

動力学シミュレータベース動作探索による 脚収納外装枠を備えた等身大二脚ロボットの起き上がり動作の実現

○グエンキムゴックカン 野田晋太郎 小椎尾侑多 小島邦生

菅井文仁 垣内洋平 岡田慧 稲葉雅幸 (東京大学)

Dynamic-Simulation based Motion Generation: Realization of Self-righting Motion on Life-sized Legged Robot equipped with Exterior Frame

○ Kim-Ngoc-Khanh NGUYEN, Shintaro NODA, Yuta KOJIO, Kunio KOJIMA,
Fumihito SUGAI, Yohei KAKIUCHI, Kei OKADA, Masayuki INABA (The University of Tokyo)

Abstract : In this paper, we propose a method for generating dynamic motion, which consider motion trajectory as a multi-variable function of time and robot's posture. Our motion trajectory generation method is optimized in dynamic simulator under adding disturbance condition by evolutionary search algorithm. It enables us to achieve self-righting motion after 1 hour searching time, and does not take any manual motion modified by human. Our proposed method is also applied to a life-sized legged robot equipped with exterior frame. The experiment in actual life-sized robot showed that the self-righting motion generated by our proposed method can be finished within 3 seconds.

1. 序論

ヒューマノイドは人間と類似形を持つため、人間と共存し、人間の生活支援、代わりに作業すること、人間との協調作業が期待される。しかし、ヒューマノイドを活用する際の大きな課題の一つは2脚での立脚状態において、重心の高さに対して、台車ロボットや四足ロボットに比べ足の支持領域が小さいため、転倒の危険性が高いことである。そのため、これまではいかに動作を安定化して、作業中に転倒させないかという方向で、数多くの研究が行われてきた。しかし、人間が普段活動している環境では、整備されていない環境や人間との接触、故障など予測困難な外乱によって、転倒を完全に防ぐのは不可能である。現場では継続的に動作することを求められるため、万が一転倒してしまっても、人間の介入無く起き上がることが要求される。

ヒューマノイドの転倒回復には、静的な起き上がり動作を実現した例がいくつか存在する^{1), 2)}。しかし、それらの起き上がり動作は姿勢安定性を常に維持するため、動作完了までに、等身大ヒューマノイドでは40秒以上かかる。人間の運動計測に基づいて、解析に基づく成功条件の部分的決定と動力学シミュレーションに基づく成功条件の探索により、動的な起き上がり動作を実現した例もある³⁾。しかし、これは単純化した力学モデルとシミュレーション上の解析に基づくため、シミュレーション軌道から実機実験を試行錯誤しながら、手動で軌道を調整して成功軌道を探索する必要がある。このプロセスは多くの時間と労力が必要で、さらに開発者の経験への依存性が高い。

本研究では、ロボットの動作軌道を時間とロボットの姿

勢の多変数関数として扱うことで、姿勢を考慮する動作生成法を提案する。更に、その動作を構成する中間姿勢を動力学シミュレータ上でロボットへの外乱を考慮して最適化することにより、シミュレータと実機実験の差分を吸収し、少ない時間と労力で、動的起き上がり動作を脚収納外装枠を備えた等身大二脚ロボットで実現する。

2. 姿勢を考慮した多変数関数関節軌道

多くの従来研究では、関節角軌道 q を時間の関数 $q = f(t)$ として、動作から軌道の始点と終点を決めて、それを曲線で補間し、更にその曲線の制御点を人間の手動調整³⁾、²⁾ または最適化すること⁴⁾、⁵⁾ により、動作軌道生成している。この手法は実機実験では、決まった動作しかできず、モデル誤差や外乱により、姿勢状態が変化した場合を考慮できない。本研究の特徴は、ロボットの関節軌道 q を時間 t とロボットの姿勢 (r, p, y) の多変数関数 $f(t, r, p, y)$ として扱う。そのため、軌道を結ぶ曲線も多次元曲線になる、本研究では軌道を結ぶ曲線で3次スプライン曲線を使用する。その理由は、多項式係数求めるには、対称三重対角行列を扱う問題に帰着するので、少ない計算量で実装できるからである。本節では、2.1項で多次元スプライン曲線を説明し、2.2項で軌道関数の次元削減と姿勢状態の選択について論じる。

