のヤコビアン J_P を用いて

$$\dot{r}_P = J_P \dot{\theta} \quad (7)$$

と表される．ドラッグリンクが複数ある場合は，それぞれの速

度およびヤコビアンを縦に並べて $\dot{\mathbf{r}}_P, \mathbf{J}_P$ とするが, $\mathbf{J}_P$ が行フルランクである必要がある. 目標速度 $\dot{\mathbf{r}}_P^d$ を実現する関節速度 $\dot{\boldsymbol{\theta}}$ の一般解は次式で計算される.

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_P^\# \dot{\mathbf{r}}_P^d + (\mathbf{E} - \mathbf{J}_P^\# \mathbf{J}_P) \mathbf{y} \quad (8)$$

ここでフィードバック制御を行っているのは積分誤差を補償するためである. また通常の擬似逆行列の代わりに重み付き擬似逆行列 [20] を用いると, 各関節の“固さ”を調整することができる.

3.3 その他の拘束

式 (8) を

$$\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{\theta}}_0 + \mathbf{W} \mathbf{y} \quad (9)$$

と書き換える. ここで $\mathbf{W} \triangleq \mathbf{E} - \mathbf{J}_P^\# \mathbf{J}_P$, $\dot{\boldsymbol{\theta}}_0 \triangleq \mathbf{J}_P^\# \dot{\mathbf{r}}_P^d$ である.

ピン止めされたリンクが N_F 個あり, それらの位置が $\mathbf{r}_{Fi}$ ($i = 1 \dots N_F$) で表されているとする. また目標関節値が与えられた関節が N_D 個あり, それらの関節値が $\boldsymbol{\theta}_D$ で表されているとする. N_L 個の関節が可動範囲を超え, それらの関節値が $\boldsymbol{\theta}_L$ で表されているとする. N_L は動作中いつでも変化しうる. これらを用いて, ベクトル $\mathbf{p}_{aux}$ を以下のように定義する.

$$\mathbf{p}_{aux} \triangleq (\mathbf{r}_{F1}^T \quad \dots \quad \mathbf{r}_{FN_F}^T \quad \boldsymbol{\theta}_D^T \quad \boldsymbol{\theta}_L^T)^T \quad (10)$$

$\dot{\mathbf{p}}_{aux}$ と関節速度 $\dot{\boldsymbol{\theta}}$ の間には 式 (2) と同様に以下のような関係がある.

$$\dot{\mathbf{p}}_{aux} = \mathbf{J}_{aux} \dot{\boldsymbol{\theta}} \quad (11)$$

$\mathbf{J}_{aux}$ の計算は次節で扱う.

任意ベクトル $\mathbf{y}$ は以下のように計算する. まず $\mathbf{p}_{aux}$ の目標速度 $\dot{\mathbf{p}}_{aux}^d$ を 3.5 節に示す方法で計算する. 式 (9) を 式 (11) に代入すると

$$\dot{\mathbf{p}}_{aux} = \dot{\mathbf{p}}_{aux}^0 + \mathbf{J}_{aux} \mathbf{W} \mathbf{y} \quad (12)$$

が得られる. ここで $\dot{\mathbf{p}}_{aux}^0 \triangleq \mathbf{J}_{aux} \dot{\boldsymbol{\theta}}_0$ である. $\mathbf{S} \triangleq \mathbf{J}_{aux} \mathbf{W}$ と $\Delta \dot{\mathbf{p}}_{aux} \triangleq \dot{\mathbf{p}}_{aux}^d - \dot{\mathbf{p}}_{aux}^0$ を用いて以下の簡単な形の式に変形する.

$$\mathbf{S} \mathbf{y} = \Delta \dot{\mathbf{p}}_{aux} \quad (13)$$

$\mathbf{S}$ はフルランクであるとは限らないので, この方程式を解くために SR-inverse を用いる. $\mathbf{S}$ の SR-inverse を $\mathbf{S}^*$ と表すと, $\mathbf{y}$ は次式のように計算できる.

$$\mathbf{y} = \mathbf{S}^* \Delta \dot{\mathbf{p}}_{aux} \quad (14)$$

関節速度 $\dot{\boldsymbol{\theta}}$ はこれを 式 (9) に代入して得られ, さらにそれを積分すると各フレームの関節角データが得られる.

3.4 $\mathbf{J}_{aux}$ の計算

$\mathbf{r}_{Fi}$ の関節値に関するヤコビアンを $\mathbf{J}_{Fi}$ ($i = 1 \dots N_F$) で表すと, ピン止めされたすべてのリンクに対して

$$\dot{\mathbf{r}}_{Fi} = \mathbf{J}_{Fi} \dot{\boldsymbol{\theta}} \quad (15)$$

が成り立つ.

