

複数のシミュレータを用いた2足歩行ロボットの歩行動作の検証

AWA, Yuhi / 粟, 雄飛

(発行年 / Year)

2009-03-24

(学位授与年月日 / Date of Granted)

2009-03-24

(学位名 / Degree Name)

修士(工学)

(学位授与機関 / Degree Grantor)

法政大学 (Hosei University)

2008 年度 修士論文

複数シミュレータを用いた

2足歩行ロボットの歩行動作の検証

指導教授 小林 一行 准教授

大学院工学研究科

システム工学専攻修士課程

学生証番号 07R6104

氏名 あわ ゆうひ
粟 雄飛

VERIFICATION OF WALKING MOTION ON BIPED ROBOT BY USING MULTIPLE SIMULATORS

Yuhi Awa

Abstract

This paper describes the physical model simulation by using the Open Dynamics Engine (ODE) and matrix based simulation by using MATLAB. In order to confirm the simulation results of the physical model, we derive a mathematical model and perform simulation by using both ODE and MATLAB. Independent simulation model-building and simulation by using both ODE and MATLAB can avoid careless errors. We used this approach to build the KHR-1, and confirmed the validity of the physical simulations by both ODE and MATLAB simulations.

Key Words: Open Dynamics Engine (ODE), MATLAB, KHR-1

目次

Abstract.....	i
目次	ii
List of Figure	iii
List of Photo.....	iii
List of Table.....	iii
1. はじめに	1
1.1. KHR-1.....	2
2. Open Dynamics Engine (ODE).....	4
2.1. ODE の特徴	5
2.2. ODE の単位系	5
2.3. ODE の座標系	5
2.4. ロボットのモーター制御.....	6
3. MATLAB.....	7
3.1. MATLAB における計算エンジン	7
3.1.1. 行列指数関数による回転行列の定義.....	7
3.1.2. モーターの位置および姿勢の算出.....	8
3.1.3. モーターの速度および角速度の算出.....	10
3.1.4. モーターの加速度および角加速度の算出.....	12
3.1.5. モーターの力およびモーメントの算出.....	16
3.2. ハードウェアでの接続環境.....	18
4. 2足歩行ロボットの歩行動作の検証.....	20
4.1. シミュレーションモデルの構築.....	20
4.2. 2足歩行ロボットの歩行シミュレーションおよび歩行実験.....	21
4.2.1. ODE でのシミュレーション	21
4.2.2. MATLAB でのシミュレーション	22
4.2.3. KHR-1 での検証.....	26
4.3. 結果および考察.....	26
5. むすび	27
謝辞	28
付録	29
研究業績	32
参考文献	33

List of Figure

Fig. 1 Joint configuration of the KHR-1	3
Fig. 2 Demonstration program using ODE	4
Fig. 3 System of ODE coordinates.....	6
Fig. 4 Axis vector of the KHR-1	8
Fig. 5 Kinematics of KHR-1 on three-dimensional space	15
Fig. 6 Force and moment that acts on the arm j	17
Fig. 7 Hardware association engine	19
Fig. 8 The simple models of the KHR-1	21
Fig. 9 Simulation screen of walking control by using ODE	22
Fig. 10 The flow chart in MATLAB simulation	22
Fig. 11 Walking motion of KHR-1 by using MATLAB.....	24
Fig. 12 Kinematics parameters of simulation results by using MATLAB (end-point of the right leg)	25
Fig. 13 Dynamics parameters of simulation results by using MATLAB (end-point of the right leg)	25
Fig. 14 Composition of folder and outline of program	29

List of Photo

Photo 1 KHR-1	1
---------------------	---

List of Table

Table 1 Specifications of the KHR-1	2
Table 2 Specifications of the KRS786ICS.....	2
Table 3 Homogeneous transformation matrix of the KHR-1	10
Table 4 Specifications of the KHR-1 model on ODE.....	20
Table 5 Home position of the KHR-1	20
Table 6 Parameters in MATLAB simulation	29
Table 7 Software version	31

1. はじめに

現在, 3次元モデリングとシミュレーションがロボットのラピッドプロトタイプ開発においてコスト面の観点からも重要視されている. しかし, 複数のソフトウェアによるシミュレーションでは, 座標系の取り方の違いなどにより, 大幅な修正が必要となってしまうことがある. また, 実機での検証では, シミュレーションと同様の内容を実行したとしても, シミュレーション環境と実環境の違いから, 思い通りの結果が得られない可能性が出てしまう. そのため, シミュレーション結果が本当に信頼性の高いものであるのか確認する必要がある.

本研究では, 異なったタイプの 2 種類のシミュレータでシミュレーションを行うことでこの問題の解決を図る. 一方は, オープンソースで C 言語ベースの物理計算エンジンである Open Dynamics Engine (ODE)で, もう一方は, プロプライエタリで行列ベースの数値解析ソフトウェアである MATLAB によるロボットの運動学シミュレータ環境である. 実機には近藤科学株式会社の KHR-1 (Photo 1) を使用し, それぞれのソフトウェア上で KHR-1 に準じたモデルを作製する. シミュレーションの内容は歩行動作を行い, ロボットのモデルの整合性を確認する. また, MATLAB 上で KHR-1 を動作させることができるようにすることで, ODE のシミュレーション結果を確認できるようにする.

最終的には, それぞれ独立しているロボットモデルの構築とシミュレーションにおいて, ODE と MATLAB の両方を使用することにより, 誤差を減らすことを目指す. また, 実機とシミュレーション間の整合性のチェックを行う.

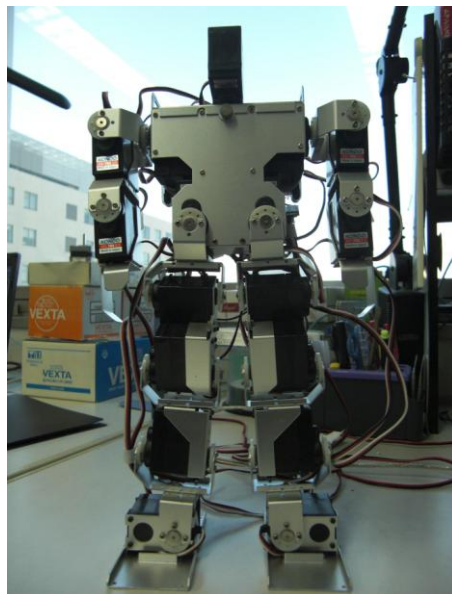


Photo 1 KHR-1

1.1. KHR-1

各ソフトウェアについて説明する前に、KHR-1 について説明を行う。

KHR-1 の主なスペック^[1]および KHR-1 に使用するモーターKRS786ICS のスペック^[2]をそれぞれ Table 1, Table 2 に、KHR-1 の関節構成図を Fig. 1 に示す。ここで、Fig. 1 のカッコ内の数字はモーターの ID 番号を表している。これは、コントロールボードである RCB-1 に接続しているチャンネルを基にしている。

Table 1 Specifications of the KHR-1

Height	340×180[mm]
Weight	1.2[kg]
Degree of freedom	17 (Head : 1 Arm 3×2 Leg : 5×2)
Connection form to Personal Computer	RS232C
Servo & Controller	RCB-1×2
Motor	KRS786ICS

Table 2 Specifications of the KRS786ICS

Size	41×35×21[mm]
Weight	0.045[kg]
Maximum working degree	180[deg]
Maximum torque	8.7[kg/cm]
Maximum speed	0.17[sec/60deg]
Normal-rated voltage	6.0[V]

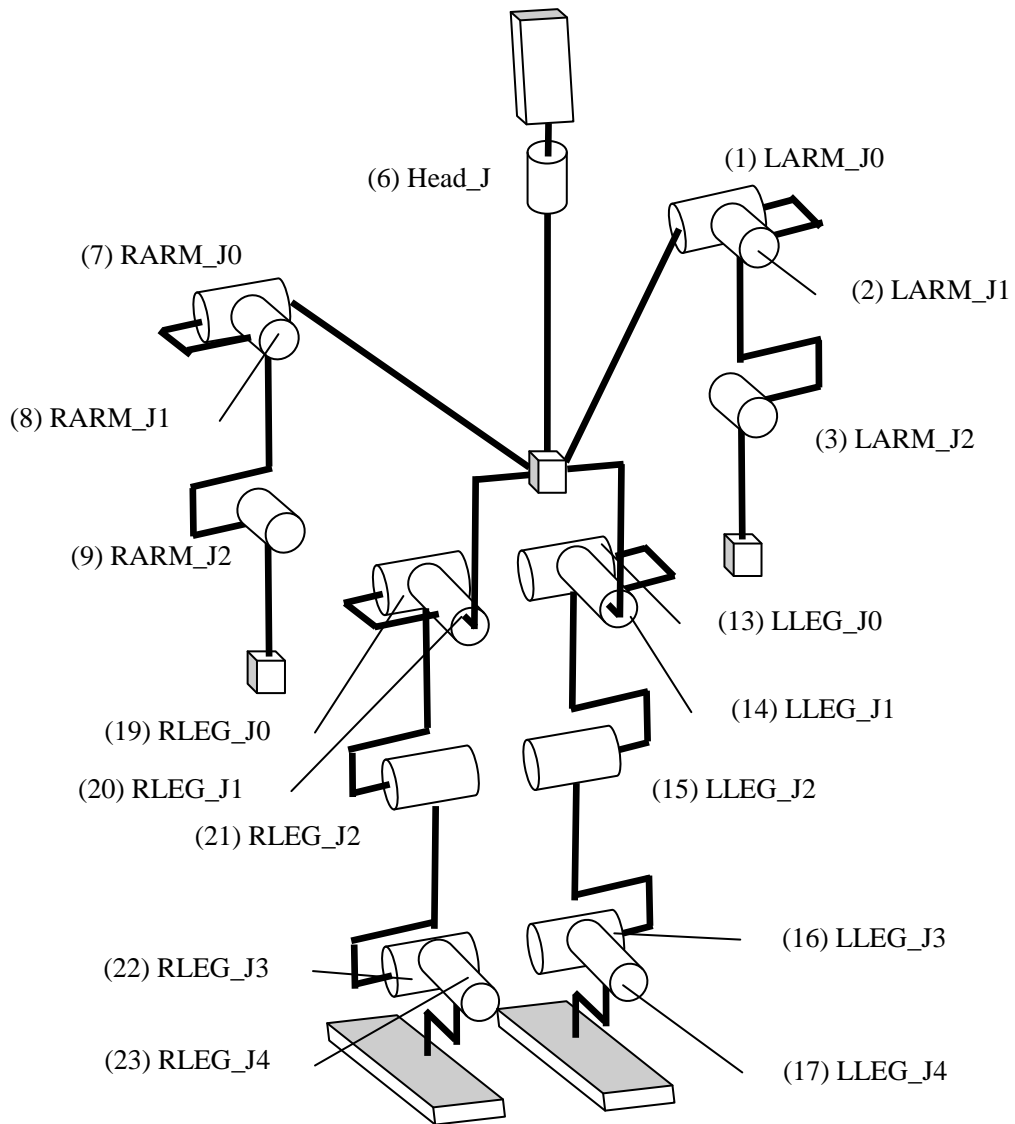


Fig. 1 Joint configuration of the KHR-1

2. Open Dynamics Engine (ODE)

ODE は、オープンソースかつ無料でありながら商用並みの高いクオリティを実現し、商用ゲームや研究シミュレータなどに採用されている物理計算エンジンである。

ODE は、制御の原因でシミュレーションエラーが生じないように、高度で安定した積分器(integrator)が使用されている。この物理的の意味はシミュレートシステムが理由もなく爆発したかのように高速分解することがないようにするためである。ODE は物理的な精度よりもスピードと安定性が重視されている^[3]。