2.1 多次元スプライン曲線

2.1.1 一次元3次スプライン曲線

閉区間 $[a, b]$ を等間隔で区切られた $n + 1$ 点の集合 $S = \{x_0, x_1, \dots, x_n\}$ とする。

$$x_i = a + i\Delta t, \quad \Delta t = \frac{b-a}{n}, \quad i = 0, 1, \dots, n \quad (1)$$

閉区間において、任意の制御点 $y_0, y_1, \dots, y_n \in \mathbb{R}$ を通る一次元 3 次スプライン曲線 $s(x)$ を以下のように定める。

$$s(x) = \sum_{i=1}^{n+3} c_i u_i(x) \quad (2)$$

ただし、基底関数 $u_i(x)$ は次のように与えられる。

$$u_i(x) = \phi\left(\frac{x-a}{\Delta t} - (i-2)\right), \quad i = 1, 2, \dots, n+3 \quad (3)$$

with

$$\phi(t) = \begin{cases} (2-|t|)^3 & 1 < |t| \leq 2 \\ 4-6|t|^2+3|t|^3 & |t| \leq 1 \\ 0 & \text{elsewhere} \end{cases} \quad (4)$$

また、係数 c_i は与えられた $n+1$ 制御点 y_i の拘束条件と 2 次微分の境界条件 $\ddot{s}(a) = \alpha$, $\ddot{s}(b) = \beta$ から計算される。

2.1.2 多次元 3 次スプライン曲線への一般化

閉区間 $[a_i, b_i]$ を等間隔で区切られた $n_i + 1$ 点の集合 $S_i = x_0, x_1, \dots, x_{n_i}$ を d 個とそれに応じる $\prod_{i=1}^d (n_i + 1)$ 制御点 $y_{i_1 i_2 \dots i_d} \in \mathbb{R}$ を与えられた時、多次元 3 次スプライン曲線 $s(x, z)$ は以下のように定められる。

$$s(\mathbf{x}) = \sum_{i_1=1}^{n_1+3} \dots \sum_{i_d=1}^{n_d+3} c_{i_1 i_2 \dots i_d} \prod_{j=1}^d u_{i_j}^j(x^j) \quad \mathbf{x} = (x^1, \dots, x^d) \quad (5)$$

基底関数 $u_{i_j}^j(x^j) = \phi\left(\frac{x^j - a_j}{\Delta t_j} + 2 - i_j\right)$, 係数 $c_{i_1 i_2 \dots i_d}$ は以下の Algorithm 1 で再帰的に計算できる。⁶⁾

2.2 姿勢補間関数

ロボットの関節軌道は多次元関数で表せるが、2.1.2 項に示したように、軌道の次元数を増やすほど、求められる軌道の制御点の数は指数関数的に増加する。これは人間の手動設計でも、最適化探索にも不利になると考えられる。そのため、まずロボットの関節軌道を式 (6) のように、時間に対する補間曲線と姿勢に対する補間曲線の平均値で表せる。

$$\mathbf{q} = \mathbf{f}(t, \mathbf{r}, \mathbf{p}, \mathbf{y}) = \frac{\mathbf{g}(t) + \mathbf{h}(\mathbf{r}, \mathbf{p}, \mathbf{y})}{2} \quad (6)$$

更に、以下の理由で姿勢に対する補間曲線の次元数も削減できる。剛体の姿勢の運動自由度は 3 つである。しかし、本研究では起き上がり問題に限り、ロボットの姿勢状態、z

Algorithm 1 Calculate coefficients $c_{i_1 i_2 \dots i_d}$

```

1: for  $q = 0 \dots n_d$  do
2:   Compute coefficients  $c_{i_1 i_2 \dots i_{(d-1)} q}^*$  of the  $d-1$ 
   dimensions splines  $s_{q,1}(x^1, \dots, x^{d-1})$  with the inter-
   polation data  $y_{i_1 i_2 \dots i_{d-1} q}$ 
3: end for
4: for  $i_1 = 1 \dots n_1 + 3$  do
5:    $\vdots$ 
6:   for  $i_{d-1} = 1 \dots n_{d-1} + 3$  do
7:     Compute coefficients  $c_{i_1 i_2 \dots i_d}$  of the splines
     with the interpolation data  $c_{i_1 i_2 \dots i_{d-1} q}^*$  and nodes  $x_q$ 
     ( $q = 0 \dots n_d$ )
8:   end for
9:    $\vdots$ 
10: end for