目標関節値が与えられた関節について, その速度 $\dot{\boldsymbol{\theta}}_D$ と $\dot{\boldsymbol{\theta}}$ の関係は次式のように表される.

$$\dot{\boldsymbol{\theta}}_D = \mathbf{J}_D \dot{\boldsymbol{\theta}} \quad (16)$$

ここで $\mathbf{J}_D$ は $\boldsymbol{\theta}_D$ の第 i 番目の関節が $\boldsymbol{\theta}$ の第 j 番目の関節に対応するとき第 (i, j) 要素が 1, それ以外するとき 0 になるような行列である.

同様に, $\dot{\boldsymbol{\theta}}$ と $\boldsymbol{\theta}_L$ の速度との関係は

$$\dot{\boldsymbol{\theta}}_L = \mathbf{J}_L \dot{\boldsymbol{\theta}} \quad (17)$$

と表される. ここで $\mathbf{J}_L$ は $\boldsymbol{\theta}_L$ の第 i 番目の関節が $\boldsymbol{\theta}$ の第 j 番目の関節に対応するとき第 (i, j) 要素が 1, それ以外するとき 0 になるような行列である.

以上の行列をまとめて, $\mathbf{J}_{aux}$ は次のような形になる.

$$\mathbf{J}_{aux} = \begin{pmatrix} \mathbf{J}_{F1} \\ \vdots \\ \mathbf{J}_{FN_F} \\ \mathbf{J}_D \\ \mathbf{J}_L \end{pmatrix} \quad (18)$$

$\mathbf{J}_{Fi}, \mathbf{J}_P, \mathbf{J}_L$ の球面関節に対応する部分については 3.6 節において扱う.

3.5 $\dot{\mathbf{p}}_{aux}^d$ の計算

ピン止めされたリンクの目標速度 $\dot{\mathbf{r}}_{Fi}^d$ は次式で計算される.

$$\dot{\mathbf{r}}_{Fi}^d = \mathbf{K}_{Fi} (\mathbf{r}_{Fi}^{ref} - \mathbf{r}_{Fi}) \quad (19)$$

ここで $\mathbf{r}_{Fi}^{ref}$ ピンの位置, $\mathbf{K}_{Fi}$ は正定値のゲイン行列である.

目標関節値が与えられた関節の目標速度 $\dot{\boldsymbol{\theta}}_D^d$ は

$$\dot{\boldsymbol{\theta}}_D^d = \mathbf{K}_D (\boldsymbol{\theta}_D^{ref} - \boldsymbol{\theta}_D) \quad (20)$$

と計算される. ここで $\boldsymbol{\theta}_D^{ref}$ は目標関節値を並べたベクトル, $\mathbf{K}_D$ は正定値のゲイン行列である.

可動範囲を超えた関節の目標速度は次のように計算する.

$$\dot{\theta}_{Li}^d = \begin{cases} K_{Li}(\theta_{Li}^{max} - \theta_{Li}) & \text{if } (\theta_{Li} > \theta_{Li}^{max}) \\ K_{Li}(\theta_{Li}^{min} - \theta_{Li}) & \text{if } (\theta_{Li} < \theta_{Li}^{min}) \end{cases} \quad (21)$$

ここで θ_{Li}^{max} と θ_{Li}^{min} それぞれ関節角の最大・最小値を表し, K_{Li} は正のゲインである.

式 (20), (21) は 1 自由度関節にのみ適用できる. 次節で球面関節の取り扱いについて述べる.

3.6 球面関節の取り扱い

3.6.1 目標関節値

球面関節の関節値 $\mathbf{R}_i$ と関節速度 $\boldsymbol{\omega}_i$ は, それぞれ親リンク座標系で表された 3×3 の回転行列とその角速度で定義する.

球面関節に対して目標関節値 $\mathbf{R}_{Di} \in \mathbf{R}^{3 \times 3}$ が与えられると, 目標速度は以下のようにして得られる. まず現在の関節値 $\mathbf{R}_i$ と目標の関節値 $\mathbf{R}_{Di}$ の間の誤差ベクトル $\mathbf{e}_i$ を

$$\mathbf{e}_i = \frac{1}{2} \begin{pmatrix} \Delta R_i(1, 2) - \Delta R_i(2, 3) \\ \Delta R_i(1, 3) - \Delta R_i(3, 1) \\ \Delta R_i(2, 1) - \Delta R_i(3, 2) \end{pmatrix} \quad (22)$$

$$\Delta \mathbf{R}_i \triangleq \mathbf{R}_{Di} \mathbf{R}_i^T \quad (23)$$

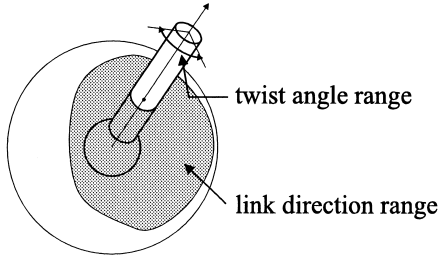
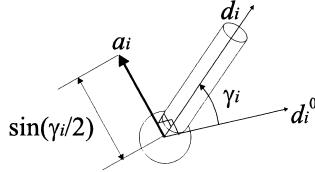


