

拘束を伴うマルチボディシステムのための 数値・数式ハイブリッドモデリング

Symbolic-Numeric Hybrid Modeling of MBS With Constraints

学 高橋 準 (都市大)
正 金宮 好和 (都市大)

学 三橋 勝 (都市大)
正 佐藤 大祐 (都市大)

Jun TAKAHASHI, Tokyo City University

Masaru MITSUHASHI, Tokyo City University, mitsuhashi@rls.mse.tcu.ac.jp

Yoshikazu KANAMIYA, Tokyo City University

Daisuke SATO, Tokyo City University

The Assembly-Disassembly Algorithm (ADA) is known for its computational efficiency. We propose a way of extending ADA to further improve the efficiency. This is achieved via symbolic-numeric hybrid modeling. We prepare the symbolic equations of motion of specific subchains using the Lagrange formulation. The equations are then integrated into the numerical computation procedures of ADA. Finally, we compare the computation time of our hybrid modeling approach with that of ADA.

Key Words: Multibody Systems, Forward Dynamics, Symbolic-Numeric Hybrid Computation

1 緒言

近年, ロボット研究において, 人型ロボットの走行に代表される高速運動が要求されるようになってきている. このような動作の解析や制御系の設計など, 効率的な研究には動力学シミュレータが必要不可欠である. 動力学シミュレータにおいて, 順動力学計算はシミュレータの性能を大きく左右する部分であり, 大自由度系やリンクの構造変化に対応するために, 高速化やモデリングの汎用性が課題となっている.

そこで我々は, 閉リンク機構を含む様々なリンク系のシミュレーションを効率的に行うために, 高速な順動力学エンジンを開発することを目的とする. これまでに, 山根らによって1個のCPUで $O(n)$, n 個のCPUによる並列計算で $O(\log n)$ の計算複雑性を持つ順動力学計算法, Assembly-Disassembly Algorithm (ADA) [1][2]が開発されており, 同様の計算複雑性を持つアルゴリズムでは並列計算を用いた場合に最速であることが示されている [3]. また, 数値計算と数式処理を組み合わせ, 計算精度を維持しつつ計算効率を高めることを目指して, 数値・数式ハイブリッド手法の研究が進められている [4][5]. 本稿ではADAとLagrange法を組み合わせる数値・数式ハイブリッド手法により, 順動力学計算の効率化を行い, その有用性を検証する.

2 Assembly-Disassembly Algorithm (ADA)

我々は, 空間ベクトル [6] を用いて数式を表現する. 空間ベクトルは, 角速度 $\omega \in \mathbb{R}^3$ と並進の速度 $v \in \mathbb{R}^3$ やモーメント $n \in \mathbb{R}^3$ と並進の力 $f \in \mathbb{R}^3$ など, 回転成分と並進成分を一つに組み合わせた六次元のベクトルである. 速度と力はそれぞれ $\mathcal{V} = [\omega^T, v^T]^T$ と $\mathcal{F} = [n^T, f^T]^T$ となる. また, ADAでは, 複数のリンクを一つの剛体として表す Articulated Body (AB) の運動方程式を用いて表現される.

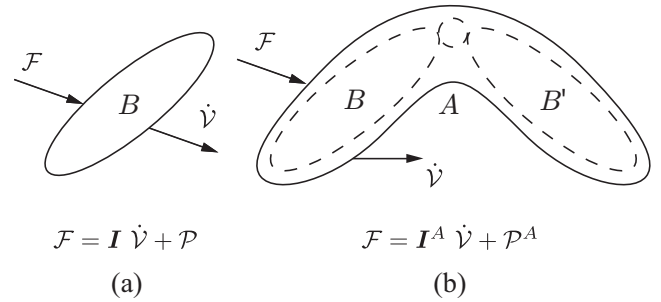


Fig. 1: A rigid body and an articulated body.