運動方程式は, Trinkle/Stewart と Anitescu/Potra のラグランジュ乗数速度ベースのモデルから導出され^[4] ^[5], Time-stepping 法という一定の時間間隔(ステップ)でシミュレーションを更新する方法が採用されている。この時間間隔の間に、微分方程式の数値積分と衝突検出計算が行われている^[6]。つまり、ステップサイズの大小により、数値積分の精度と速度がトレードオフの関係になる。また、衝突検出計算に関しては、ステップサイズを大きくすると、剛体が相互に侵入する物理的に存在し得ない現象が生じてしまう。

Fig. 2 に、ODE のデモンストレーションプログラムのシミュレーション画面を示す。

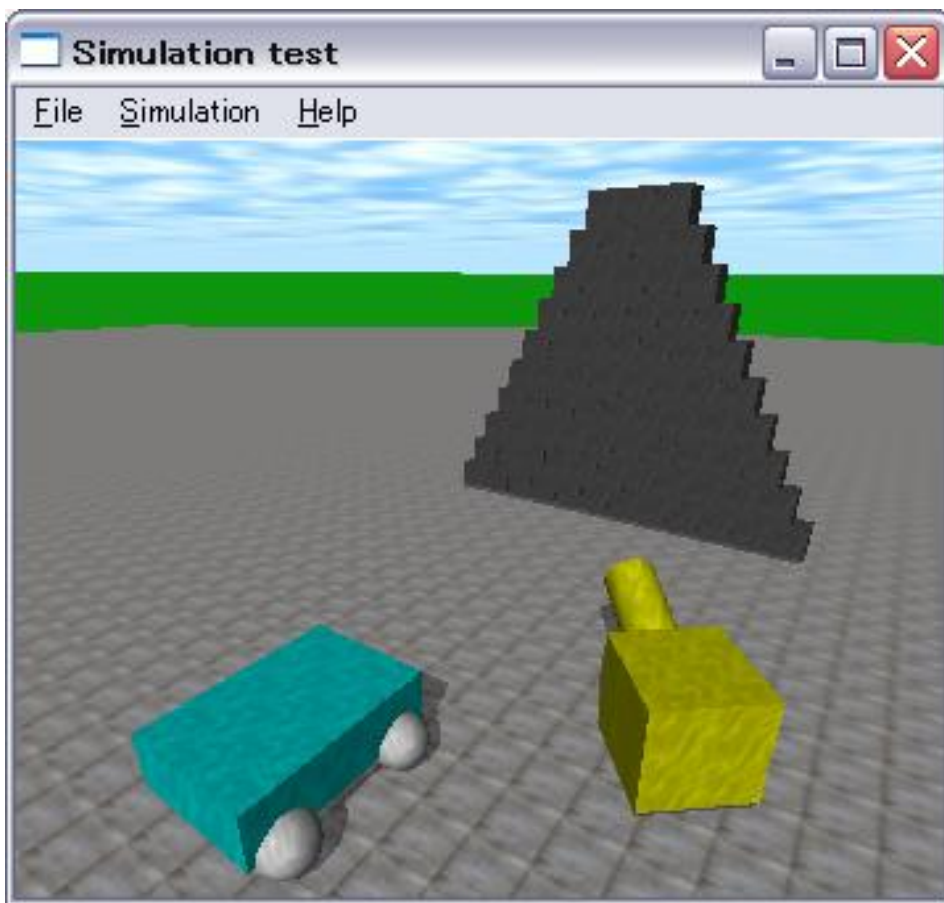


Fig. 2 Demonstration program using ODE

2.1. ODE の特徴

ODE の特徴は、一言で言えばフリーで使いやすい物理計算エンジン(剛体動力学計算)ということである。以下に特徴の詳細について紹介する。

まずは、フリーということである。オープンソースでライセンスは GNU の LGPL か BSD となる。商用でもフリーで利用することができる。

3次元グラフィクス(Open GL)と衝突検出機構が組み込まれているため、すぐ使うことができ、商用のゲームエンジンなどにも使われるくらいなので、スピードは高速である。遺伝的アルゴリズム、強化学習などの研究用に適している。ただし、計算精度が高くないので精度が求められる用途には適さない。

ODE は、プラットフォームとして Windows, Linux, Mac で使用可能であり、プログラミング言語は C か C++ が使用できる。

2.2. ODE の単位系

ODE の単位系は、基本的に決まっていない。しかし、角度だけは[radian]となっている。そのため、本論文では世界標準である SI 単位系を用いることとする。つまり、長さは[m], 質量は[kg], 時間は[s]として定義する。

2.3. ODE の座標系

ODE の座標系は、Fig. 3 に示すように物理や数学で一般的に使われている右手系の直交座標である。ODE のシミュレーション画面には必ず 9 個の小さなピラミッドが表示され、原点は 9 個ある小さいピラミッドの中心に相当する。中心から赤いピラミッド方向が x 軸、青いピラミッド方向が y 軸、上空方向が z 軸となっている。なお、本論文では SI 単位系を採用しているため、各ピラミッドは 1 [m] の間隔で並んでいることとなる。

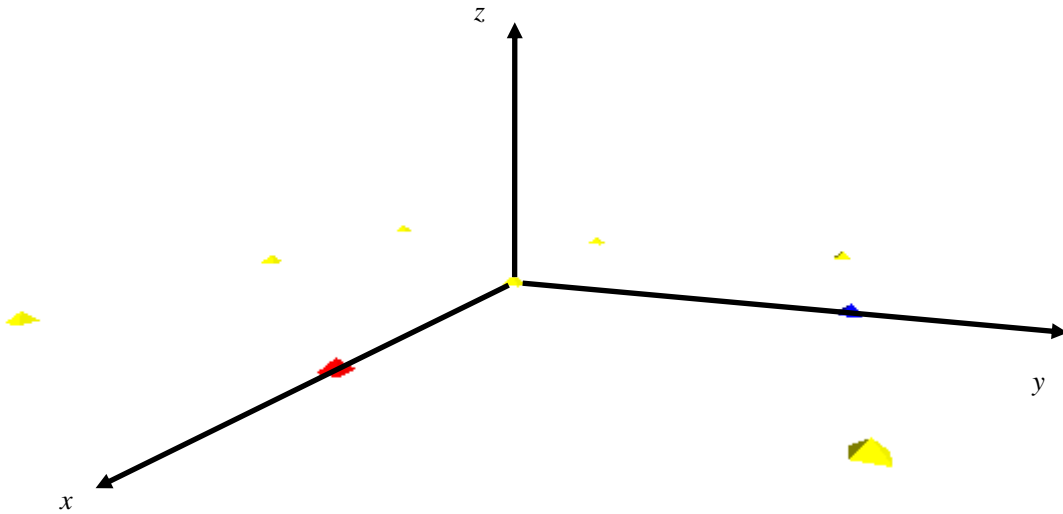


Fig. 3 System of ODE coordinates

2.4. ロボットのモーター制御

ODE では、各関節に目標の角度を直接指定しても、それに基づいて角度が自動で調節されない。このため、KHR-1 の歩行制御を行う際、各関節の角度を目標の角度にもっていくため、PID 制御を用いる。

ある時刻 t において、目標の角度 θ_d と現在の角度 $\theta(t)$ の偏差を $e(t)$ としたとき、

$$u(t) = K_p \left\{ e(t) + \frac{1}{T_i} \int_{\tau=0}^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right\} \quad (1)$$

$$e(t) = \theta_d - \theta(t) \quad (2)$$

なお、 K_p を比例ゲイン、 T_i を積分時間、 T_d を微分時間と呼ばれている。

モーターの制御方法は、 $\theta(t)$ が θ_d に近づくにつれて目標の角速度を 0 に近づけ、それを実現するための最大トルクを設定する。これを繰り返すことで $e(t)$ を小さくするよう制御を行う。 $\theta(t) = \theta_d$ になったときは角速度が 0 [rad/s] となり、必然的にトルクが 0 [N·m] となるため、モーターの動作が停止することとなる^[6]。

3. MATLAB

ODE と同様のシミュレーションを行うことができるようにするため、MATLAB では、第 3.1 節で述べる計算エンジンと第 3.2 節で述べるハードウェアでの接続環境を提供する。

3.1. MATLAB における計算エンジン

3.1.1. 行列指数関数による回転行列の定義

KHR-1 のモーターなどは、あらかじめ特定の軸が設定され、その軸を中心に回転して動作している。この計算を MATLAB 上で行う場合、 x 軸、 y 軸、 z 軸周りそれぞれの回転行列を組み合わせて任意の回転姿勢を得るのは計算が面倒である。そこで、原点を中心とする単位ベクトル $\mathbf{a} = [a_x \ a_y \ a_z]^T$ を軸にとり、その周りの回転角度 θ を回転する回転行列 \mathbf{R} を以下のように定義する:

$$\mathbf{R} = \mathbf{E}_3 + \mathbf{a} \times \sin\theta + (\mathbf{a} \times)^2 (1 - \cos\theta) \quad (3)$$

ちなみに、 \mathbf{E}_3 は 3×3 の単位行列であり、 \times は外積を表す。これを、行列指数関数を使って表現すると

$$\mathbf{R} = e^{\mathbf{a} \times \theta} \quad (4)$$

となり、MATLAB 上では `expm` 関数を用いることで計算が容易にできる。

また、外積 $\mathbf{a} \times$ を計算することは、

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_y z - a_z y \\ a_z x - a_x z \\ a_x y - a_y x \end{bmatrix} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5)$$

のように歪対称行列を左から乗算することと等価である。今後の計算の表記上、これを明確にするため $[\mathbf{a} \times]$ と表す。すなわち、

$$[\mathbf{a} \times] = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (6)$$