Fig. 2 Joint motion range of a spherical joint

Fig. 3 Graphical representation of $\mathbf{a}_i$ and γ_i

によって計算する。ここで $\Delta R_i(m, n)$ は $\Delta \mathbf{R}_i$ の第 (m, n) 要素を表し、 $\mathbf{e}_i$ は方向が $\mathbf{R}_i$ から $\mathbf{R}_{Di}$ への等価回転軸の方向、大きさが等価回転角を θ としたとき $\sin(\theta/2)$ となるベクトルである。次に、目標角速度 ω_{Di}^d を

$$\omega_{Di}^d = -\mathbf{K}_{Di} \mathbf{e}_i \quad (24)$$

によって計算する [21]。ここで $\mathbf{K}_{Di}$ は正定値のゲイン行列である。球面関節に対しては、式 (22)~(24) が式 (20) の代わりに用いられる。

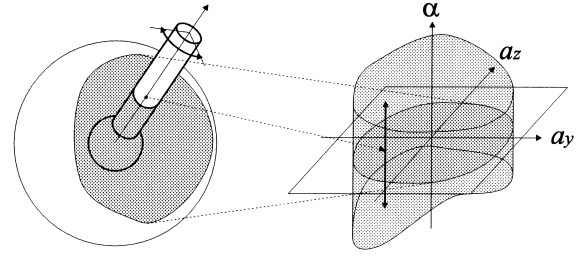
球面関節に対応して $\mathbf{J}_{Fi}, \mathbf{J}_D, \mathbf{J}_L$ の一部となるヤコビアンは 3 列からなり、それぞれの列が x, y, z の周りの回転に対応する。各列はそれぞれ対応する方向を軸とする回転関節と同様に計算できる。

3.6.2 関節可動範囲

球面関節は三つの自由度を持つので、その可動範囲は三次元のも多様体として表現される。姿勢の表現には様々な方法があり、適切な方法を選択することが計算速度の向上に重要である。例えば、よく使われるオイラー角を姿勢表現に用いると、非線形性のために可動範囲が非常に複雑な形状となり、結果的に計算速度は落ちてしまう。ここでは、球面関節の関節値を直感的に理解しやすい三つのパラメータで表現する。可動範囲がその表現の特異点を含まないと仮定すると、可動範囲は三次元空間中の領域とみなすことができる。

Fig. 2 に示すように、リンクの方向を表す二つのパラメータと、ひねり角を表す一つのパラメータを合わせた計 3 個のパラメータを用いて球面関節可動範囲を表現する。

$\mathbf{R}_i$ が単位行列のときのリンクの方向を単位ベクトル $\mathbf{d}_i^0$ で表す。実際のリンクの方向 $\mathbf{d}_i$ は Fig. 3 のように $\mathbf{d}_i^0$ を $\mathbf{a}_i$ に鉛直な平面に含まれるベクトル $\mathbf{a}_i$ の周りに回転することで得られる。 $\mathbf{a}_i$ の長さは回転角を γ_i とおいたとき $\sin(\gamma_i/2)$ とする。ひねり角 α は、 $\mathbf{R}_i$ が単位行列の状態から $\mathbf{a}_i$ の周りに回転して得られるフレームを $\mathbf{R}_i$ と一致させるために必要な回転角として定義される。以上から、球面関節が取り得るすべての

Fig. 4 Motion range of a spherical joint projected onto (a_y, a_z, α) space

姿勢は $\mathbf{d}_i^0$ を軸とする半径 1、高さ 2π の円柱内に含まれることになる。

筆者らによる実装では、すべての球面関節に対して $\mathbf{d}_i^0 = (1 \ 0 \ 0)^T$ としているため、 $\mathbf{a}_i$ は yz 面内に含まれる。すなわち $\mathbf{a}_i = (0 \ a_y \ a_z)^T$ と表される。したがって、可動範囲は Fig. 4 のように (a_y, a_z, α) 空間中の α 軸に平行な軸を持つ柱状の形になる。

a_y, a_z, α は次のように計算する。 $\mathbf{d}_i^0 = (1 \ 0 \ 0)^T$ より

$$\begin{aligned} \mathbf{d}_i &= \mathbf{R}_i \mathbf{d}_i^0 \\ &= (\mathbf{R}_i(1, 1) \ \mathbf{R}_i(2, 1) \ \mathbf{R}_i(3, 1))^T \end{aligned} \quad (25)$$