2.1 Articulated Body の運動方程式

一つの剛体 B を図 1 (a) に示す. 加えられた力 $\mathcal{F} \in \mathbb{R}^6$ と発生する加速度 $\dot{\mathcal{V}} \in \mathbb{R}^6$ の関係は, 剛体の運動方程式から次式で表される.

$$\mathcal{F} = \mathbf{I} \dot{\mathcal{V}} + \mathcal{P} \quad (1)$$

ここで, 式中の $\mathbf{I} \in \mathbb{R}^{6 \times 6}$ は B の慣性行列, $\mathcal{P} \in \mathbb{R}^6$ は B のコリオリ力や外力などの加速度に依存しない力 (バイアス力) である. 次に, 図 1 (b) に示すリンク系について考える. 二つ目の剛体 B' が関節によって B と繋がっている. これを一つの AB A と考えると, \mathcal{F} と $\dot{\mathcal{V}}$ の関係は, 次のように表される.

$$\mathcal{F} = \mathbf{I}^A \dot{\mathcal{V}} + \mathcal{P}^A \quad (2)$$

式中の $\mathbf{I}^A \in \mathbb{R}^{6 \times 6}$ は Articulated-Body Inertia (ABI), $\mathcal{P}^A \in \mathbb{R}^6$ は A のバイアス力である. \mathbf{I}^A は正定値対称であり, 上式は次式と等価である.

$$\dot{\mathcal{V}} = \Phi^A \mathcal{F} + \mathcal{B}^A \quad (3)$$

ただし, $\Phi^A = (\mathbf{I}^A)^{-1}$, $\mathcal{B}^A = -\Phi^A \mathcal{P}^A$ と置いた. 式中の $\Phi^A \in \mathbb{R}^{6 \times 6}$ は Inverse Articulated-Body Inertia (IABI),

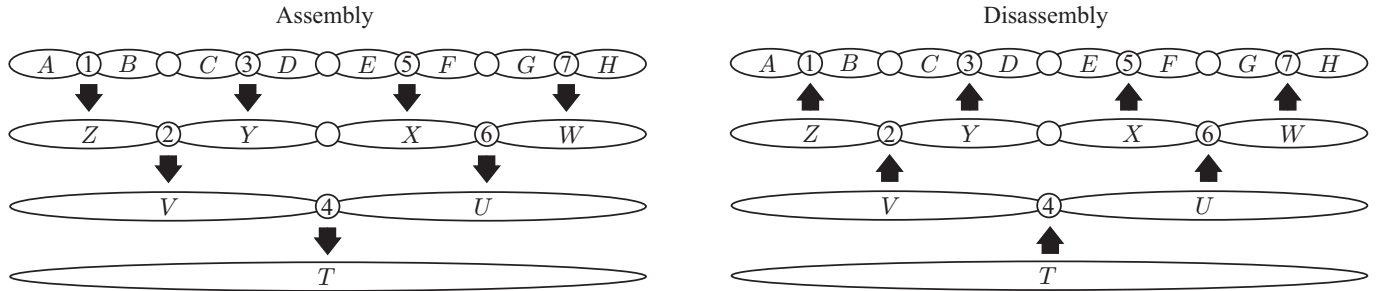


Fig. 2: Illustration of the assembly and disassembly phases based on the ADA.

$B^A \in \mathbb{R}^6$ は A のバイアス加速度である．上式の表現を用いることで，一つの AB に対して複数の力を設定できるようになる． AB に加えられた力 \mathcal{F}_i と加速度 \dot{v}_i の関係は次のようになる ($i = 1 \sim h$)．

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \vdots \\ \dot{v}_h \end{bmatrix} = \begin{bmatrix} \Phi_{11}^A & \Phi_{12}^A & \dots & \Phi_{1h}^A \\ \Phi_{21}^A & \Phi_{22}^A & \dots & \Phi_{2h}^A \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{h1}^A & \Phi_{h2}^A & \dots & \Phi_{hh}^A \end{bmatrix} \begin{bmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \\ \vdots \\ \mathcal{F}_h \end{bmatrix} + \begin{bmatrix} B_1^A \\ B_2^A \\ \vdots \\ B_h^A \end{bmatrix} \quad (4)$$