```

軸周りの回転を無視すればよい。なぜならば、Fig.1 に示したように起き上がり出来た直後の姿勢が、床滑りにより z 軸周り回転されも、目的姿勢を達成したと見なすべきである。

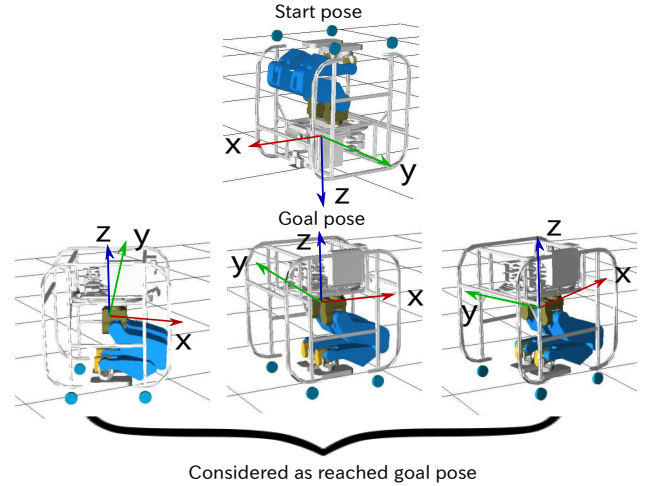


Fig. 1: Self-righting start pose and goal pose. Note that robot's pose here includes joint angle and posture

つまり、ロボットの姿勢状態をロボット座標系の z 軸の単位ベクトルで表せる。単位ベクトルは 3 変数と 1 拘束条件を用いて、実質自由度は 2 である。似たようなアイデアで、重力ベクトルを扱える先行研究もある⁷⁾。さらに、例えば各次元を 5 分割すると、3 次元の姿勢状態を扱う場合、 $5^3 = 125$ の制御点が必要であるのに対して、2 次元の姿勢状態を扱う場合、 $5^2 = 25$ の制御点だけ必要であり、これはロボットの関節数分 (本研究では $N = 12$ の自由度を持つ脚型ロボットを利用している) を考える

と、2次元の姿勢状態を扱うことにより、軌道の制御点を $(125 - 25) \times 12 = 1200$ 個の次元を削減できた。以後、 z 軸周りの回転を考慮しない姿勢状態の2次元を表現する方法について論じる。

ここでは、上記の2つの自由度を表現する4つの方法を検討する。

- (i) ロボットの回転角度 roll と pitch.
- (ii) ロボット座標 z 単位ベクトル ez' からワールド座標 z 単位ベクトル ez までの最小 roll-pitch 回転角度. Fig.2 に示したように、 $v = (v_x, v_y, v_z) = ez \times ez'$ は ez と ez' の外積で、 α は ez と ez' 間の角度で、その場合 $v_z = 0$ であり、 z 軸周りを考慮しないロボット姿勢状態の2自由度は αv_x と αv_y に表せる。
- (iii) ロボット座標 z 単位ベクトルを球面座標系で表した角度 (θ, ϕ)
- (iv) ロボットの回転行列対数関数の最初の2つの要素

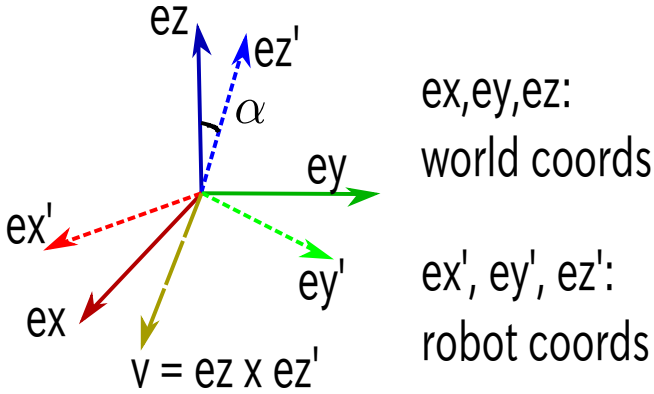


Fig. 2: Method for calculating minimum roll-pitch rotation from robot coordinate's z -axis unit vector ez' to world coordinate's z -axis unit vector ez