である。よって a_y, a_z は

$$a_y = -\frac{\mathbf{R}_i(3, 1)}{\sqrt{2(1 + \mathbf{R}_i(1, 1))}} \quad (26)$$

$$a_z = \frac{\mathbf{R}_i(2, 1)}{\sqrt{2(1 + \mathbf{R}_i(1, 1))}} \quad (27)$$

と求められる。 $\mathbf{a}_i$ 周りの回転を行ったあとの座標系と $\mathbf{R}_i$ の x 軸は一致しているので、ひねり角 α はそれらの y 軸と z 軸のなす角から求められる。式 (26) と式 (27) は $\gamma_i = \pm\pi$ のとき特異点となるが、通常は可動範囲の外にあるので実際は問題にならない。

ある姿勢に対応する三つのパラメータが求まったら、次にそれが可動範囲内にあるかどうかを判定する。筆者らの実装では、 a_y - a_z 平面内の可動範囲を三角形の集合で、ひねり角 α の可動範囲を三角形の各頂点における上限・下限値でそれぞれ表す。このような形状に対して、可動範囲内外の判定は以下のように行う。まず $(a_y, a_z, 0)$ が含まれる三角形を探し、見つからなければ可動範囲外である。見つかった場合は、さらに (a_y, a_z, α) が上下限内にあるかどうかを調べる。

可動範囲外であることが分かったら、それを可動範囲内に戻すための目標関節速度 $\dot{\mathbf{p}}_{aux}^d$ を計算する。可動範囲内で現在の状態に最も近い点を探し、それを目標姿勢とするのが理想であるが、可動範囲は複雑な形状であるため、最近点を求める計算量が多くなる。そこで、各球面関節に対して基準姿勢 $\mathbf{R}_{Si}$ をあらかじめ与えておき、 $\mathbf{R}_{Si}$ へ向かうための目標角速度 ω_{Li} を計算する。これは $\mathbf{R}_{Si}$ を式 (23) の $\mathbf{R}_{Di}$ へ、 ω_{Li} を式 (24) の ω_{Di} へそれぞれ代入することによって行う。

例として Fig. 5 に右肩の可動範囲を表現したものを示す。上下方向の軸がひねり角、ほかの 2 軸は腕を体の前後および左右

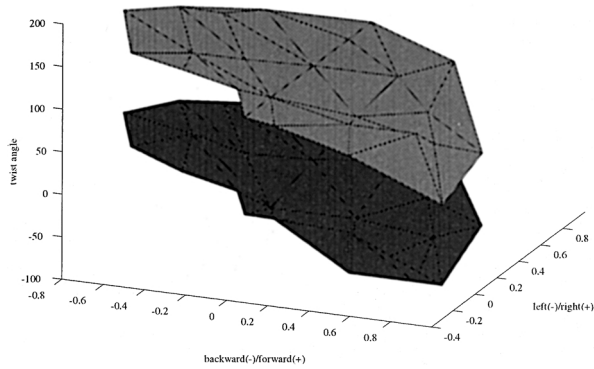


Fig. 5 An example of a motion range of a spherical joint

に振る回転に対応する。また明るい色の面がひねり角の上限，暗い色の面が下限を表している。この例では，リンク方向の可動範囲が 35 個の三角形に分割されている。筆者らのモデルでは 10 個の球面関節の可動範囲は 8 個から 35 個の三角柱で表されているが，リアルタイムの計算は十分可能である。

4. 動作のリアルタイム編集

前章までの議論では，ピンの位置や目標関節値が時間変化しないと仮定していた。しかし，それらの時間変化を含むように拡張するのは容易である。これによって，モーションキャプチャなど他の手段で得られた動作をリアルタイムで編集し，新たな動作を生成することができるようになる。

拡張に必要な作業は以下の二つである。

- 参照運動におけるピンの位置 $\mathbf{r}_{Fi}^{ref}$ ・速度 $\dot{\mathbf{r}}_{Fi}^{ref}$ を順運動学計算によって求め，式 (19) の代わりに次式を用いる。

$$\dot{\mathbf{r}}_{Fi}^d = \dot{\mathbf{r}}_{Fi}^{ref} + \mathbf{K}_{Fi}(\mathbf{r}_{Fi}^{ref} - \mathbf{r}_{Fi}) \quad (28)$$