ここで， Φ_{ij}^A は j に加えられた力による i の加速度の関係を表す干渉項である．ADA では (4) を基にリンク系の運動方程式が導出される．

2.2 アルゴリズムの概要

ADA は assembly ステップと disassembly ステップの二つの計算ステップからなる．各計算ステップについて，図 2 のシリアル 8 リンク機構を例に説明する．

Assembly ステップ

1. 各リンクの IABI とバイアス加速度を求める．
2. 隣り合うリンクを接続し，新たな AB の IABI とバイアス加速度を求める．
3. AB の接続を繰り返し，全体の AB を作成する．

Disassembly ステップ

1. 最後に接続した関節 4 で働く拘束力を求める．
2. 関節 4 から V, U に加わる力を求める．
3. 関節 4 の加速度を求める．
4. 関節 2, 6 で働く拘束力を求める．
5. 「2.」から「4.」の計算を assembly ステップと逆の順序で行う．

3 数値・数式ハイブリッドモデリング

2.2 節で示したアルゴリズムから， V, U の IABI とバイアス加速度を事前に計算し，分解することができれば，計算量を削減できる．しかし，ADA では一つ一つ計算していく必要があるため，このようなアプローチをとることができない．そこで，Lagrange 法を用いて， V, U の運動方程式を用意し，数値計算と組み合わせることで IABI とバイアス加速度を求める手法，ハイブリッドモデリングを提案する．ここで用いるハイブリッドという言葉には，数式処理と数値計算を組み合わせることと，Lagrange 法と ADA を組み合わせることの二つの意味がある．

3.1 アルゴリズムの概要

ハイブリッドモデリングは，ADA と同様に二つの計算ステップからなるが，ステップ内の計算に違いがある．図 3 のシリアル 8 リンク機構の計算ステップを例として ADA とのその違いを示す．

Assembly ステップ

1. 4 リンクの運動方程式を Lagrange 法より求める．
2. 「1.」で求めた運動方程式から V と U の IABI とバイアス加速度を求める．
3. 隣り合う AB を接続し，全体の AB を作成する．

Disassembly ステップ

1. 最後に接続した関節 4 で働く拘束力を求める．
2. 関節 4 から V, U に加わる力を求める．
3. 関節 4 の加速度を求める．
4. 関節 2, 6 で働く拘束力を求める．
5. 一度に四つのリンクを分解し，関節の加速度を求める．

ADA との違いは，assembly ステップの「2.」と disassembly ステップの「5.」で，四つのリンクを一度に接続・分解する部分である．なお，ADA は一度に二つのリンクしか接続・分解できないのに対し，ハイブリッドモデリングは，三つ以上のリンクを接続・分解できる．以下では，その運動方程式を導出する．

3.2 Articulated Body の接続

図 4 に示すように，四つの AB A から D を接続した AB V を考える．関節 i から A に伝わる力を \mathcal{F}_i^A ，関節 m から D に伝わる力を \mathcal{F}_m^D とすると，それぞれの加速度 \dot{v}_i^A ， \dot{v}_m^D との関係は次のように表される．

$$\begin{bmatrix} \dot{v}_i^A \\ \dot{v}_m^D \end{bmatrix} = \begin{bmatrix} \Phi_{iV}^V & \Phi_{mV}^V \\ \Phi_{mV}^D & \Phi_{mV}^D \end{bmatrix} \begin{bmatrix} \mathcal{F}_i^A \\ \mathcal{F}_m^D \end{bmatrix} + \begin{bmatrix} B_i^V \\ B_m^V \end{bmatrix} \quad (5)$$

