計測自動制御学会東北支部　第 218 回研究集会（2004.10.9）
資料番号　218－23

ゲームコントローラを用いたヒューマノイドロボットのスポーツ動作生成システムの開発

Development of a Sports Motion Generation System using a Game Controller for a Humanoid Robot

○ 安部 賢人，大沼 俊一，山野 光裕，水戸部 和久，
那須 康雄，金子 慎一郎，浅井　武，南後 淳

○Masato Abe, Syun-ichi Ohnuma, Mitsuhiro Yamano, Kazuhisa Mitobe,
Yasuo Nasu, Shin-ichiro Kaneko, Takeshi Asai, Jun Nango

山形大学
Yamagata University

キーワード：ヒューマノイドロボット(Humanoid Robot)，ゲームコントローラ(Game Controller)，動作生成(Motion Generation)
連絡先：〒992－0038　米沢市城南 4－3－16　山形大学工学部機械システム工学科
渡辺克巳研究室　安部　賢人
Tel：(0238)26－3242，Fax：(0238)26－3205，
Email：dtn94595@dipfr.dip.yz.yamagata-u.ac.jp

1．はじめに

現在，産業分野では精度を要する作業や繰り返し作業，重労働を伴う作業にはロボットが用いられ，自動化されている．しかし，今後のロボットには，より生活空間に近い場所での共存が求められている．なかでも，ヒューマノイドロボットの研究が各企業でもさかんに行われている．

ヒューマノイドロボットには人間の動作と同等で多様，かつ複雑な動作が求められる．それを実現するため，可動範囲や運動性能を考慮した動作を生成する必要がある．しかしながら，ロボットの動作生成は専門知識を持っていても困難な作業である．

そこで本研究では，専門知識を持たない人でも容易に動作生成をすることができ，かつ作成した動作に対して転倒可能性判別を行えるような 3 DCG (3 Dimensional Computer Graphics) を利用した動作

生成システムを作成した．動作生成システムには，ロボットの手，脚，腰の座標位置をゲームコントローラで教示すること，ロボットの動きを随時3DCGアニメーションに表示すること，生成した動作に対して転倒可能性判別を行う機能を持たせた．

作成した動作生成システムを用いて，いくつかのスポーツ動作を生成し，ヒューマノイドロボットに動作させて，転倒可能性判別の検討を行う．

2．動作生成システム

2.1　動作生成システムの構成

Fig.1 に動作生成システムの流れを示す．ユーザーはゲームコントローラ，マウスにより与えられた手先や足先についての教示情報を PC 側に入力する．そして PC 側はそれらの情報を元に逆運動学による角度計算, 重心計算, Zero Moment Point　(ZMP) 計算など行い，それらについてアニメーション表示を行い，PC 画面に出力する．そしてユーザーはその画面を逐次視認しながら操作することで，動作生成および転倒可能性判別を行うことができるシステムである．

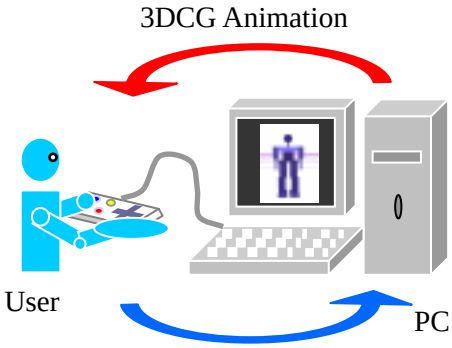

Fig.1　Motion generation system

2.2　開発環境

動作生成システムは Visual C++ (VC++) を用いて開発し, Windows 用アプリケーションソフトとして構成した．また，VC++ の機能である MFC (Microsoft Foundation Class)　を用いて作成する．MFC とは Windows 規約に従った画面制御　(ウィンドウの取得や開放)　などの定期的な GUI コーティングを行うためのライブラリである[1].

2.3　OpenGL によるモデリング

3DCG の表示には OpenGL を利用する[2]．モデルは下胴体の底を中心点とした階層構造にする．そして手先，足先座標の原点はそれぞれの肩，股関節を原点とする．

Fig.2 に階層モデルを示す[3]．上位レベルのリンクにそれぞれ優位性がある．上位レベルにあるリンクの変換操作は下位レベルに影響を与え，逆に下位レベルの変換操作は上位レベルには影響を与えることはない．アプリケーション上で体全体を移動させる場合は，脚を支持脚の状態にしその足を動作させると移動する．