これをもとに、関節軸ベクトル \mathbf{a} を KHR-1 に適用したものを Fig. 4 に示す。なお、 \mathbf{a} の

添え字は、ID 番号を表している。

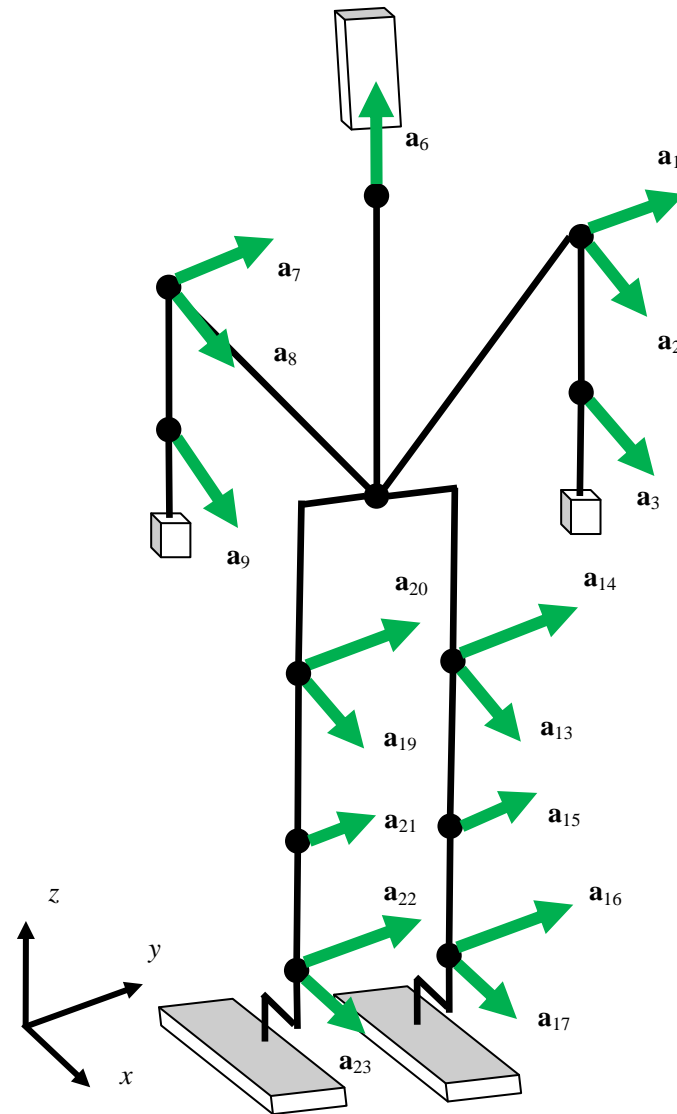


Fig. 4 Axis vector of the KHR-1

3.1.2. モーターの位置および姿勢の算出

モーターの先端位置 $\mathbf{p} = [x \ y \ z]^T$ を求めるために、前節の行列指数関数 \mathbf{R} を用い、 4×4 の同次変換行列 \mathbf{T} を次式のように定義する:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

この表記により、1つの同次変換行列がそのままモーターの位置および姿勢を表すことがで

きるため、同次変換行列を乗算するだけで位置座標に姿勢を含めて拡張することができる。

これより、第 n 番目のモーターの先端の絶対位置 ${}^0\mathbf{p}_n$ を求めていく。話を単純明快にするため、Fig. 5 に、1 から n までモーターがつながったときの運動学の計算方法について示す。この場合、0 フレームが基準座標とし、胴体側からモーターを順番に番号を振っていく、このとき、隣り合うモーター、つまり $n-1$ 番目と n 番目のモーターの間の同次変換行列を、

$${}^{n-1}\mathbf{T}_n = \begin{bmatrix} {}^{n-1}\mathbf{R}_n & {}^{n-1}\mathbf{p}_n \\ \mathbf{0} & 1 \end{bmatrix} \quad (8)$$

と表すことにする。ここで、 ${}^{n-1}\mathbf{p}_n$ および ${}^{n-1}\mathbf{R}_n$ は $n-1$ 番目のモーターから見たときの n 番目のモーターの先端の相対位置および相対姿勢である。これより、第 n 番目のモーターの先端の絶対位置 ${}^0\mathbf{p}_n$ を求める。これは、ロボットの同次変換行列を胴体側から手先・足先側に向かって右から乗算することで、姿勢 ${}^0\mathbf{R}_n$ を含めて拡張して求めることができる。

$${}^0\mathbf{T}_n = \begin{bmatrix} {}^0\mathbf{R}_n & {}^0\mathbf{p}_n \\ \mathbf{0} & 1 \end{bmatrix} = {}^0\mathbf{T}_1\mathbf{T}_2\cdots{}^{n-1}\mathbf{T}_n \quad (9)$$

これより、それぞれの腕や脚先の同次変換行列は Table 3 のように表現できる。

同様の考えでモーターの重心位置 $\mathbf{p}_g = [x_g \ y_g \ z_g]^T$ を求めると、同次変換行列 \mathbf{T}_g を使って次式のように定義することで計算できる。

$$\mathbf{T}_g = \begin{bmatrix} \mathbf{E}_3 & \mathbf{p}_g \\ 0 & 1 \end{bmatrix} \quad (10)$$

これより、第 n 番目のモーターの重心の絶対位置 ${}^0\mathbf{p}_{g_n}$ を求めると、隣り合うモーター間の重心の同次変換行列を、

$${}^{n-1}\mathbf{T}_{g_n} = \begin{bmatrix} \mathbf{E}_3 & {}^{n-1}\mathbf{p}_{g_n} \\ \mathbf{0} & 1 \end{bmatrix} \quad (11)$$

と定義したとき、

$${}^0\mathbf{T}_{g_n} = \begin{bmatrix} \mathbf{E}_3 & {}^0\mathbf{p}_{g_n} \\ \mathbf{0} & 1 \end{bmatrix} = {}^0\mathbf{T}_1\mathbf{T}_2\cdots{}^{n-1}\mathbf{T}_{g_n} \quad (12)$$

と求めることができる。なお、 ${}^{n-1}\mathbf{p}_{g_n}$ は $n-1$ 番目のモーターから見たときの n 番目のモーターの重心の相対位置である。

Table 3 Homogeneous transformation matrix of the KHR-1

Parts	Homogeneous transformation matrix	ID
Left arm	${}^0\mathbf{T}_3 = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3$	1→2→3
Face	${}^0\mathbf{T}_6$	6
Right arm	${}^0\mathbf{T}_9 = {}^0\mathbf{T}_7 {}^7\mathbf{T}_8 {}^8\mathbf{T}_9$	7→8→9
Left leg	${}^0\mathbf{T}_{17} = {}^0\mathbf{T}_{13} {}^{13}\mathbf{T}_{14} {}^{14}\mathbf{T}_{15} {}^{15}\mathbf{T}_{16} {}^{16}\mathbf{T}_{17}$	13→14→15→16→17
Right leg	${}^0\mathbf{T}_{23} = {}^0\mathbf{T}_{19} {}^{19}\mathbf{T}_{20} {}^{20}\mathbf{T}_{21} {}^{21}\mathbf{T}_{22} {}^{22}\mathbf{T}_{23}$	19→20→21→22→23

3.1.3. モーターの速度および角速度の算出

同次変換行列 \mathbf{T} の時間微分 $\dot{\mathbf{T}}$ を求めることで、3次元空間におけるモーターの移動時に生じる速度を計算する。このとき、ある 4×4 の行列 \mathbf{V} を乗算するだけで計算できるとすると、つまり、

$$\dot{\mathbf{T}} = \begin{bmatrix} \dot{\mathbf{R}} & \dot{\mathbf{p}} \\ 0 & 0 & 0 & 0 \end{bmatrix} = \mathbf{V}\mathbf{T} \quad (13)$$

となるためには、

$$\mathbf{V} = \dot{\mathbf{T}}\mathbf{T}^{-1} \quad (14)$$

が成り立つ必要がある。この式から \mathbf{V} を定義する。この \mathbf{V} は空間速度行列と呼ばれ、以下のように表現される:

$$\mathbf{V} = \begin{bmatrix} [\boldsymbol{\omega} \times] & \mathbf{v}_o \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (15)$$

$$\mathbf{v}_o = \dot{\mathbf{p}} - [\boldsymbol{\omega} \times] \mathbf{p} \quad (16)$$

$$\boldsymbol{\omega} = \mathbf{a}\dot{\theta} \quad (17)$$

なお, $\mathbf{v}_o = [v_{ox} \ v_{oy} \ v_{oz}]^T$ は 3×1 の並進速度ベクトル, $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$ は 3×1 の角速度ベクトルで, これらを合わせた $[\mathbf{v}_o^T \ \boldsymbol{\omega}^T]^T$ は空間速度ベクトル^[7]と呼ばれる.

これらを用いて, 第 n 番目のモーターの先端の並進速度 ${}^0\mathbf{v}_{on}$ および角速度 ${}^0\boldsymbol{\omega}_n$ を求めていく. 隣り合うモーターの間の空間速度行列を,

$${}^{n-1}\mathbf{V}_n = \begin{bmatrix} {}^{n-1}[\boldsymbol{\omega} \times]_n & {}^{n-1}\mathbf{v}_{on} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (18)$$

と表す. ここで, ${}^{n-1}[\boldsymbol{\omega} \times]_n$ および ${}^{n-1}\mathbf{v}_{on}$ は $n-1$ 番目のモーターから見たときの n 番目のモーターの先端の相対並進速度および相対角速度である. このとき, 原点から n 番目のモーターの空間速度行列は,

$${}^0\mathbf{V}_n = \begin{bmatrix} {}^0[\boldsymbol{\omega} \times]_n & {}^0\mathbf{v}_{on} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = {}^0\mathbf{V}_{n-1} + {}^0\mathbf{T}_{n-1} {}^{n-1}\mathbf{V}_n {}^0\mathbf{T}_{n-1}^T \quad (19)$$

により求められる. これから, モーターの先端の絶対並進速度 ${}^0\mathbf{v}_{on}$ および絶対角速度 ${}^0[\boldsymbol{\omega} \times]_n$

を求めることができる. なお, ${}^0\boldsymbol{\omega}_n = [{}^0\omega_{xn} \ {}^0\omega_{yn} \ {}^0\omega_{zn}]$ は

$${}^0[\boldsymbol{\omega} \times]_n = \begin{bmatrix} 0 & -{}^0\omega_{zn} & {}^0\omega_{yn} \\ {}^0\omega_{zn} & 0 & -{}^0\omega_{xn} \\ -{}^0\omega_{yn} & {}^0\omega_{xn} & 0 \end{bmatrix} \quad (20)$$

なので, ここから ${}^0\omega_{xn}$, ${}^0\omega_{yn}$, ${}^0\omega_{zn}$ を取り出すことで求めることができる. また, 絶対位置の微分 ${}^0\dot{\mathbf{p}}_n$ および絶対姿勢の微分 ${}^0\dot{\mathbf{R}}_n$ は, 以下の式で求めることができる.

$${}^0\dot{\mathbf{T}}_n = {}^0\mathbf{V}_n {}^0\mathbf{T}_n = \begin{bmatrix} {}^0\dot{\mathbf{R}}_n & {}^0\dot{\mathbf{p}}_n \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (21)$$

同様にして、重心の同次変換行列 \mathbf{T}_g に対する空間速度行列 \mathbf{V}_g を定義すると、

$$\dot{\mathbf{T}}_g = \begin{bmatrix} \mathbf{0} & \dot{\mathbf{p}}_g \\ 0 & 0 \end{bmatrix} = \mathbf{V}_g \mathbf{T}_g \quad (22)$$

となり、第 n 番目のモーターの重心の位置の微分 ${}^0\dot{\mathbf{p}}_{g_n}$ は、

$${}^{n-1}\mathbf{V}_{g_n} = \begin{bmatrix} \mathbf{0} & {}^{n-1}\dot{\mathbf{p}}_{g_n} \\ \mathbf{0} & 0 \end{bmatrix} \quad (23)$$