Fig.3 には逆さま状態から 180 度後転する実験データから得られた姿勢を上記の4つの表現で表した際の結果である。

どの表現でも不連続点が現れた。これにより関節角軌道が不連続になり、その点では関節トルクが過大になって、実機には同じ軌道の再現が困難である。姿勢軌道の不連続点は2つの原因に分けられる。一つ目は姿勢表現の特異点による不連続点である。(i)に関しては、pitch が $\frac{\pi}{2}$ の時、roll が定まらないため、その付近で不連続になった。(ii)に関しては、逆さまな状態で ez と ez' が並行になるので、外積ベクトル v が一意的に定まらないため、その付近で不連続になった。(iii)に関しては、逆さまな状態で角度 ϕ が一意的に定まらないため、その付近で不連続になった。

二つ目は角度を $(-\pi, \pi]$ の範囲で表すため、 $-\pi$ と π 付近には、値が不連続になる。これは姿勢軌道関数が式 (7)

のように 2π 周期の周期関数にすれば、関節角軌道が連続につなげることができる。

$$h(x, y) = h(x + 2\pi, y) = h(x, y + 2\pi) \quad (7)$$

これにより、(iv) により得られた関節角軌道が連続になる。

上記の検討を踏まえて、本研究では z 軸周りを考慮しないロボット姿勢状態の2自由度として、(iv) ロボットの回転行列対数関数の最初の2つの要素を選択する。

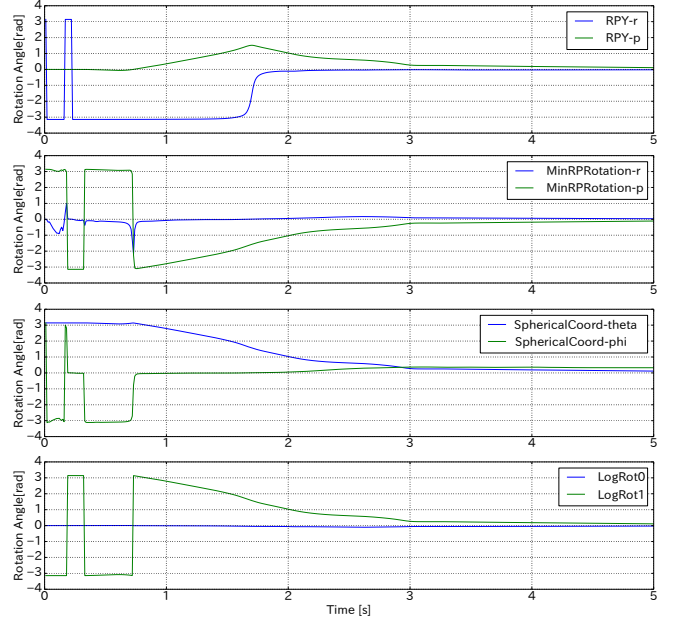


Fig. 3: Result of 4 types 2-dim robot posture with data from experiment

以上をまとめて、本研究での関節角軌道 q は以下のよう

$$q = f(t, x, y) = \frac{g(t) + h(x, y)}{2} \quad (8)$$

その中に、 $g(t) = \sum_{i=1}^{n+3} c_i \otimes u_i(t)$ は時間に対する補間曲線。これは $n + 1$ 点の制御点 $y_i, (i = 0, \dots, n)$ により定められる。 $h(x, y) = \sum_{i=1}^{n_1+3} \sum_{j=1}^{n_2+3} c_{i_1 i_2} \otimes u_{i_1}(x) \otimes u_{i_2}(y)$ は2次元の姿勢 (x, y) に対する補間曲面。これは $(n_1 + 1) \times (n_2 + 1)$ 点の制御点 $y_{i_1 i_2}$ により定められる。(\otimes 記号の意味: $x = [x_1, \dots, x_n], y = [y_1, \dots, y_n]$ の時、 $x \otimes y = [x_1 y_1, \dots, x_n y_n]$ である)