一方，図 4 を浮遊ベースを含んだ 4 リンクマニピュレータとして考える．関節変位 $q_V = [q_j^T, q_k^T, q_l^T]^T$ を一般化座標とすると，このモデルの運動方程式は次式で表される．ただし， $q_\alpha \in \mathbb{R}^{n_\alpha}$ は関節 α の関節変位， n_α は関節の自由度とする ($\alpha = j, k, l$)．

$$\begin{bmatrix} M_A & M_{AV} \\ M_{AV}^T & M_V \end{bmatrix} \begin{bmatrix} \dot{q}_i^A \\ \dot{q}_V \end{bmatrix} + \begin{bmatrix} C_A \\ c_V \end{bmatrix} + \begin{bmatrix} G_A \\ g_V \end{bmatrix} = \begin{bmatrix} 0 \\ \tau_V \end{bmatrix} \quad (6)$$

式中の変数名は次の通りである．

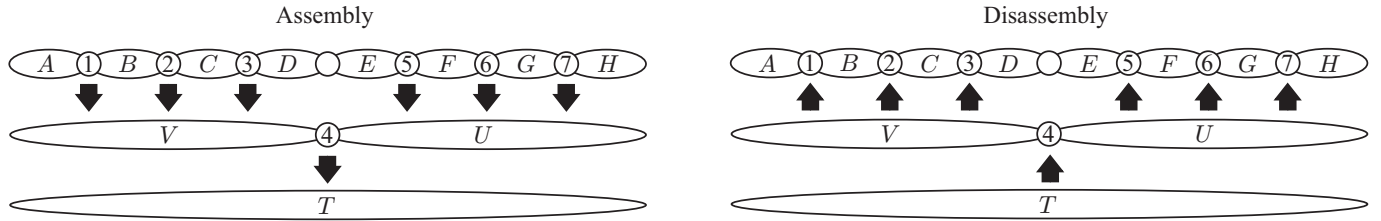


Fig. 3: Illustration of the assembly and disassembly phases based on the hybrid modeling.

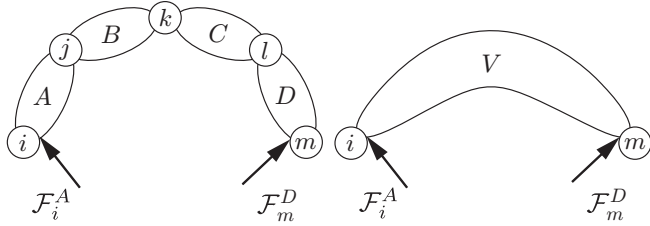


Fig. 4: Assembly of articulated bodies for hybrid modeling. V has four articulated bodies.

- $M_A \in \mathbb{R}^{6 \times 6}$: A におけるシステム全体の慣性行列
- $M_{AV} \in \mathbb{R}^{6 \times n}$: A と V の干涉慣性行列
- $M_V \in \mathbb{R}^{n \times n}$: V の慣性行列
- $C_A \in \mathbb{R}^6$: A のコリオリ遠心力項
- $c_V \in \mathbb{R}^n$: V のコリオリ遠心力項
- $G_A \in \mathbb{R}^6$: A の重力項
- $g_V \in \mathbb{R}^n$: V の重力項
- $\mathcal{V}_i^A \in \mathbb{R}^6$: 関節 i の並進速度と角速度
- $q \in \mathbb{R}^n$: V の関節変位
- $\tau_V \in \mathbb{R}^n$: V の各関節を駆動する一般化力

また、関節速度 \mathcal{V}_i^A と \mathcal{V}_m^D は次式で表される。

$$\begin{bmatrix} \mathcal{V}_i^A \\ \mathcal{V}_m^D \end{bmatrix} = \begin{bmatrix} U & 0 \\ J_{mi} & J_{mV} \end{bmatrix} \begin{bmatrix} \mathcal{V}_i^A \\ \dot{q}_V \end{bmatrix} \quad (7)$$