を隣り合うモーターの間の重心の空間速度行列をしたとき、

$${}^0\mathbf{V}_{g_n} = \begin{bmatrix} \mathbf{0} & {}^0\dot{\mathbf{p}}_{g_n} \\ \mathbf{0} & 0 \end{bmatrix} = {}^0\mathbf{V}_{n-1} + {}^0\mathbf{T}_{n-1} {}^{n-1}\mathbf{V}_{g_n} {}^0\mathbf{T}_{n-1}^T \quad (24)$$

と求めることができる。なお、重心の同次変換行列の微分 ${}^0\dot{\mathbf{T}}_{g_n}$ は、 ${}^0\mathbf{V}_{g_n}$ と ${}^0\mathbf{T}_{g_n}$ を乗算して

も結局 ${}^0\mathbf{V}_{g_n}$ と変わらない。

$${}^0\dot{\mathbf{T}}_{g_n} = {}^0\mathbf{V}_{g_n} {}^0\mathbf{T}_{g_n} = {}^0\mathbf{V}_{g_n} \quad (25)$$

3.1.4. モーターの加速度および角加速度の算出

$\dot{\mathbf{T}}$ をもう 1 回時間で微分することで、3 次元空間におけるモーターの移動時に生じる加速度および角加速度を計算できるとすると、

$$\ddot{\mathbf{T}} = \begin{bmatrix} \ddot{\mathbf{R}} & \ddot{\mathbf{p}} \\ 0 & 0 \end{bmatrix} = \frac{d}{dt}(\mathbf{V}\mathbf{T}) = \dot{\mathbf{V}}\mathbf{T} + \mathbf{V}\dot{\mathbf{T}} = \dot{\mathbf{V}}\mathbf{T} + \mathbf{V}(\mathbf{V}\mathbf{T}) = (\dot{\mathbf{V}} + \mathbf{V}\mathbf{V})\mathbf{T} \quad (26)$$

となる。ここで、 $\mathbf{A} = \dot{\mathbf{V}}$ を 3×3 の空間加速度行列と定義すると：

$$\mathbf{A} = \dot{\mathbf{V}} = \begin{bmatrix} [\dot{\boldsymbol{\omega}} \times] & \dot{\mathbf{v}}_o \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (27)$$

$$\dot{\mathbf{v}}_o = \ddot{\mathbf{p}} - [\boldsymbol{\omega} \times] \dot{\mathbf{p}} - [\dot{\boldsymbol{\omega}} \times] \mathbf{p} \quad (28)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{a} \ddot{\theta} \quad (29)$$

なお, $\dot{\mathbf{v}}_o = [\dot{v}_{ox} \ \dot{v}_{oy} \ \dot{v}_{oz}]^T$ は 3×1 の並進加速度ベクトル, $\dot{\boldsymbol{\omega}} = [\dot{\omega}_x \ \dot{\omega}_y \ \dot{\omega}_z]^T$ は 3×1 の角加速度ベクトルで, これらを合わせた $[\dot{\mathbf{v}}_o^T \ \dot{\boldsymbol{\omega}}^T]^T$ は空間加速度ベクトル^[7]と呼ばれる. この表記から, 第 n 番目のモーターの先端の絶対加速度 ${}^0\dot{\mathbf{v}}_{on}$ は, 隣り合うモーターの間の空間加速度行列を,

$${}^{n-1}\mathbf{A}_n = {}^{n-1}\dot{\mathbf{V}}_n = \begin{bmatrix} {}^{n-1}[\dot{\boldsymbol{\omega}} \times]_n & {}^{n-1}\dot{\mathbf{v}}_{on} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (30)$$

と定義したとき,

$${}^0\mathbf{A}_n = \begin{bmatrix} {}^0[\dot{\boldsymbol{\omega}} \times]_n & {}^0\dot{\mathbf{v}}_{on} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = {}^0\mathbf{A}_{n-1} + {}^0\mathbf{T}_{n-1} {}^{n-1}\mathbf{A}_n {}^0\mathbf{T}_{n-1}^T + {}^0\mathbf{V}_{n-1} {}^0\mathbf{V}_n - {}^0\mathbf{V}_n {}^0\mathbf{V}_{n-1} \quad (31)$$

のように計算できる. また, モーターの先端の絶対角加速度 ${}^0\dot{\boldsymbol{\omega}}_n$ は, 前式の

$${}^0[\dot{\boldsymbol{\omega}} \times]_n = \begin{bmatrix} 0 & -{}^0\dot{\omega}_{zn} & {}^0\dot{\omega}_{yn} \\ {}^0\dot{\omega}_{zn} & 0 & -{}^0\dot{\omega}_{xn} \\ -{}^0\dot{\omega}_{yn} & {}^0\dot{\omega}_{xn} & 0 \end{bmatrix} \quad (32)$$

から ${}^0\dot{\omega}_{xn}$, ${}^0\dot{\omega}_{yn}$, ${}^0\dot{\omega}_{zn}$ を取り出すことで求めることができる. また, 絶対位置の 2 回微分 ${}^0\ddot{\mathbf{p}}_n$ および絶対姿勢の 2 回微分 ${}^0\ddot{\mathbf{R}}_n$ は, 以下の式で求めることができる.

$${}^0\ddot{\mathbf{T}}_n = ({}^0\mathbf{A}_n + {}^0\mathbf{V}_n {}^0\mathbf{V}_n) {}^0\mathbf{T}_n = \begin{bmatrix} {}^0\ddot{\mathbf{R}}_n & {}^0\ddot{\mathbf{p}}_n \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (33)$$

同様にして、重心の空間加速度行列 \mathbf{A}_g を、

$$\mathbf{A}_g = \dot{\mathbf{V}}_g = \begin{bmatrix} \mathbf{0} & \ddot{\mathbf{p}}_g \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (34)$$

と定義すると、第 n 番目のモーターの重心加速度 ${}^0\ddot{\mathbf{p}}_{g_n}$ は、隣り合うモーターの間の重心の空間加速度行列を、

$${}^{n-1}\mathbf{A}_{g_n} = {}^{n-1}\dot{\mathbf{V}}_{g_n} = \begin{bmatrix} \mathbf{0} & {}^{n-1}\ddot{\mathbf{p}}_{g_n} \\ \mathbf{0} & 0 \end{bmatrix} \quad (35)$$

と定義すると、

$${}^0\mathbf{A}_{g_n} = \begin{bmatrix} \mathbf{0} & {}^0\ddot{\mathbf{p}}_{g_n} \\ \mathbf{0} & 0 \end{bmatrix} = {}^0\mathbf{A}_{n-1} + {}^0\mathbf{T}_{n-1} {}^{n-1}\mathbf{A}_{g_n} {}^0\mathbf{T}_{n-1}^T + {}^0\mathbf{V}_{n-1} {}^0\mathbf{V}_{g_n} - {}^0\mathbf{V}_{g_n} {}^0\mathbf{V}_{n-1} \quad (36)$$

より求めることができる。

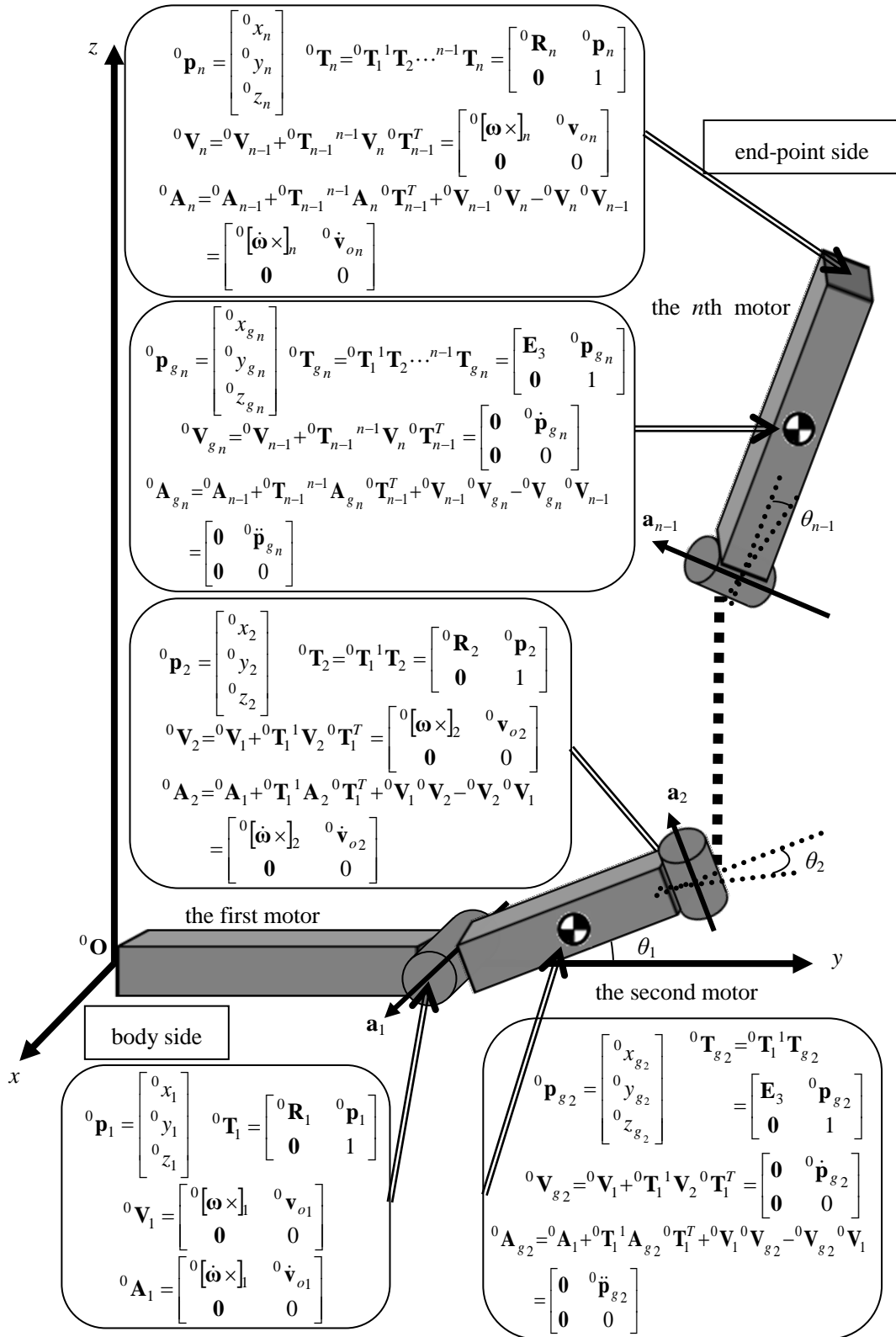


Fig. 5 Kinematics of KHR-1 on three-dimensional space

3.1.5. モーターの力およびモーメントの算出

次に、力とモーメントについて考える。空間速度ベクトルによる剛体の運動方程式を次式に示す。

$$\begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix} = \mathbf{I}^S \begin{bmatrix} \dot{\mathbf{v}}_o \\ \dot{\boldsymbol{\omega}} \end{bmatrix} + \begin{bmatrix} [\boldsymbol{\omega} \times] & \mathbf{0} \\ [\mathbf{v}_o \times] & [\boldsymbol{\omega} \times] \end{bmatrix} \begin{bmatrix} \mathbf{P} \\ \mathbf{L} \end{bmatrix} \quad (37)$$