3. 動力学シミュレータベース動作探索法

(8) の関節角軌道を元に、軌道の制御点が動力学シミュレーター内で進化計算により最適化される。各試行に対して、達成すべき姿勢との差分、軌道を実行中の関節角度、関節角速度、トルクの制限と自己干渉により評価値を計算する⁵⁾。

ただし、この最適化で得られた軌道をそのまま実機で実行しても、モデル化誤差、サーボ剛性、外装の弾性変

形, 接触状態の相違などのために, 成功しない. そのため, シミュレーター内に動作実行する際の外乱として, 0.5 秒ごとに, 0.1 秒間に外力を加える. 耐えられる外乱を最大化するために, 評価関数の計算法は以下の Algorithm 2 示すようにに変更した. なお, 上記の⁵⁾による各試行に対する評価値計算は Algorithm 2 の中に, 行 7 の *iteration_evaluate_value()* 関数に相当する. 本手法で用いる評価関数は全試行の平均値 \bar{J}_{it} に耐えられる外乱を最大化するための $J_{disturb}$ を加えている.

Algorithm 2 Evaluation Function for Evolutionary Optimization

Require: $dF, \omega_{it}, \omega_{disturb}, \theta$

Global Variables:

dF : Disturbance increment in each trial

ω_{it} : Weight for minimize self-righting evaluating point calculated as⁵⁾

$\omega_{disturb}$: Weight for maximize accept disturbance

Input:

θ : Parameters for joint angle trajectory

```

1: function eval_with_maximize_accept_disturbance( $\theta$ )
2:    $F \leftarrow 0, \text{ iter} \leftarrow 1$ 
3:    $S_{jit} \leftarrow 0, \text{ prev}J_{it} \leftarrow +\infty$ 
4:    $SuccFlag \leftarrow True$ 
5:   while  $SuccFlag = True$  do
6:     simulate( $\theta, F$ )  $\triangleright$  Simulate with parameters  $\theta$ 
       and disturbance  $F(N)$ 
7:      $J_{it} \leftarrow \text{iteration\_evaluate\_value}()$   $\triangleright$ 
       Calculate evaluate value as5)
8:      $S_{jit} \leftarrow S_{jit} + \min(J_{it}, \text{prev}J_{it})$ 
9:     if motion_success() then  $\triangleright$  Reach goal
       posture and no self-collision occurred
10:       $F \leftarrow F + dF$ 
11:       $iter \leftarrow iter + 1$ 
12:     else
13:        $SuccFlag = False$ 
14:     end if
15:   end while
16:    $\bar{J}_{it} \leftarrow \frac{S_{jit}}{it}, J_{disturb} \leftarrow \frac{dF}{F}$ 
17:    $J \leftarrow \omega_{it}\bar{J}_{it} + \omega_{disturb}J_{disturb}$ 
18:   return  $J$ 
19: end function

```

探索変数とその次元数を Table 1 にまとめる. ここでは, N はロボットの自由度数である. 全自由度の合計は $1 + 4N + 14N = 18N + 1$ 次元であり, $N = 12$ の自由度を持つ脚型ロボットでは 217 次元が必要である.

Fig.4 には, 独立に 7 回の探索を行い, その 7 回の評価

値曲線を示した. 7 回とも約 10 分後, 成功動作 (評価値が 1000 より小さい) が得られて, さらに評価関数が 1 時間を経過すると, ほぼ変動しない状態になったと確認できた.

Table 1: Search variables and their Degrees of Freedom

	Search Variable	DoF
t_m	Interpolation duration of joint angle trajectory	1
y_i	Control points for Time Interpolation	$4 \times N$
$y_{i_1 i_2}$	Control points for Posture Interpolation	$14 \times N$
θ	All parameters	$18N + 1$

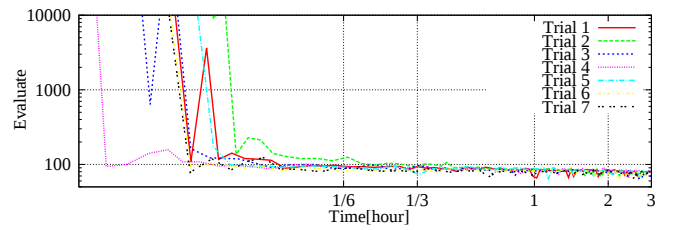


Fig. 4: Evaluate value of evolutionary search in 7 trial (log scale)

4. 動作実験

本節では, 姿勢状態を考慮した動作生成の提案手法が有効であることを確認するために行った実機実験について述べる.