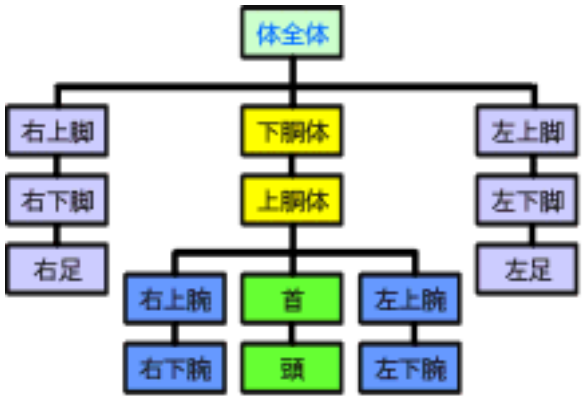

Fig.2　Hierarchical structure model

2.4　操作画面

上記のシステム，動作生成法，転倒可能性判別法をもとにアプリケーションを作成した．Fig.3 はアプリケーションの操作画面である．アニメーションモードでは動作の確認をしやすくするため，早送りや巻き

戻し，一時停止などの機能も取り付けた．

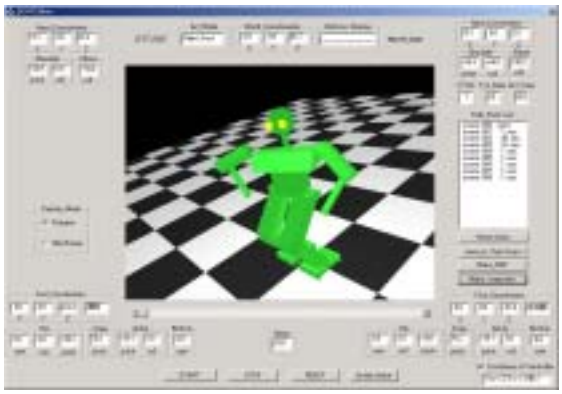

Fig.3　Operation window

2.5　ゲームコントローラの操作法

このシステムでの動作生成は主にコントローラを用いる．コントローラはロアス株式会社製の USB JOYPAD‘ JOP－U233W ’を用いた．特徴として，コントローラの十字キーは正面左側に，コントローラの正面右側と上面にそれぞれ 4 ボタン配置されている．以下にコントローラのボタンの説明, Fig.4, Fig.5 にコントローラの上面図と背面図を示す．

十字キー

縦方向：指定位置の X 方向のスクロール
(上：プラス方向，下：マイナス方向)

横方向：指定位置の Y 方向のスクロール
(右：プラス方向，左：マイナス方向)

①～⑧ボタンについて

①：支持脚・遊脚の切り替え

②：足を動かすとき，⑤か⑥ボタン同時押しで足底の Yaw 方向を指示

③：動作生成モードとアニメーションモードの切り替え

④：X・Y・Z 方向のスクロール時に同時押しで倍速

⑤：指定位置の Z 方向のマイナス方向へのスクロール

⑥：指定位置の Z 方向のプラス方向へのスクロール

⑦，⑧：動作生成モード時に手先・足先・腰の選択

Fig.4　USB JOYPAD (front view)

Fig.5　USB JOYPAD (top surface view)

2.6　マウスの操作法

マウスは動作生成したものについて動作データを作成したり，アニメーションをスタートするなど主にアプリケーション操作画面上での操作に用いる．また，アニメーション画面について視点の変更したい場合にも用いる．左クリックしながらドラックすると視点角度, Shift キーと左クリックしながらドラックすると視点の平行移動, Ctrl キーと左クリックしながらドラックすると視点の縮小拡大が行える．また視点はアプリケーション起動中ならいつでも変更できる．

3. 動作生成までの流れ

動作生成には各関節角度を求めることが必須条件である．本研究では短時間で容易に動作を生成できる動作生成法を考案した．

3.1　通過点の教示

動作生成するためには左右の腕, 足, 腰, 首についての各関節角度を求める必要がある. 本研究では直接移動に関係する腕・足・腰についてそれらの関節角度を求める．

動作はおもにゲームコントローラで手先・足先について３次元座標を教示することにより生成していく．手先・足先をコントローラで動かしながら，ユーザーは動作の通過点を設けていく．ゲームコントローラでは手先・足先は同時には動かせないが，手，足を同時に動作させる動作を生成する際には，順不同でよいので手先，足先をそれぞれ通過させたい場所へ移動してから，通過点として指定すればよい．また，腰に関しては回転軸が一つだけなので，コントローラにより，腰の角度を直接増減させて動かすようにする. このような動作を行い，通過点を複数，プログラムに記憶させる．

通過点を指定する際，静的動作での転倒可能性判別を行う．転倒すると判別できるなら通過点の指定からやり直す必要がある．

3.2　関節角度データの生成

動作の通過点を指定する前に，補間時間を与える．関節角度データの生成の特徴は，動作の通過点を複数指定し，それらの間について通過点のデータを 3 次補間して関節角度データを生成することである．

補間時間を大きくすると，その通過点間のかかる時間が大きくなる．これにより，生成される角度データが多くなり，補間時間を小さくすると生成される角度データは少なくなる．