- 目標関節値と目標関節速度 $\dot{\theta}_D^{ref}$ として参照運動における関節値・関節速度を使い，式 (20) の代わりに次式を用いる。

$$\dot{\theta}_D^d = \dot{\theta}_D^{ref} + \mathbf{K}_D(\theta_D^{ref} - \theta_D) \quad (29)$$

5. 動作生成例[†]

前章までに述べた計算法を *Animanium^{TM††}* の計算エンジンとして実装した。このソフトウェアはピン止め・ドラッグリンクの選択，重み・ゲインなどの各種パラメータの設定などのためのユーザインタフェースを備え，ユーザによるマウスの動きをドラッグリンクの三次元空間中での運動にマッピングする。PentiumIII 1 [GHz] プロセッサを持った PC 上では，45 から 54 自由度のヒューマンフィギュアに対してリアルタイムでの操作が可能である。

[†] 生成されたアニメーションの例を
<http://www.ynl.t.u-tokyo.ac.jp/~katz/videos/index.html>
 において公開している。

^{††} *AnimaniumTM*: CG アニメーション制作支援ソフトウェア ((株) セガ)

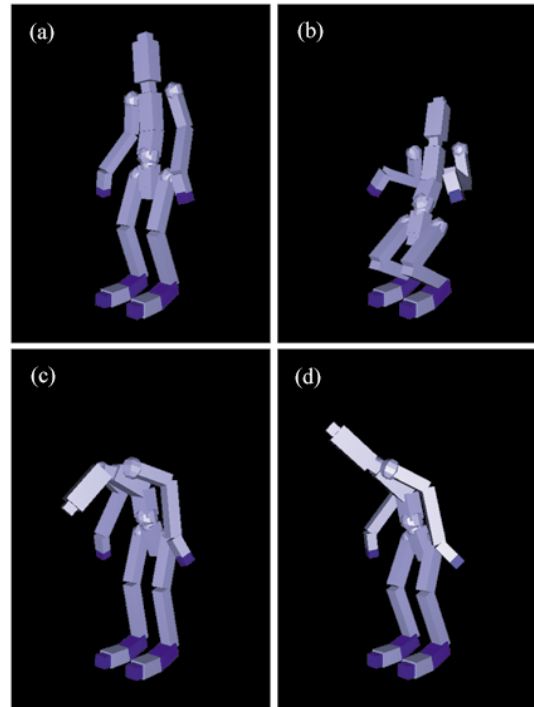


Fig. 6 Postures generated by a single pin-and-drag; (a) original posture, (b) pelvis dragged, (c) head dragged, (d) left shoulder dragged

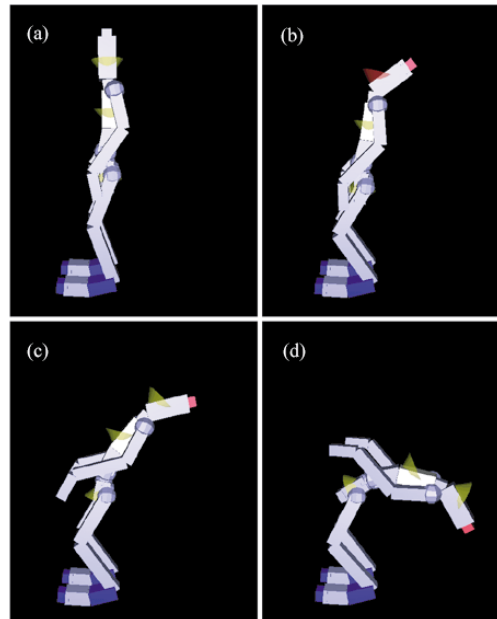


Fig. 7 Effect of applying joint motion range constraint; (a) original posture, where the motion ranges are represented by cones, (b) the neck joint exceeds its range, (c) neck joint returns back into the range, (d) the upper body is bent further without exceeding the ranges