ここで、 $U \in \mathbb{R}^{6 \times 6}$ は単位行列、 $J_{mi} \in \mathbb{R}^{6 \times 6}$ は \mathcal{V}_i^A を \mathcal{V}_m^D に変換するヤコビ行列、 $J_{mV} \in \mathbb{R}^{6 \times n}$ は \dot{q}_V を \mathcal{V}_m^D に変換するヤコビ行列である。上式を時間微分すると次式を得る。

$$\begin{bmatrix} \dot{\mathcal{V}}_i^A \\ \dot{\mathcal{V}}_m^D \end{bmatrix} = \begin{bmatrix} U & 0 \\ J_{mi} & J_{mV} \end{bmatrix} \begin{bmatrix} \dot{\mathcal{V}}_i^A \\ \dot{\mathcal{V}}_m^D \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \dot{J}_{mi} & \dot{J}_{mV} \end{bmatrix} \begin{bmatrix} \mathcal{V}_i^A \\ \dot{q}_V \end{bmatrix} \quad (8)$$

また、上式のヤコビ行列を用いて \mathcal{F}_i^A と \mathcal{F}_m^D を考えると、(6) は次式で表せる。

$$\begin{bmatrix} M_A & M_{AV} \\ M_{AV}^T & M_V \end{bmatrix} \begin{bmatrix} \dot{\mathcal{V}}_i^A \\ \dot{\mathcal{V}}_m^D \end{bmatrix} + \begin{bmatrix} C_A \\ c_V \end{bmatrix} + \begin{bmatrix} G_A \\ g_V \end{bmatrix} = \begin{bmatrix} 0 \\ \tau_V \end{bmatrix} + \begin{bmatrix} U & J_{mi}^T \\ 0 & J_{mV}^T \end{bmatrix} \begin{bmatrix} \mathcal{F}_i^A \\ \mathcal{F}_m^D \end{bmatrix} \quad (9)$$

上式から求めた $[(\dot{\mathcal{V}}_i^A)^T, \dot{\mathcal{V}}_m^D]^T$ を (8) に代入し (5) と比較す

ると、 Φ^V と \mathcal{B}^V は次のように求まる。

$$\begin{bmatrix} \Phi_i^V & \Phi_{im}^V \\ \Phi_{mi}^V & \Phi_m^V \end{bmatrix} = \begin{bmatrix} U & 0 \\ J_{mi} & J_{mV} \end{bmatrix} \begin{bmatrix} M_A & M_{AV} \\ M_{AV}^T & M_V \end{bmatrix}^{-1} \begin{bmatrix} U & J_{mi}^T \\ 0 & J_{mV}^T \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} \mathcal{B}_i^V \\ \mathcal{B}_m^V \end{bmatrix} = \begin{bmatrix} U & 0 \\ J_{mi} & J_{mV} \end{bmatrix} \begin{bmatrix} M_A & M_{AV} \\ M_{AV}^T & M_V \end{bmatrix}^{-1} \left(\begin{bmatrix} 0 \\ \tau_V \end{bmatrix} - \begin{bmatrix} C_A \\ c_V \end{bmatrix} - \begin{bmatrix} G_A \\ g_V \end{bmatrix} \right) + \begin{bmatrix} 0 & 0 \\ J_{mi} & J_{mV} \end{bmatrix} \begin{bmatrix} \mathcal{V}_i^A \\ \dot{q}_V \end{bmatrix} \quad (11)$$

上式より (6) の各行列成分とヤコビ行列から Φ^V , \mathcal{B}^V を求めることができる。つまり assembly ステップにおいて、一度に三つ以上のリンクを接続できる。

3.3 Articulated Body の分解

次に disassembly ステップを考える。ADA の計算により \mathcal{F}_m^D が求まったものとする。(9) の 2 行目より、 \dot{q}_V が次式から求まる。

$$\dot{q}_V = M_V^{-1} \left(\tau_V + J_{mV}^T \mathcal{F}_m^D - M_{iV}^T \dot{\mathcal{V}}_i^A - c_V - g_V \right) \quad (12)$$