4.1 実験条件

実験では, 脚収納外装枠を備えた等身大二脚ロボット CHIDORI-F (身長 1.3[m], 体重 67[kg]) を用いた. CHIDORI-F には, 各足首に六軸力覚センサが搭載され, 胴体にジャイロセンサと加速度センサも搭載されて, 姿勢推定するために使われている. 推定した姿勢を姿勢を考慮した多変数関数関節軌道に使用されている. 胴体の周りには外装を付いており, Fig.5 に示すように膝を曲げることで転倒時にロボットと環境が接触せず, ロボットの破損を防ぐことができ, 実験の開始から終了までクレーンなしで運用することが可能である. また, フレーム底にはねじりバネつきキャスタがついており, 転倒時の衝撃緩和と, 平地での車輪を活用した移動が可能となる.

4.2 脚収納外装枠を備えた等身大二脚ロボットの起き上がり動作実験

Fig.6 および Fig.7 に, 起き上がり動作実験の様子を示す. この実験では, ロボットが逆さまであるのが開始で,

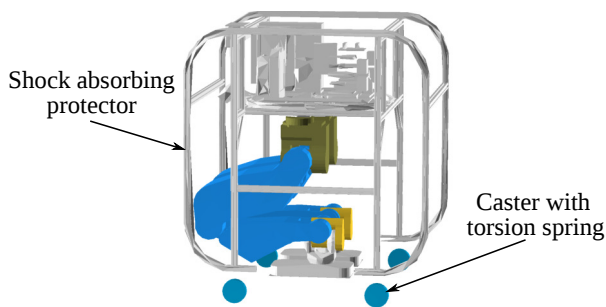


Fig. 5: CHIDORI-F robot in knee-bending pose

そこから 180 度回転して、足が地面に向くのが目標姿勢である。この 2 つの状態の姿勢と関節角度が入力で、その 2 点をつなげる軌道のパラメータ（制御点と軌道時間）を動力学シミュレータで進化計算によって最適化された。最適化により得られたパラメータを実機ロボットで動作実行した様子が Fig.6 と Fig.7 一行目に示す。二行目は動力学シミュレータの結果である。三行目は実験データからロボットの関節角軌道とロボットの姿勢を可視化した。赤色で示したロボットは実機実験のデータで、青色で示したロボットはシミュレーションのデータである。緑色で示したロボットは実機の姿勢に応じて関節角軌道が修正されることを示すため、姿勢は実機実験のデータを用いて、関節軌道はシミュレーションのデータを用いた。ただし、Fig.6 では姿勢を考慮した多変数関数関節軌道を用いないため、赤色で示したロボットと緑色で示したロボットの関節角は完全一致する。

Fig.6 では、動作軌道が姿勢を考慮しないため、実機実験ではシミュレータ上より早く回転しても、動作の軌道修正がないため、最後に回転の勢いが足りず、90 度しか回転できなかった。Fig.7 では、動作軌道が姿勢を考慮したため、実機実験ではシミュレータ上より早く回転した場合、関節角度が修正されて、目標姿勢まで回転できた。特に、シミュレータより早めに目標姿勢に達成した際（ $t = 2.4[s]$ ）、実機実験ロボットが早めに膝しやがむ姿勢に遷移したことを観測できた。

5. 結論

本研究では、動的起き上がり動作を実現することを目指し、ロボットの動作軌道を時間とロボットの姿勢の多変数関数で表現するモデルを提案した。また、動作軌道を構成する中間姿勢を動力学シミュレータ上で外乱を与えて最適化することで、シミュレータと実機実験の差分を吸収し、少ない時間と労力で、動的起き上がり動作を脚収納外装枠を備えた等身大二脚ロボットで実現した。これにより、

ヒューノイドの在るべき姿を「決して倒れないもの」から「倒れても自力で起き上がりできて、活動が続けられるもの」へと変えられた。これによってヒューノイドの転倒防止技術への要求は緩和しながら、システムとしての全体の信頼性を向上させることが可能となる。