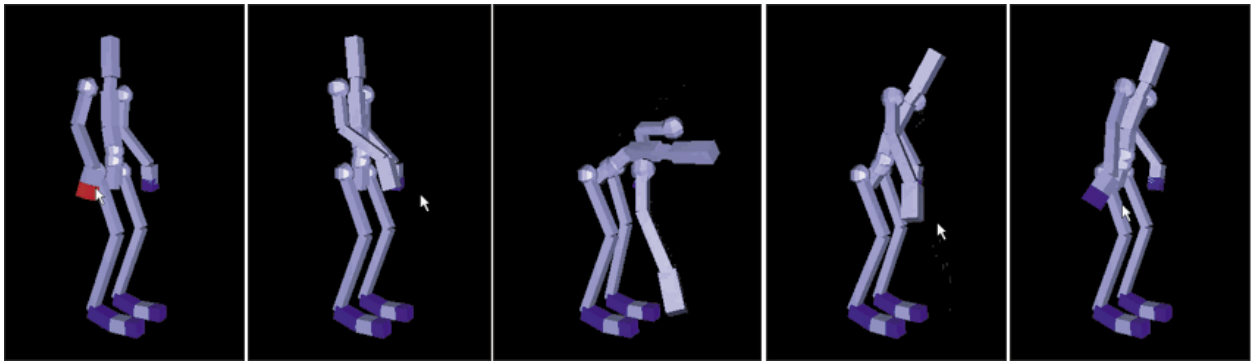


Fig. 8 An example of real-time motion generation

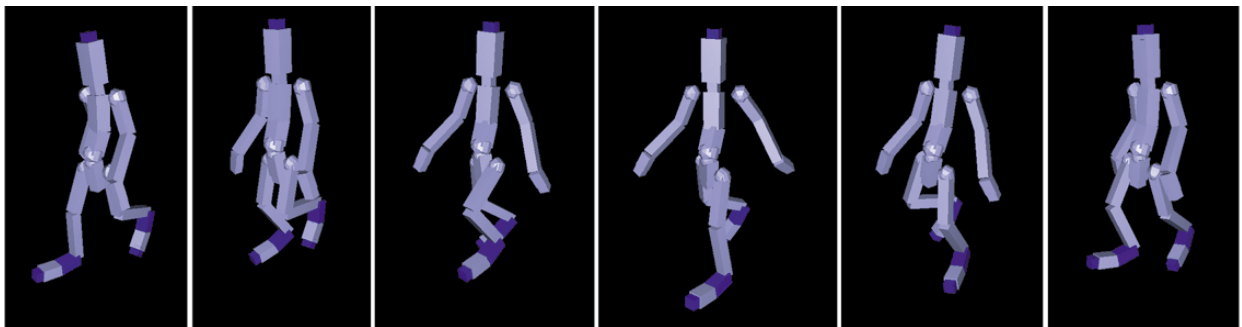


Fig. 9 Original motion used for motion editing in motion

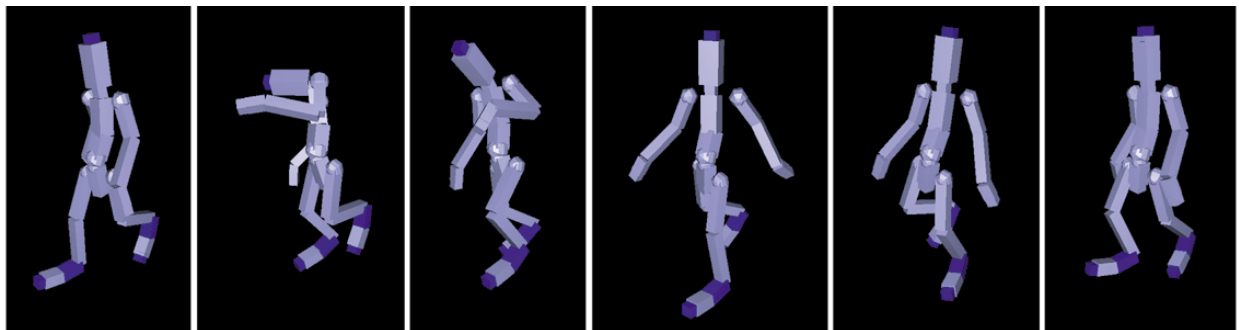


Fig. 10 Modified motion from the motion in Fig. 9

5.1 基本機能

初期姿勢 (a) において両手・両足をピン止めし、ほかの三つのリンクをそれぞれ動かしたときに生成された動作を Fig. 6 (b)～(d) に示す。

5.2 関節可動範囲の効果

Fig. 7 に関節可動範囲に関する拘束条件を与えた効果を示す。ここでは両足をピン止めた状態で、初期姿勢 (a) から最終姿勢 (d) まで頭を後ろへ動かしている。また簡単のため球面関節の可動範囲は円錐で近似して表示され、可動範囲を超えたものについては円錐を赤で表示している。(b) では首関節が可動範囲を超えているが、胸関節をさらに曲げることによって (c) では可動範囲内に戻っている。同様に最終姿勢 (d) まで各関節は可動範囲内に保たれる。

5.3 実時間動作生成

両足先、両かかと、左手をピン止めし、右手を上下に約 4 秒間にわたってドラッグすると Fig. 8 のように床の上の物を拾うような自然な動作が得られた。このようにエンドリンク以外のリンクや隣接したリンクにも自由にピンを設定することができる。