ここで、 \mathbf{I}^S は以下のように定義される 6×6 の対称行列であり、空間慣性行列^[7]と呼ばれている。

$$\mathbf{I}^S \equiv \begin{bmatrix} m\mathbf{E}_3 & m[\mathbf{p}_g \times]^T \\ m[\mathbf{p}_g \times] & m[\mathbf{p}_g \times][\mathbf{p}_g \times]^T + \mathbf{I} \end{bmatrix} \quad (38)$$

ここで、 m はモーターの質量、 \mathbf{I} はモーターの慣性テンソルのことである。モーターの形状を、縦が l_x 、横が l_y 、高さが l_z の一様密度の長方形と仮定したとき、次式のようになる：

$$\mathbf{I} = \begin{bmatrix} \frac{m}{12}(l_y^2 + l_z^2) & 0 & 0 \\ 0 & \frac{m}{12}(l_x^2 + l_z^2) & 0 \\ 0 & 0 & \frac{m}{12}(l_x^2 + l_y^2) \end{bmatrix} \quad (39)$$

また、 \mathbf{P} 、 \mathbf{L} はそれぞれモーターの運動量・角運動量であり、以下の関係が成り立つ。

$$\begin{bmatrix} \mathbf{P} \\ \mathbf{L} \end{bmatrix} = \mathbf{I}^S \begin{bmatrix} \mathbf{v}_o \\ \boldsymbol{\omega} \end{bmatrix} \quad (40)$$

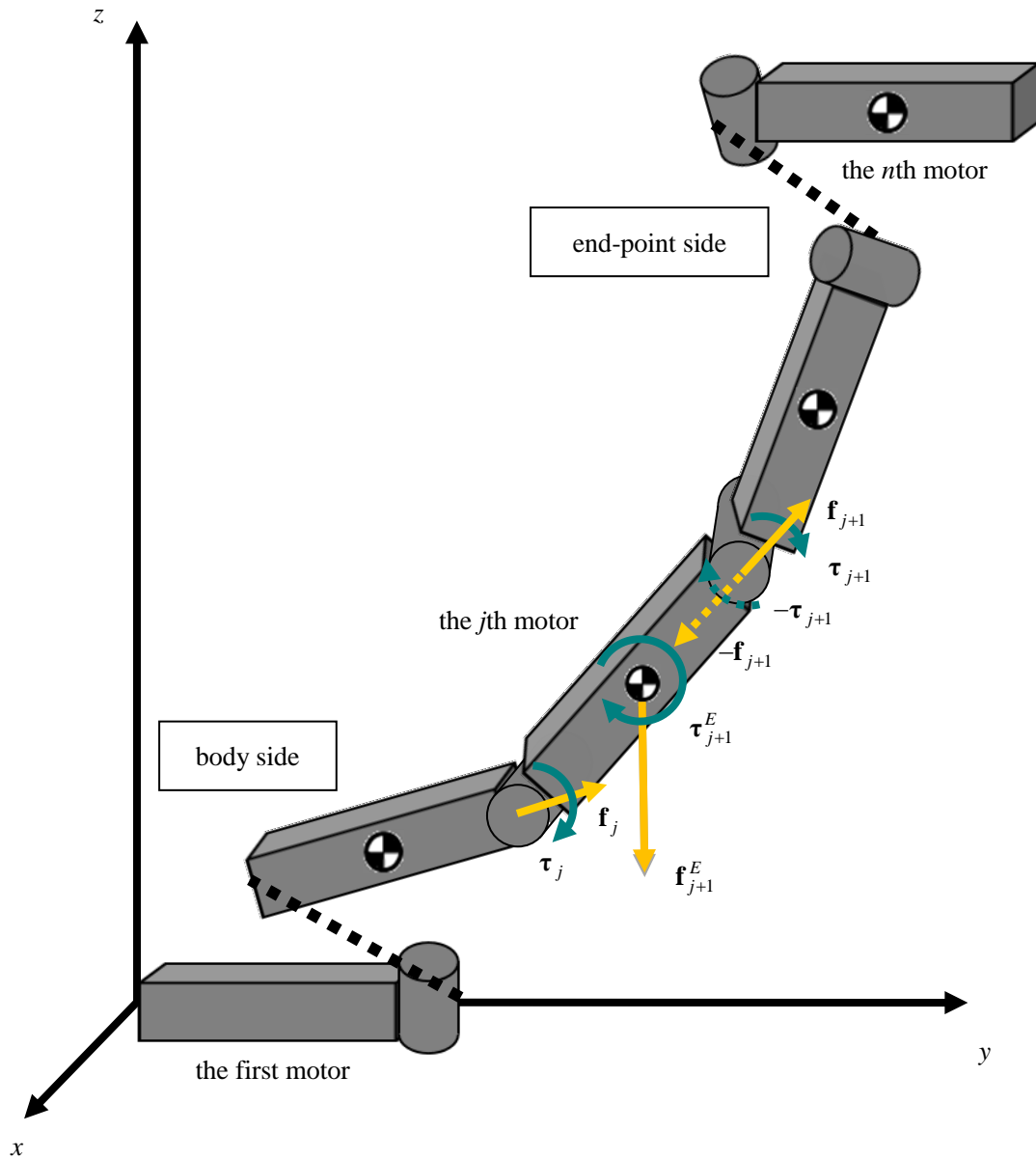
Fig. 6 Force and moment that acts on the arm j

Fig. 6 に、第 j 番目のモーターに作用する力とモーメントを示す。ここで、胴体側から作用する力とモーメントをそれぞれ \mathbf{f}_j , $\boldsymbol{\tau}_j$ とする。また、第 j モーターに直接作用する力とモーメント (今回は、重心に作用する重力) をそれぞれ \mathbf{f}_{j+1}^E , $\boldsymbol{\tau}_{j+1}^E$ とする。胴体側からの作用、環境からの作用、そして、手足、足先側からの作用を合計したものが、第 j モーターの運動を引き起こす。すなわち、運動方程式は、

$$\begin{bmatrix} \mathbf{f}_j \\ \boldsymbol{\tau}_j \end{bmatrix} + \begin{bmatrix} \mathbf{f}_{j+1}^E \\ \boldsymbol{\tau}_{j+1}^E \end{bmatrix} - \begin{bmatrix} \mathbf{f}_{j+1} \\ \boldsymbol{\tau}_{j+1} \end{bmatrix} = \mathbf{I}_j^S \begin{bmatrix} \dot{\mathbf{v}}_{o_j} \\ \dot{\boldsymbol{\omega}}_j \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}_j \times \\ \mathbf{v}_{o_j} \times \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\omega}_j \times \end{bmatrix} \begin{bmatrix} \mathbf{P}_j \\ \mathbf{L}_j \end{bmatrix} \quad (41)$$

となり，これを書き換えることにより，モーターを越えて伝わる力とモーメントの漸化式を得る．

$$\begin{bmatrix} \mathbf{f}_j \\ \boldsymbol{\tau}_j \end{bmatrix} = \mathbf{I}_j^S \begin{bmatrix} \dot{\mathbf{v}}_{o_j} \\ \dot{\boldsymbol{\omega}}_j \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}_j \times \\ \mathbf{v}_{o_j} \times \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\omega}_j \times \end{bmatrix} \begin{bmatrix} \mathbf{P}_j \\ \mathbf{L}_j \end{bmatrix} - \begin{bmatrix} \mathbf{f}_{j+1}^E \\ \boldsymbol{\tau}_{j+1}^E \end{bmatrix} + \begin{bmatrix} \mathbf{f}_{j+1} \\ \boldsymbol{\tau}_{j+1} \end{bmatrix} \quad (42)$$

j がモーターの末端であるとき， \mathbf{f}_{j+1} ， $\boldsymbol{\tau}_{j+1}$ は 0 となり， \mathbf{f}_j ， $\boldsymbol{\tau}_j$ が計算できる．従って，ロボットの末端から根元に向かって順番に計算をすすめることで，全関節に関する力とモーメントを知ることができる．

また，各関節軸で発生するトルク u_j は，

$$u_j = \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_j \times \mathbf{a}_j \\ \mathbf{a}_j \end{bmatrix}^T \begin{bmatrix} \mathbf{f}_j \\ \boldsymbol{\tau}_j \end{bmatrix} \quad (43)$$

で求めることができる．

3.2. ハードウェアでの接続環境

ハードウェアでの接続環境は，KHR-1 と MATLAB が動作する PC を，RS232C ケーブルを介して接続し (Fig. 7)，MATLAB の figure 上のスライダーを動かすことによって，KHR-1 が動作できるような環境を構築した．

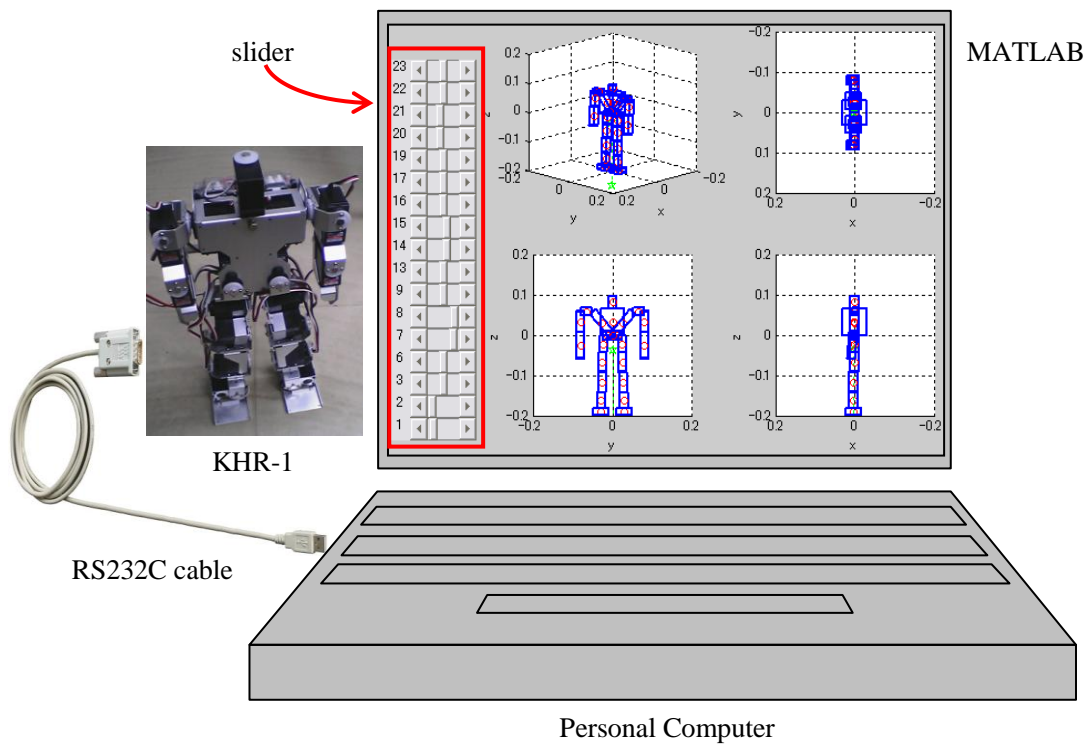


Fig. 7 Hardware association engine

4.2 足歩行ロボットの歩行動作の検証

4.1. シミュレーションモデルの構築

まず, Table 1, Fig. 4 の情報をもとに, シミュレーションモデルの構築を行った.