今後の展望として、ヒューノイドの腕を活用した起き上がり動作に取り組み、さらに斜面や凸凹の地面でも起き上がることを目指していく。また、転倒判定機能を備えた歩行動作生成法⁸⁾と統合することにより、屋外での自律移動を実現する。

参考文献

- [1] 金広文男, 金子健二, 藤原清司, 原田研介, 梶田秀司, 横井一仁, 比留川博久, 赤地一彦, 五十棲隆勝. ヒューノイドの転倒回復機能の実現. 日本ロボット学会誌, Vol. 22, No. 1, pp. 37–45, 2004.
- [2] Yohei Kakiuchi, Masayuki Kamon, Nobuyasu Shimomura, Sou Yukizaki, Noriaki Takasugi, Shunichi Nozawa, Kei Okada, and Masayuki Inaba. Development of life-sized humanoid robot platform with robustness for falling down, long time working and error occurrence. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 689–696. IEEE, 2017.
- [3] 國吉康夫, 大村吉幸, 寺田耕志, 長久保晶彦. 等身大ヒューノイドロボットによるダイナミック起き上がり行動の実現. 日本ロボット学会誌, Vol. 23, No. 6, pp. 706–717, 2005.
- [4] Ryo Terasawa, Shintaro Noda, Kunio Kojima, Ryo Koyama, Fumihito Sugai, Shunichi Nozawa, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Achievement of dynamic tennis swing motion by offline motion planning and online trajectory modification based on optimization with a humanoid robot. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 1094–1100. IEEE, 2016.
- [5] Shintaro Noda, Yohei Kakiuchi, Hiroki Takeda, Kei Okada, and Masayuki Inaba. Goal-oriented simulation-based motion interpolator for complex contact transition: Experiments on knee-contact behavior. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pp. 761–767. IEEE, 2018.
- [6] Christian Habermann and Fabian Kindermann. Multidimensional spline interpolation: Theory and applications. *Computational Economics*, Vol. 30, No. 2, pp. 153–169, 2007.
- [7] Joonho Lee, Jemin Hwangbo, and Marco Hutter. Robust recovery controller for a quadrupedal robot using deep reinforcement learning. *CoRR*, Vol. abs/1901.07517, , 2019.
- [8] Yuta Kojio, Yasuhiro Ishiguro, Kim-Ngoc-Khanh Nguyen, Fumihito Sugai, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Unified balance control for biped robots including modification of footsteps with angular momentum and falling detection based on capturability. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, Accepted.