5.4 動作の実時間編集

Fig. 9 と Fig. 10 に、動作の実時間編集前後の動作を示す。現在のインタフェースでは同時に 2 個以上のリンクをドラッグすることはできないため、この例ではいったん頭の軌道を変更した後、再度再生して左手の軌道を変更するという手順で動作を編集し、歩きながら飛んでくるものを避ける動作に変更した。両足先・両かかとはピンが設定されているため、編集しても

足の運びは変化しない。

6. お わ り に

本研究の成果は以下の4点にまとめられる。

- (1) ピン/ドラッグインタフェースによりヒューマンフィギュアの自然な動作を生成するための計算法を開発した。
- (2) 開発した計算法は以下のような特徴を持つ。
 - 通常の逆運動学計算法とは異なり、特異点に関係なく任意のリンクにピンを設定することができる。
 - ピンのほか、関節可動範囲・目標関節値の拘束を追加することも可能である。
 - 既存の動作データの実時間編集へも応用することができる。
- (3) CGアニメーションソフトウェアの計算エンジンとして実装し、提案した計算法の有効性を示した。
- (4) ヒューマノイドの動作を柔軟に指示できる直感的なインタフェースとしても有用である。

課題としては動力学を考慮していないことが挙げられる。本研究はCGアニメーション制作ツールとして利用されることを想定しているため、リアルタイム性と直感性を重視して計算が高速で結果を容易に想像できる運動学計算のみを行っている。しかし、生成された運動をヒューマノイドなどに適用するためには、力学的に実現可能である必要があり、動力学の考慮が不可欠である。現状では、生成された運動にAutoBalancer [22]や力学フィルタ [6]を適用することが考えられる。

力学を用いない運動生成が自然な動作を生成できるかどうかについては、筆者たちも当初確信を持てなかった。本研究の成果である動作生成支援ソフトウェアを経験の短いCGアニメーターが使用してつくった作品を見たときには驚きであった。“人間の自然な動作”というものの本質のほとんどは、運動学的な拘束の付加・解除にある。これを適切に行えば、後は複雑な力学計算を用いなくとも大自由度系の重み付き最小二乗によって見かけの自然さを作ることができるという印象をもった。“自然さ”についての定量的な評価を定めることは本研究の目的ではないので、これについての正確な議論は今後の研究にゆだねたい。

筆者らは力学的な整合性の重要性を疑うものではない。ヒューマノイドの制御などの物理世界の問題において力学的整合性は本質的である。本論文の主張は、見かけの自然さが上述のような運動学的計算で可能であり、このようにして生成した運動パターンにさらに物理世界の条件を反映させることで、より自然で正確な運動パターンを生成できるのではないかという点にある。

謝 辞 本研究の理論面は科学技術振興事業団の戦略的基礎研究推進事業“自律行動単位の力学的結合による脳型情報処理機械の開発”（代表：中村仁彦）の支援を受けた。また実装面では情報処理振興事業団の“人間型ロボットの制御技術に基づく次世代動作生成アプリケーションの開発”（代表：今川洋尚）の支援を受けた。開発したソフトウェアは（株）セガのCGアニメーション制作支援ソフトウェア *AnimaniumTM* の計算エンジンとして使われている。第5章で示した例はこのソフトウェアを使って作られたものである。