ODE では, モデルの簡略化のために, モーター, 足の裏, 基盤の部品に, 3 つの長方形に近似してサイズと質量の情報を与えた (Table 4). そして, KHR-1 に設定したホームポジション (Table 5) を入力したときの関節の位置と関節軸, 物体の重心位置を指定することで, KHR-1 の形状を実現した (Fig. 8(a)).

Table 4 Specifications of the KHR-1 model on ODE

	Height	Weight
Motor	41×35×21[mm]	45[g]
Sole	80×45×1[mm]	2[g]
Board	35×70×45[mm]	25[g]

Table 5 Home position of the KHR-1

Part	Left arm			Face	Right arm					
ID	1	2	3	6	7	8	9			
degree[deg]	5	0	90	90	175	180	90			
Part	Left leg					Right leg				
ID	13	14	15	16	17	19	20	21	22	23
degree[deg]	88	115	115	90	92	92	65	65	90	88

MATLAB では, 関節の相対位置および関節軸を設定し, 重心を相対位置の半分とすることで実現した (Fig. 8(b)).

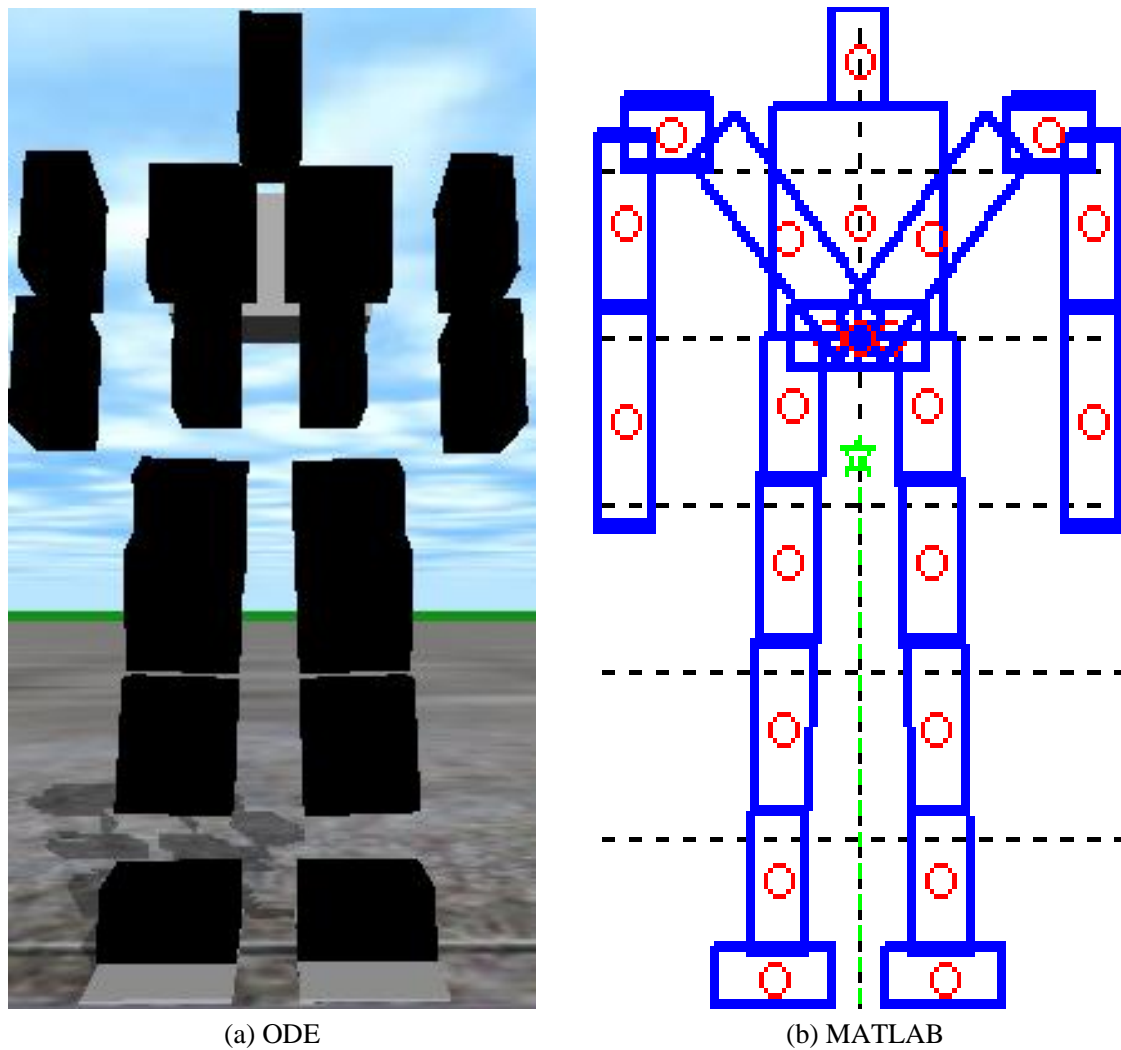


Fig. 8 The simple models of the KHR-1

4.2. 2足歩行ロボットの歩行シミュレーションおよび歩行実験

KHR-1 のサンプルモーション^[8]の csv ファイルから関節の角度データを読み込み、ODE と MATLAB で歩行制御のシミュレーションを行った。

4.2.1. ODE でのシミュレーション

ODE でのシミュレーションの際、 $K_p=5$ 、 $T_i=0.5$ 、 $T_d=0.1$ と設定して行った。その様子を Fig. 9 に示す。



Fig. 9 Simulation screen of walking control by using ODE

4.2.2. MATLAB でのシミュレーション

MATLAB でのシミュレーションのフローチャートを Fig. 10 に示す.

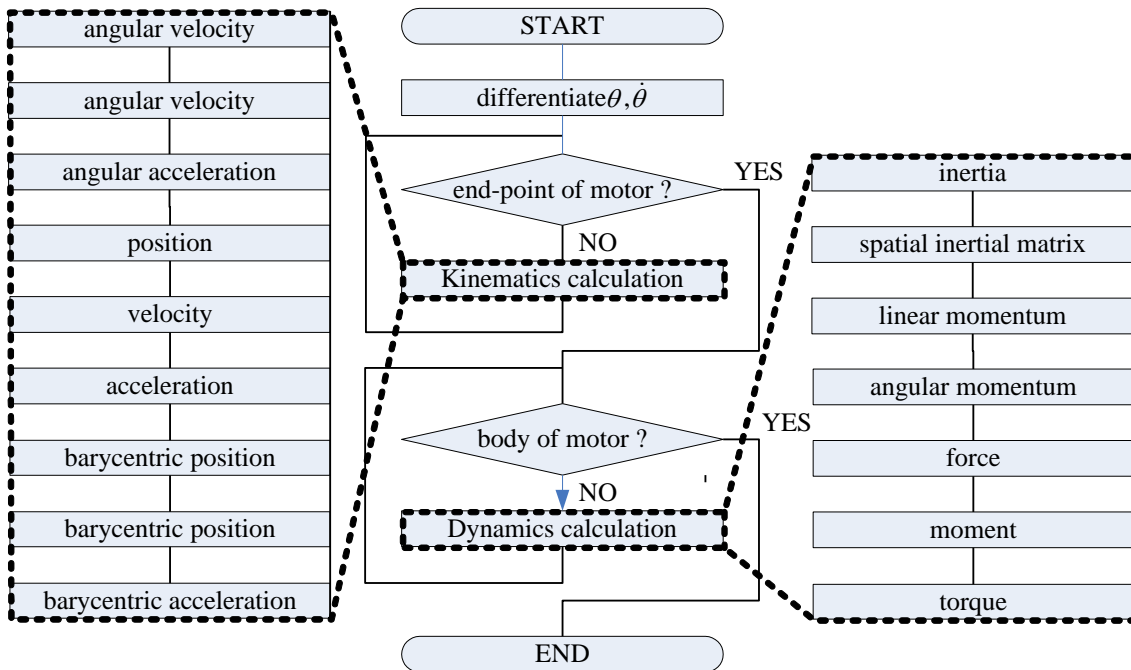


Fig. 10 The flow chart in MATLAB simulation

既知な値は初期の位置 \mathbf{p} と姿勢 \mathbf{R} およびサンプルモーションの角度 θ である。これを用いて、運動学計算および逆動力学計算を行うことでロボットの運動を実現する。

- 運動学計算

- ① θ を次式で数値微分して $\dot{\theta}$, $\ddot{\theta}$ を計算する。

$$\dot{\theta} \cong \frac{\theta(k+1) - \theta(k)}{dt}, \quad k = 1, 2, \dots \quad (44)$$

$$\ddot{\theta} \cong \frac{\dot{\theta}(k+1) - \dot{\theta}(k)}{dt}, \quad k = 1, 2, \dots \quad (45)$$

- ② 角度 θ , 角速度 $\boldsymbol{\omega}$, 角加速度 $\dot{\boldsymbol{\omega}}$ を更新する。
- ③ 同次変換行列 \mathbf{T} , \mathbf{T}_g , 空間速度行列 \mathbf{V} , \mathbf{V}_g , 空間加速度行列 \mathbf{A} , \mathbf{A}_g を使い, 位置 \mathbf{p} , \mathbf{p}_g , 速度 \mathbf{v}_o , $\dot{\mathbf{p}}_g$, 加速度 $\dot{\mathbf{v}}_o$, $\ddot{\mathbf{p}}_g$ を更新する。
- ④ ②, ③を与えられた角度ごとで, モーターの根元から末端へと繰り返す。

- 動力学計算

- ① 運動学計算によって姿勢が更新されたため, それに応じた慣性テンソル \mathbf{I}_n を以下の式で更新する。

$$\mathbf{I}_n = \mathbf{R} \mathbf{I} \mathbf{R}^T \quad (46)$$

- ② 運動学計算で得た ${}^0\mathbf{p}_{g_n}$ を使い, 空間慣性行列 \mathbf{I}^S を更新
- ③ 運動量 \mathbf{P} , 角運動量 \mathbf{L} を更新
- ④ 力 \mathbf{f} , モーメント $\boldsymbol{\tau}$ を更新
- ⑤ 関節トルク \mathbf{u} を更新
- ⑥ ②, ③を与えられた角度ごとで, モーターの末端から根元へと繰り返す。