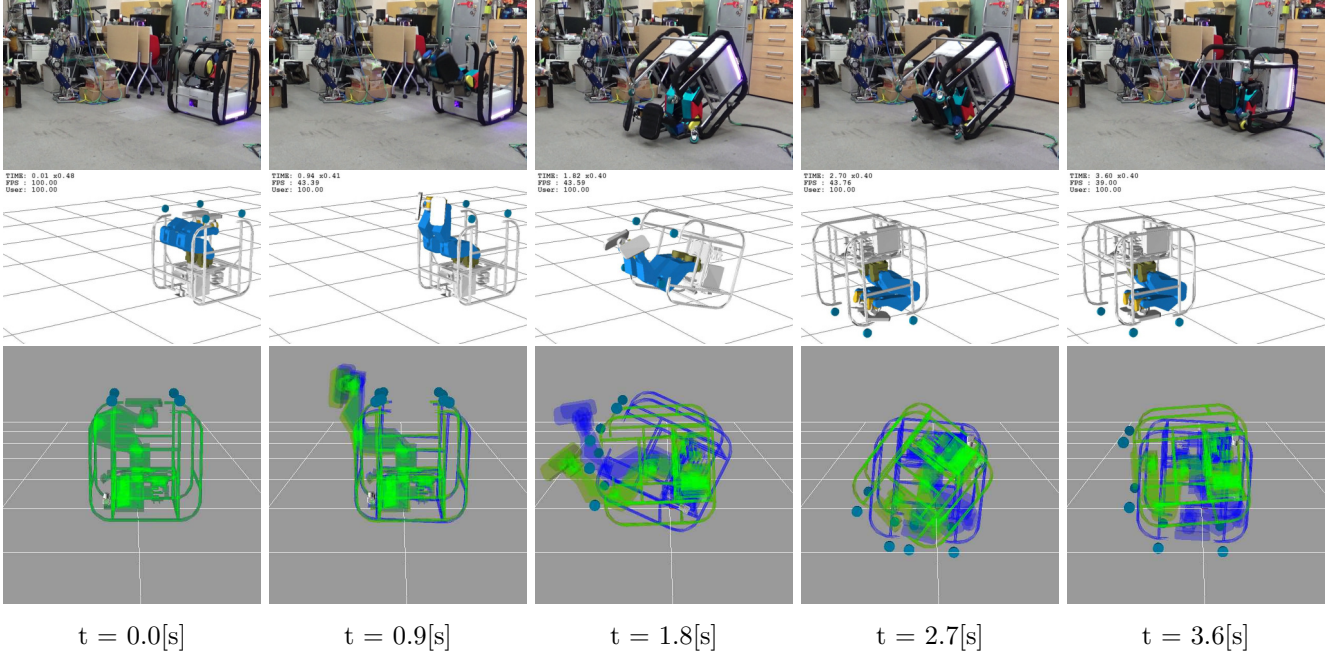


Fig. 6: CHIDORI-F's self-righting motion without posture feedback in real robot experiment (first line) and simulator (second line) and log visual analyzation(third line)

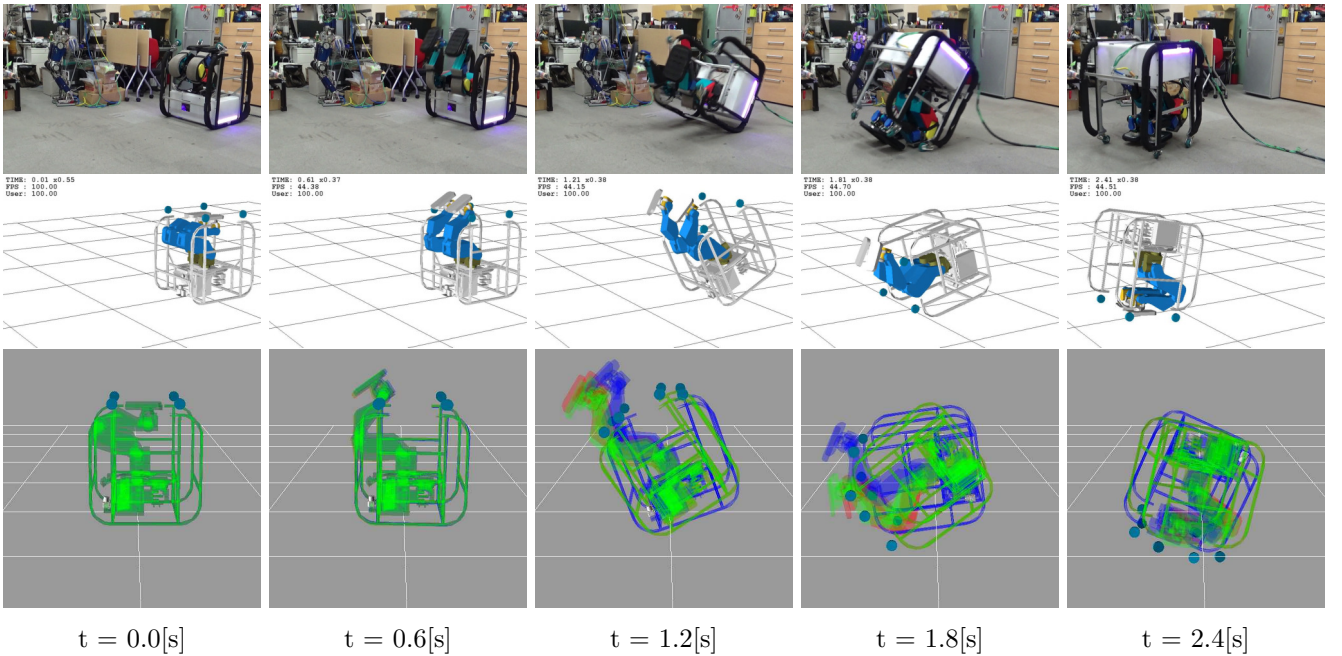


Fig. 7: CHIDORI-F's self-righting motion with posture feedback in real robot experiment (first line) and simulator (second line) and log visual analyzation(third line)