参 考 文 献

- [1] M. Gleicher: “Retargetting Motion to New Characters,” In Proceedings of SIGGRAPH '98, pp.33–42, 1998.
- [2] K.J. Choi and H.S. Ko: “Online Motion Retargetting,” The Journal of Visualization and Computer Animation, vol.11, pp.223–235, 2000.
- [3] C. Rose, M.F. Cohen and B. Bodenheimer: “Verbs and Adverbs: Multidimensional Motion Interpolation,” IEEE Computer Graphics and Applications, vol.18, no.5, pp.32–40, 1998.
- [4] Z. Popovic: “Editing Dynamic Properties of Captured Human Motion,” In Proceedings of IEEE International Conference on Robotics and Automation, pp.670–675, San Francisco, CA, Apr. 2000.
- [5] N.S. Pollard and F. Behmaram-Mosavat: “Force-Based Motion Editing for Locomotion Tasks,” In Proceedings of IEEE International Conference on Robotics and Automation, pp.663–669, San Francisco, CA, Apr. 2000.
- [6] K. Yamane and Y. Nakamura: “Dynamics Filter — Concept and Implementation of On-Line Motion Generator for Human Figures,” In Proceedings of IEEE International Conference on Robotics and Automation, pp.688–695, San Francisco, CA, Apr. 2000.
- [7] 中村, 山根, 永嶋: “構造変化を伴うリンク系の動力学計算法とヒューマンフィギュアの運動計算”, 日本ロボット学会誌, vol.16, no.8, pp.1152–1159, 1998.
- [8] 中村, 花房: “関節形ロボットアームの特異点低感度運動分解”, 計測自動制御学会論文集, vol.20, no.5, pp.453–459, 1984.
- [9] Y. Nakamura and H. Hanafusa: “Task Priority Based Redundancy Control of Robot Manipulators,” International Journal of Robotics Research, vol.6, no.2, pp.3–15, 1987.
- [10] D.E. Orin and W.W. Schrader: “Efficient Computation of the Jacobian for Robot Manipulators,” International Journal of Robotics Research, vol.3, no.4, pp.66–75, 1984.
- [11] A.A. Maciejewski: “Dealing with the Ill-Conditioned Equations of Motion for Articulated Figures,” IEEE Computer Graphics and Applications, vol.10, no.3, pp.63–71, 1990.
- [12] Hermann Haken (斉藤他訳): シナジェティクスの基礎—不安定性の階層=システムとデバイスの自己組織化. 東海大学出版会, 1992.
- [13] C. Rose, B. Guenter, B. Bodenheimer and M.F. Cohen: “Efficient Generation of Motion Transitions using Spacetime Constraints,” In Proceedings of SIGGRAPH'96, pp.147–154, 1996.
- [14] J. Lee and S.Y. Shin: “A Hierarchical Approach to Interactive Motion Editing for Human-Like Figures,” In Proceedings of SIGGRAPH'99, pp.39–48, 1999.
- [15] N.I. Badler, K.H. Manoochchri and D. Baraff: “Multi-Dimensional Input Techniques and Articulated Figure Positioning by Multiple Constraints,” In Proceedings of the 1986 Workshop on Interactive 3D Graphics, pp.151–169, Chapel Hill, NC, Oct. 1986.
- [16] C.B. Phillips, J. Zhao and N.I. Badler: “Interactive Real-Time Articulated Figure Manipulation Using Multiple Kinematic Constraints,” In Proceedings of the 1990 Workshop on Interactive 3D Graphics, pp.245–250, March 1990.
- [17] C.B. Phillips and N.I. Badler: “Jack: A Toolkit for Manipulating Articulated Figures,” In Proceedings of ACM/SIGGRAPH Symposium on User Interface Software, 1988.
- [18] N.I. Badler, C.B. Phillips and B.L. Webber: Simulating Humans. Oxford University Press, 1993.
- [19] W. Maurel and D. Thalmann: “Human Shoulder Modeling Including Scapulo-Thoracic Constraint and Joint Sinus Cones,” Computers and Graphics, vol.24, pp.203–218, 2000.
- [20] Y. Nakamura: Advanced Robotics—Redundancy and Optimization. Addison-Wesley, Reading, MA, 1991.
- [21] J.Y.S. Luh, M.W. Walker and R.P.C. Paul: “Resolved Acceleration Control of Mechanical Manipulators,” IEEE Transactions

on Automatic Control, vol.25, no.3, pp.468–474, 1980.

- [22] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba and H. Inoue:
“AutoBalancer: An Online Dynamic Balance Compensation

Scheme for Humanoid Robots,” In Proceedings of the International Workshop on Algorithmic Foundation of Robotics, Hanover, NH, March 2000.



山根 克 (Katsu Yamane)

1974 年 8 月 1 日生. 97 年東京大学工学部機械情報工学科卒業, 99 年同大学院工学系研究科機械情報工学専攻修士課程修了, 同博士課程進学. 2000 年より 2002 年 3 月まで日本学術振興会特別研究員. マルチボディシステムの運動計算, ヒューマンフィギュアの運動計算, 人間の運動データやキープレームに基づくヒューマノイド・CG キャラクタの運動生成, 力学的整合性を考慮した多様な運動の合成などに興味を持つ. IEEE の会員.
(日本ロボット学会学生会員)



中村仁彦 (Yoshihiko Nakamura)

1954 年 9 月 22 日生. 82 年京都大学大学院工学研究科精密機械工学専攻博士課程退学. 同年より 87 年まで同大学助手. 87 年より 91 年 3 月までカリフォルニア大学サンタバーバラ校助教授, 準教授. 同年 4 月より東京大学に勤務. 現在, 同大学大学院情報理工学系研究科知能機械情報学専攻教授. 工学博士. ロボットの運動学, 動力学, 制御, および知能の問題, 特に非ホロノミックロボット, 脳型情報処理, ヒューマノイドや CG の運動・力学計算, 外科手術用ロボットなどの研究に従事. 科学技術振興事業団戦略的基礎研究推進事業「自立行動単位の力学的結合による脳型情報処理機械の開発」の研究代表者. 計測自動制御学会, システム制御情報学会, 日本機械学会, 日本コンピュータ外科学会, IEEE, ASME などの会員.
(日本ロボット学会正会員)