MATLAB でのシミュレーションのモーションおよび結果を Fig. 11 に示す。

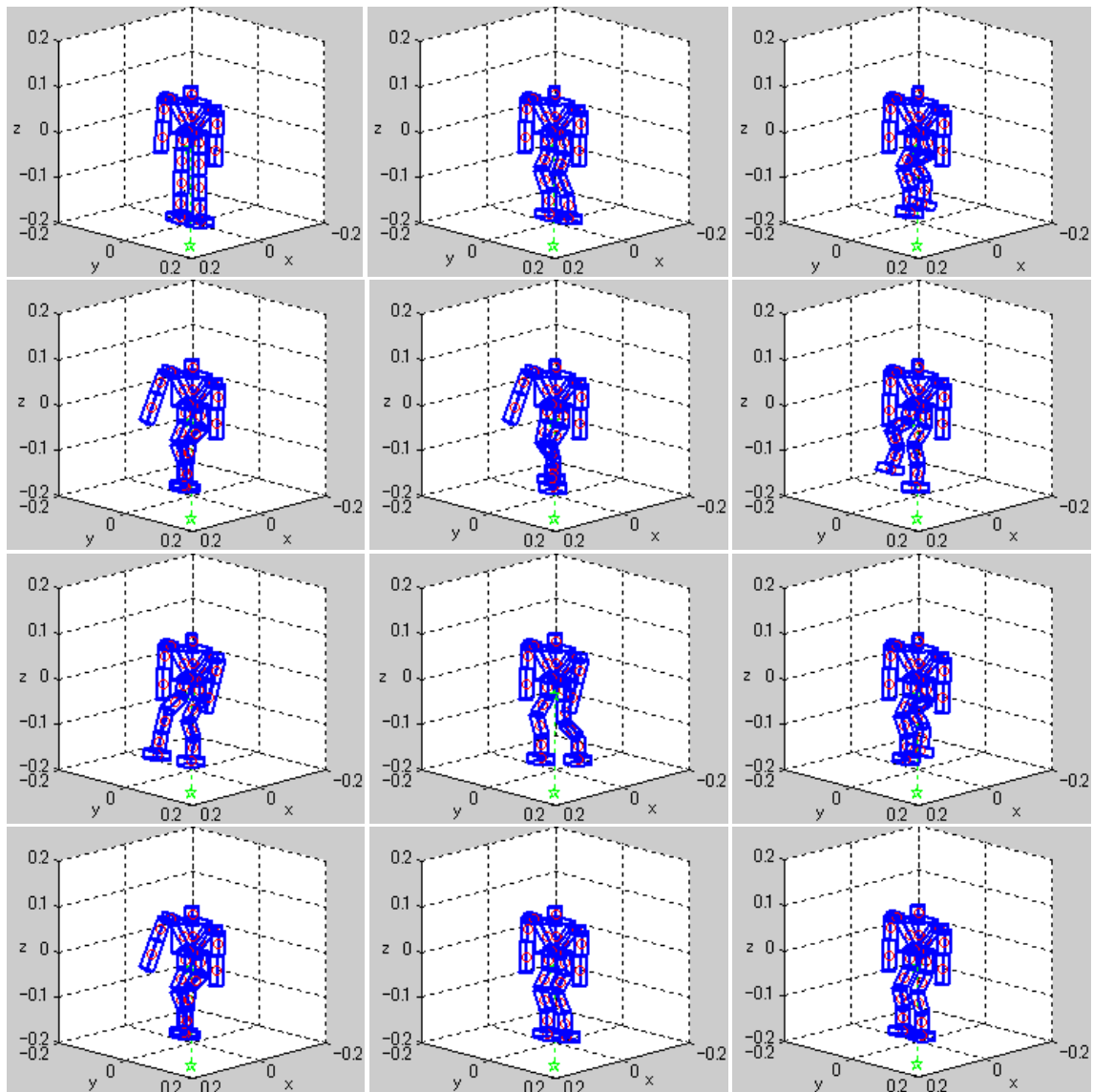


Fig. 11 Walking motion of KHR-1 by using MATLAB

また、サンプルモーションごとの右足の先端の絶対位置 ${}^0\mathbf{p}_n$ 、絶対速度 ${}^0\mathbf{v}_{on}$ 、絶対加速度 ${}^0\dot{\mathbf{v}}_{on}$ 、絶対角速度 ${}^0\boldsymbol{\omega}_n$ 、絶対角加速度 ${}^0\dot{\boldsymbol{\omega}}_n$ 、重心の絶対位置 ${}^0\mathbf{p}_{g_n}$ 、絶対速度 ${}^0\dot{\mathbf{p}}_{g_n}$ 、絶対加速度 ${}^0\ddot{\mathbf{p}}_{g_n}$ を Fig. 12 に、運動量 \mathbf{P}_n 、角運動量 \mathbf{L}_n 、力 \mathbf{f}_n 、モーメント $\boldsymbol{\tau}_n$ 、トルク \mathbf{u}_n を Fig. 13 に示す。なお、横軸は時間である。

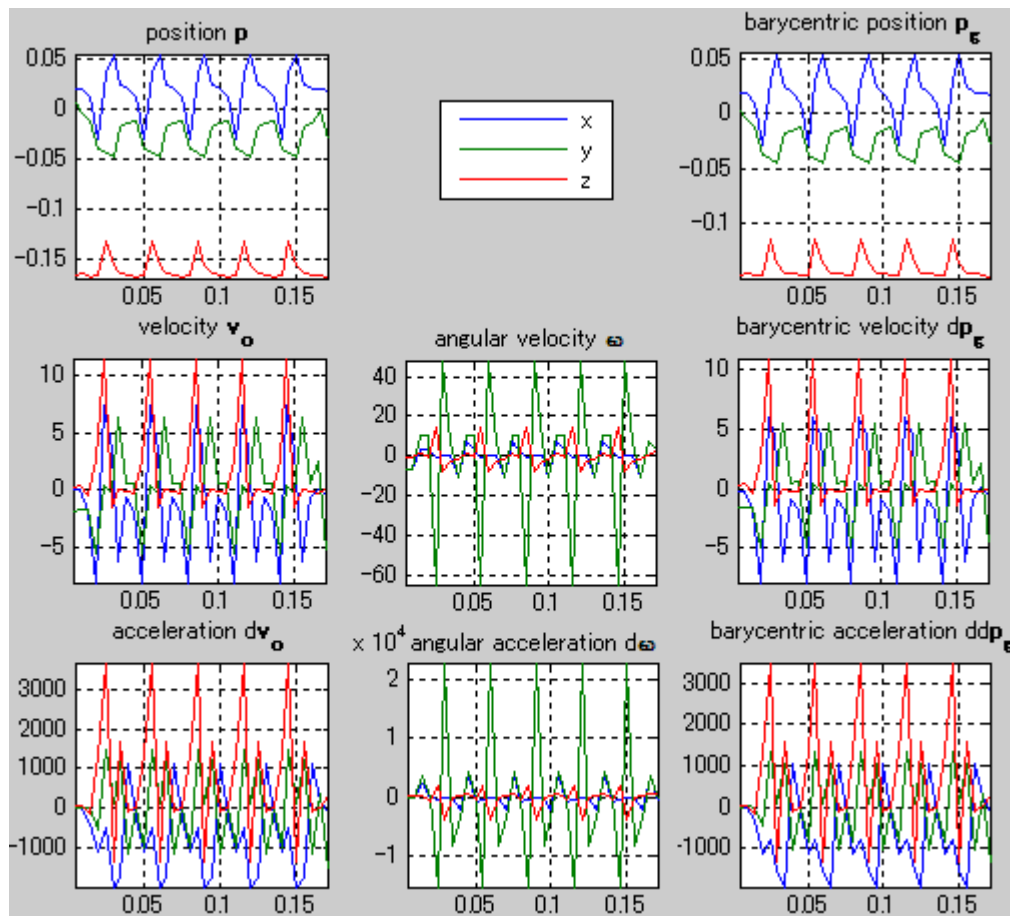


Fig. 12 Kinematics parameters of simulation results by using MATLAB (end-point of the right leg)

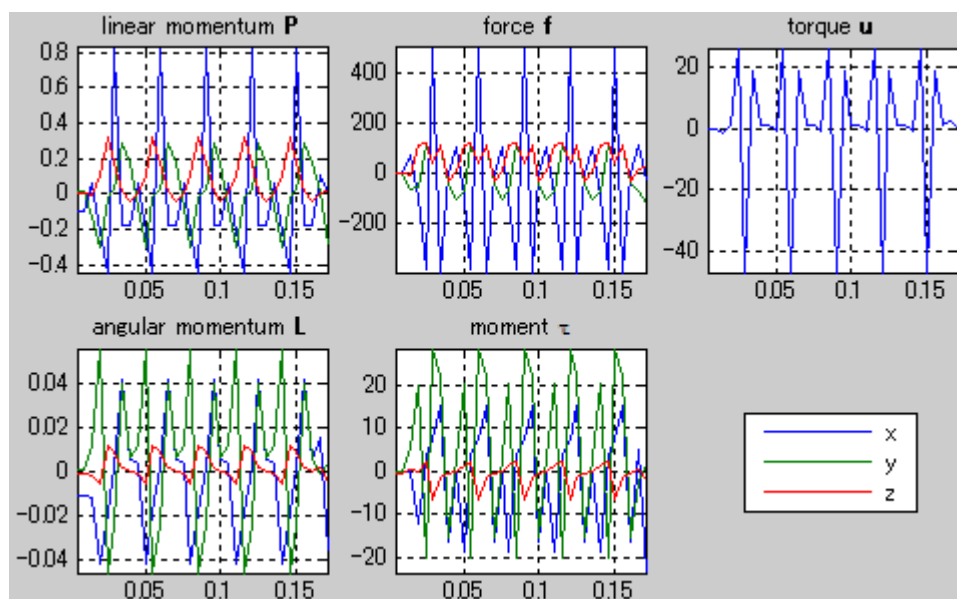


Fig. 13 Dynamics parameters of simulation results by using MATLAB (end-point of the right leg)

4.2.3. KHR-1 での検証

3.2 節の接続を用い、MATLAB のプログラムから KHR-1 が動作するよう実行した。

4.3. 結果および考察

ODE と MATLAB でのシミュレーションにおいて、転倒することなく歩行確認できた。しかし、ロボットの歩行動作自体は決して安定したものとはいえなかった。

これは、関節の角度を動かすタイミングを明確に定めていないことが考えられる。ODE では特にシミュレーション時にシミュレーションステップや dt の値、PID 制御のパラメータを調整しながら行わないと転倒してしまう事態が起きてしまう。MATLAB では、力やモーメントが現実よりも大きな値が出力されてしまう。

KHR-1 での検証も行ったところ、サンプルモーションを使っても転倒してしまうことがあった。これは、シミュレーションモデルの精度を上げることや、摩擦など、シミュレーション環境と実環境を考慮していくことで解消できるであろうと考えられる。

5. むすび

異なるタイプのシミュレータにおいて、独立しているシミュレーションモデルの構築およびシミュレーションは、ODE と MATLAB を使用することによって、歩行動作で転倒することない範囲で誤差を減らすことができた。また、ODE と MATLAB 双方のシミュレーション、物理的なシミュレーションの整合性を確認することができたことがわかる。