作成されたデータをアニメーションに表示する．このとき，動的動作での転倒判別を行う．転倒すると判別できるなら，通過点の指定からやり直すか補間時間を修正する必要がある．

3.3　転倒可能性判別法

転倒とは床面と足底との密着が無くなったときと定義する．Bonten-maruⅡにおける動作での静的動作と動的動作に分けて判別する．

3.3.1　静的動作での転倒可能性判別

静的動作による転倒可能性判別は通過点を指定するときに行う．静的動作とは，動作中に各リンクに生じる加速度がゼロ，もしくは限りなくゼロに近い動作のことを示す．この場合，重心により判別を行う．単脚支持期は重心が支持脚の足底に，両脚支持期は支持脚によってできる支持多角形の鉛直上方に存在すれば安定であると判別できる．

重心は，各リンクにおける重心の位置と重量，全体の重量より求める．各リンクにおける重心の位置は順運動学より求める．

3.3.2　動的動作での転倒可能性判別

動的動作による転倒可能性判別は動作が完成し，アニメーションに表示するときに行う．動的動作とは，動作中に各リンクに生じる加速度が大きい状態を示す．この場合，ZMP により判別を行う．ZMP とは床反力モーメントがゼロとなる点のことである. ZMP が支持脚によってできる支持多角形内に存在すれば，モーメントのつりあいにより床面とロボットの足底との密着が保

たれる．転倒しないための ZMP の存在可能領域は重心と同じように判別できる．

ロボットの運動状態から ZMP の位置を求める式は床面が x－y 平面となる座標系で ZMP の位置を $P=(x_P,y_P,0)$ とすれば，ロボットのリンク i 番目の位置，質量，慣性テンソル，角速度ベクトルをそれぞれ $r_i=(x_i,y_i,z_i)^T$，m_i，I_i，ω_i，重力加速度を g として，次のように表せる．

$$x_P=\frac{\sum m_i z_i \ddot{x}_i-\sum\{m_i(\ddot{z}_i+g)x_i+[0,1,0]^T I_i\omega_i\}}{-\sum m_i(\ddot{z}_i+g)} \quad (1)$$

$$y_P=\frac{\sum m_i(\ddot{z}_i+g)y_i-\sum\{m_i z_i \ddot{y}_i+[1,0,0]^T I_i\omega_i\}}{\sum m_i(\ddot{z}_i+g)} \quad (2)$$

ただし，慣性テンソルに関してはここでは無視している．

前節で述べた重心と ZMP はアニメーションによって表示される．そこで判別しやすくするため，重心と ZMP の Z 軸を床面に固定することで支持多角形内に存在するかを判断しやすくした．転倒可能性判別をしやすくするため，アニメーションの表示を通常のポリゴンモードとワイヤーフレームによる表示を行うワイヤーフレームモードを作成した．転倒可能性判別を行うときは Fig.6 のように視点を変更し，ワイヤーフレームモードにすることで重心と ZMP の動きが見やすくなる．

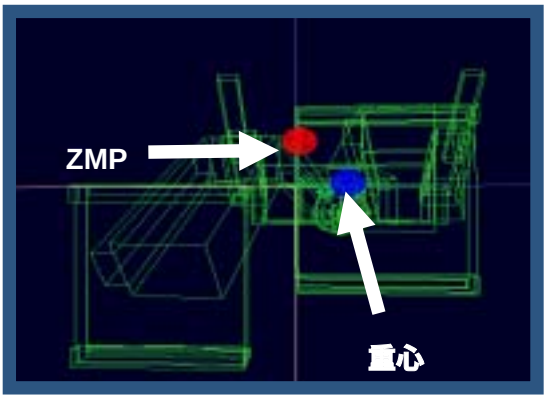

Fig.6　Bonten-Maru II　(under view)

4.　動作生成例

作成した動作生成システムを使って実際にスポーツ動作を生成した例を示し，ロボットに動作させる．アニメーションと同様の動作が生成できているか，転倒可能性判別は正しいのかを検討する．

実験で用いるヒューマノイドロボット Bonten-Maru II の概観図と自由度配置図を Fig.7 に示す. 全長 1250 [mm], 全幅 540 [mm]，全重量 31.5 [kg]となっている．構造部はアルミニウム合金を使用している．