つまり disassembly ステップにおいて、一度に三つ以上のリンクを分解できる。

3.4 固定ベースに接続された二次元モデル

ハイブリッドモデリングの有用性を検証するために、Lagrange 法、ADA、ハイブリッドモデリングの三つの手法それぞれを用いた計算時間を比較する。比較するモデルは固定ベースに接続された二次元モデルであり、1 リンクから 7 リンクのマニピュレータである。ただし、ハイブリッドモデリングでは 7 リンクマニピュレータに対してのみ、シミュレーションを行う。この際、根元の 4 リンクと手先の 3 リンクの運動方程式を事前に用意し、計算する。シミュレーション時間は 10 s、刻み幅は 1 ms とし、3.16 GHz Intel CPU、メモリ 4 GB の計算機を用いる。また、初期姿勢は第 1 関節を 90 deg、他の関節を 0 deg とし、初速および制御入力なしの条件で、自由落下するシミュレーションを行う。

計算時間を図 5 に示す。Lagrange 法は計算量が $O(n^3)$ 、ADA は計算量が $O(n)$ となるアルゴリズムであり、これらはグラフからも確認できた。また、6 リンクまでは Lagrange 法の方が速いが、7 リンクでは、ADA の方が速くなるため、大自由度系を扱う際には、ADA の方が効率的である。

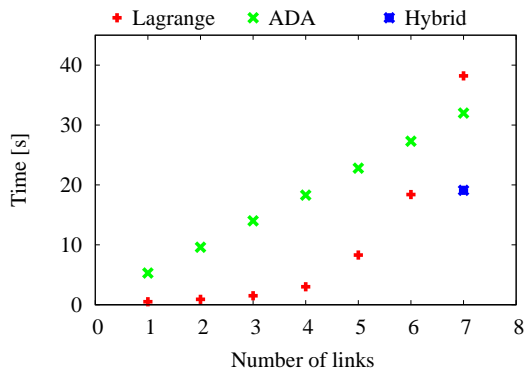


Fig. 5: Computation time of planar manipulators.

さらに、ハイブリッドモデリングを行うことで、Lagrange 法や ADA よりも 7 リンクの計算時間が短くなることも確認できた。Lagrange 法と比較して約 50 %、ADA と比較して約 60 % となった。

3.5 浮遊ベースを含んだ三次元モデル

前節で扱ったモデルは固定ベースに接続された二次元モデルである。人型ロボットをはじめとする非固定のリンク系への適応を考えると、浮遊ベースを含んだ三次元モデルでの比較を行う必要がある。また、ハイブリッドモデリングに関しては、7 リンクの計算時間のみを比較した。そのため、モデルの自由度の拡大に伴うハイブリッドモデリングの計算時間の変化も確認する必要がある。そこで、浮遊ベースを含んだ三次元モデルで ADA とハイブリッドモデリングの計算時間を比較する。

比較するシミュレーションモデルを図 6 に示す。このモデルは、浮遊ベースを含む三次元モデルであり、(a) から (e) はそれぞれ 1 リンク、3 リンク、6 リンク、9 リンク、12 リンクのマニピュレータである。ハイブリッドモデリングでは図 7 のように、3 リンクごとの数式をあらかじめ用意し、計算する。シミュレーション条件は、前のシミュレーションと同様である。ただし、制御入力を与えず、ベース部に x , y , z 軸回りの初速 1.0 rad/s を与える。

計算時間を図 8 に示す。二次元のモデルと同様に、ADA は計算量が $O(n)$ となることが確認できた。また、ハイブリッドモデリングも $O(n)$ の傾向が確認できた。さらに、ハイブリッドモデリングを用いることで、ADA よりも計算時間が短くなった。浮遊ベースを含む三次元 12 リンクマニピュレータの計算時間を比較すると、ハイブリッドモデリングは、ADA の約 66 % となった。