謝辞

まず、本研究を進めるにあたり、ご多忙の中私のためにご助言、ご指導をいただいた渡辺 嘉二郎教授、小林 一行准教授に深く感謝いたします。

小林先生には、実験方法の提案、論文の構成などの技術的な面だけでなく、基本的なことまで丁寧にご教示いただきました。SICE の国際発表のときには、何度もくじけそうになることがあり投げ出したいときもありましたが、発表の前日の夜遅くまで発表原稿を訂正していただきました。重ねて深く感謝します。

また、同じ小林研究室で3年間一緒だった、修士2年生の柴田 諭君、御園 祐介君には研究以外のことで特に迷惑をかけることが多かったと思いますが、何も言わずに優しく接していただいて感謝しております。また、研究のことで何気ない質問にも優しく答えていただいた博士1年生の大久保 友幸さんにも感謝いたします。そして、渡辺研究室・小林研究室の先輩・同期・後輩の皆様にも大変感謝しております。

最後になりましたが、帰宅時間が遅くなることも多い中、就職活動のときも含めて、サポートしていただいた両親に心より深く感謝いたします。

こんな未熟者の私ですが、本当にみなさまのサポートのおかげでここまで来ることができました。ありがとうございました。

付録

本論文に使用したプログラムおよび実験風景の動画は、別途の CD-R を添付する。フォルダの構成およびプログラムの概要を Fig. 14 に示す。

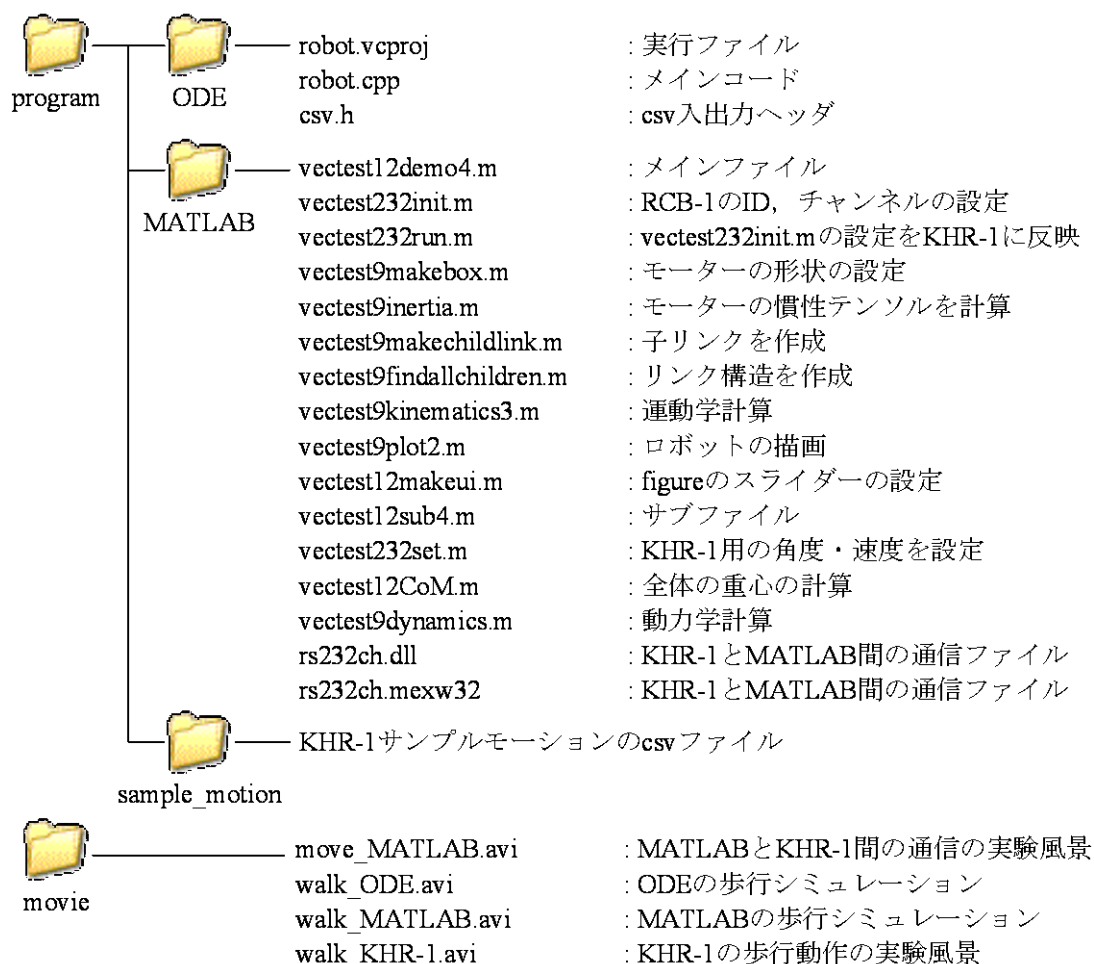


Fig. 14 Composition of folder and outline of program

また、MATLAB のプログラムでの数式と属性の関係を Table 6 に示す。

Table 6 Parameters in MATLAB simulation

Property of motor	Mathematical expression	Attribute	Size
ID	n	i	
End-point side ID	$n-1$	parent	

Body side ID	$n + 1$	children	
Size	ℓ_n	arm(i).xyz	3×1
ID and channel of RCB-1		arm(i).idch	2×1
Degree	θ_n	arm(i).th	1×1
Axis vector	\mathbf{a}_n	arm(i).a	3×1
Relative position	${}^{n-1}\mathbf{p}_n$	arm(i).l	3×1
Relative barycentric position	${}^{n-1}\mathbf{p}_{g_n}$	arm(i).lg	3×1
Relative attitude	${}^{n-1}\mathbf{R}_n$	arm(i).R	3×3
Relative barycentric attitude	${}^{n-1}\mathbf{R}_{g_n}$	arm(i).Rg	3×3
Absolute position	${}^0\mathbf{p}_n$	arm(i).p	3×1
Absolute barycentric position	${}^0\mathbf{p}_{g_n}$	arm(i).pg	3×1
Homogeneous transformation matrix	${}^0\mathbf{T}_n$	arm(i).T	4×4
The center of gravity of homogeneous transformation matrix	${}^0\mathbf{T}_{g_n}$	arm(i).Tg	4×4
Differential of degree	$\dot{\theta}_n$	arm(i).dth	1×1
Differential of relative position	${}^{n-1}\dot{\mathbf{p}}_n$	arm(i).dl	3×1
Differential of relative barycentric position	${}^{n-1}\dot{\mathbf{p}}_{g_n}$	arm(i).dlg	3×1
Differential of relative attitude	${}^{n-1}\dot{\mathbf{R}}_n$	arm(i).dR	3×3
Absolute translational velocity	${}^0\mathbf{v}_{on}$	arm(i).vo	3×1
The center of gravity of absolute translational velocity	${}^0\mathbf{v}_{og_n}$	arm(i).vog	3×1
Absolute angular velocity	${}^0\boldsymbol{\omega}_n$	arm(i).w	3×1
Spatial velocity matrix	${}^0\mathbf{V}_n$	arm(i).V	4×4
The center of gravity of spatial velocity matrix	${}^0\mathbf{V}_{g_n}$	arm(i).Vg	4×4
Differential of homogeneous transformation matrix	${}^0\dot{\mathbf{T}}_n$	arm(i).dT	4×4
Differential of the center of gravity of homogeneous transformation matrix	${}^0\dot{\mathbf{T}}_{g_n}$	arm(i).dTg	4×4
Second-order differential of degree	$\ddot{\theta}_n$	arm(i).ddth	1×1

Second-order differential of relative position	${}^{n-1}\ddot{\mathbf{p}}_n$	arm(i).ddl	3×1
Second-order differential of relative barycentric attitude	${}^{n-1}\ddot{\mathbf{p}}_{g_n}$	arm(i).ddl _g	3×1
Second-order differential of relative attitude	${}^{n-1}\ddot{\mathbf{R}}_n$	arm(i).ddR	3×3
Absolute translational acceleration	${}^0\dot{\mathbf{v}}_{o_n}$	arm(i).dvo	3×1
The center of gravity of absolute translational acceleration	${}^0\dot{\mathbf{v}}_{o_{g_n}}$	arm(i).dvog	3×1
Absolute angular acceleration	${}^0\dot{\boldsymbol{\omega}}_n$	arm(i).dw	3×1
Spatial acceleration matrix	${}^0\mathbf{A}_n$	arm(i).A	4×4
Differential of the center of gravity of spatial acceleration matrix	${}^0\mathbf{A}_{g_n}$	arm(i).Ag	4×4
Second-order differential of homogeneous transformation matrix	${}^0\dot{\mathbf{T}}_n$	arm(i).ddT	4×4
Second-order differential of the center of gravity of homogeneous transformation matrix	${}^0\dot{\mathbf{T}}_{g_n}$	arm(i).ddT _g	4×4
Mass	m_n	arm(i).m	1×1
Inertia tensor	\mathbf{I}_n	arm(i).I	3×3
Linear momentum	\mathbf{P}_n	arm(i).P	3×1
Angular momentum	\mathbf{L}_n	arm(i).L	3×1
Force	\mathbf{f}_n	arm(i).f	3×1
Moment	$\boldsymbol{\tau}_n$	arm(i).t	3×1
torque	\mathbf{u}_n	arm(i).u	3×1

また，本論文で使用したソフトウェアのバージョンは Table 7 のようになっている。

Table 7 Software version

Open Dynamics Engine	ODE 0.10.1
C compiler	Microsoft Visual C++ Express 2008
MATLAB	MATLAB7.1

研究業績

[国際会議]

- 1) Yuhi Awa and Kazuyuki Kobayashi
Kinematics Simulation by Using ODE and MATLAB
SICE Annual Conference 2007, pp.3090-3094, September (2007)

[国内会議]

- 1) 栗 雄飛, 渡辺 嘉二郎, 小林 一行
複数シミュレータによる2足歩行ロボットにおけるシミュレーションの検証
第53回知的制御研究会(SIC08-1), pp.17-20, July (2008)

参考文献

- ¹ 製品紹介キット KHR-1, http://www.kondo-robot.com/html/Product_main.html
- ² Kondo Items Servos, http://www.kondo-robot.com/html/Products_Servo.html
- ³ Open Dynamics Engine V0.5 User Guide, <http://ode.org/ode-latest-userguide.pdf>
- ⁴ Anitescu M., Potra F.A., Stewart D.E., Time-Stepping for Three-Dimensional Rigid Body Dynamics, *Comp. Methods Appl. Mech. Eng.*, 177/3-4: 183-197, 1998
- ⁵ Anitescu M., Potra F.A., Formulating Dynamic Multi-Rigid-Body Problems with Friction as Solvable Linear Complementarity Problems, *Reports on Computational Mathematics*, 93, 1997
- ⁶ 出村 公成, 簡単!実践!ロボットシミュレーション, 2007
- ⁷ 梶田 秀司, ヒューマノイドロボット, 2005
- ⁸ KHR-1 サンプルモーション, <http://www.kondo-robot.com/guide/samplemotion-khr-1.html>