作成する動作はボールを蹴る動作とする．Fig.8 に作成した動作をアニメーションにして，それを 5 コマに区切り，図にして示す．

Fig.9 には動作生成システムで生成した動作させ，それを 5 コマに区切り，図にして示す．

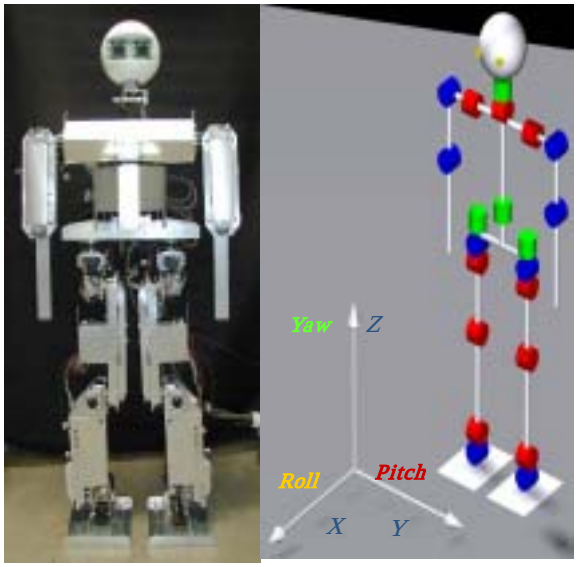

Fig.7　Bonten-Maru II and its joint location

Fig.9 より，Fig.8 のアニメーションと同様の重心移動，片足立ち，脚を引く，ボールを蹴る一連の動作を転倒することなく動作を実行することができた．とくに片足立ちの動作で転倒が心配されたが，転倒可能性判別どおりだった．

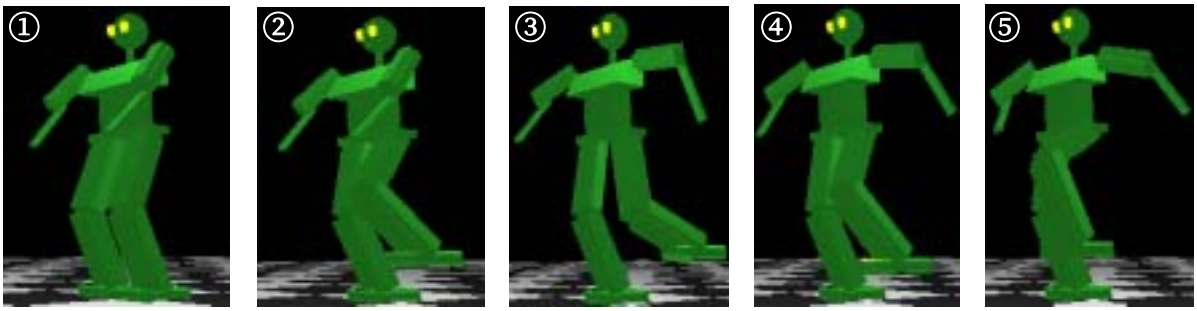

Fig.8　Animation

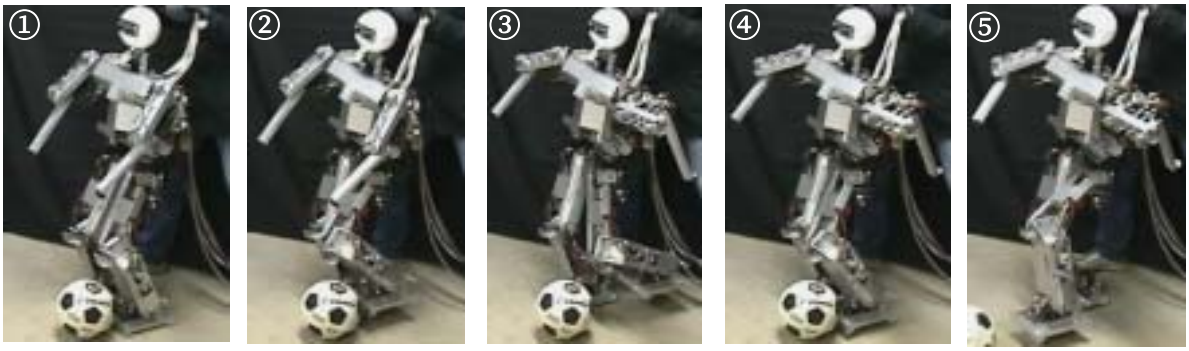

Fig.9　Experiment

5. おわりに

本研究では，ゲームコントローラを用いたヒューマノイドロボットの動作生成システムの開発をした．これにより，ユーザーの専門知識が無くても視覚的に転倒可能性判別を行いながら動作を生成できるようになった．また手足先座標入力をコントローラで行うことによってアニメーション操作が容易になった上，角度データは自動で計算をするため，生成にかかる時間は非常に短縮された．

今後，これを使ってより複雑なスポーツ動作をつくる予定である．スポーツ動作には速い動作が要求されるので，慣性テンソルは無視できないものとなる．ZMP の計算に慣性テンソルの項も含める予定である．

参考文献

[1]　山地秀美：はじめての Visual C++, 技術評論社

[2]　Crayton・Walnum：Win32 OpenGL プログラミング，株式会社ピアソンエデュケーション

[3]　酒井幸市：OpenGL 3D プログラミング，CQ 出版社