4 結言

閉リンク機構を含むリンク系のシミュレーションを効率的に行うために、高速な順動力学エンジンを開発することを目的として、数値・数式ハイブリッド手法による順動力学計算 (ADA) の高速化を行い、有用性を示した。

今後の課題として、用意する数式の自由度による計算時間の違いを検証し、あらかじめ用意しておく数式の自由度を決定する必要がある。また、既存のロボットの機構を把握し、それに合わせて用意する数式の機構を決め、一般的なロボッ

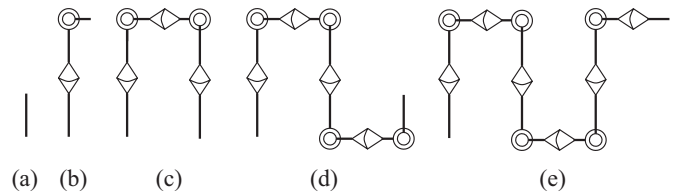


Fig. 6: 3D manipulators with free base.

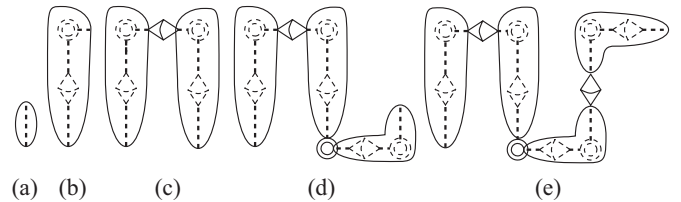


Fig. 7: Illustration of the hybrid modeling. We prepare the equation of motion of a three link manipulator except (a). (a) is a single rigid body.

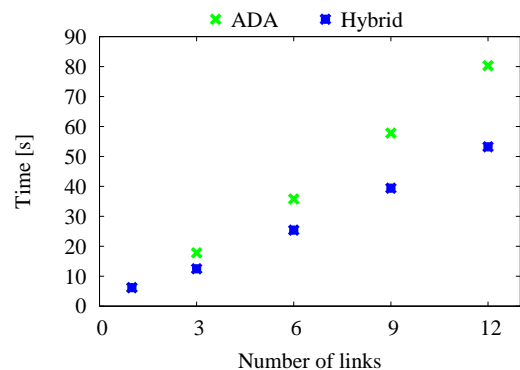


Fig. 8: Computation time of 3D manipulators.

トの大半に本手法を対応する必要もある。

文 献

- [1] K. Yamane and Y. Nakamura, "Efficient parallel dynamics computation of human figures," in *Proc. of the 2002 IEEE Int. Conf. on Robotics and Automation*, Washington, 2002, pp. 530–537.
- [2] K. Yamane and Y. Nakamura, "Parallel $O(\log N)$ algorithm for dynamics simulation of humanoid robots," in *Proc. of the 6th IEEE-RAS Int. Conf. on Humanoid Robots*, Genova, Italy, 2006, pp. 554–559.
- [3] K. Yamane and Y. Nakamura, "Comparative study on serial and parallel forward dynamics algorithms for kinematic chains," *The Int. J. of Robotics Research*, vol. 28, no. 5, pp. 622–629, May 2009.
- [4] H. Iwane, H. Yanami, H. Anai and K. Yokoyama, "An effective implementation of a symbolic-numeric cylindrical algebraic decomposition for quantifier elimination," in *Proc. of the 2009 Conf. on Symbolic Numeric Computation*, Kyoto, Japan, 2009, pp. 55–64.
- [5] 穴井宏和, 菅野政明, "数値・数式ハイブリッド手法に基づく高品質計算とその展開", 計測と制御, vol. 5, 2010, pp. 309–314.
- [6] R. Featherstone, *Rigid Body Dynamics Algorithms*, Canberra, AU: Springer, pp. 7–38, 2007.