

ネットワーク分散計算による動力学シミュレーションの高速化

High-Speed Dynamics Computation via Network Distributed Computation

○ 山根 克 (東大)

正 中村 仁彦 (東大)

Katsu YAMANE

Yoshihiko NAKAMURA

University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656 Japan

Abstract— This paper describes the parallel computation scheme for dynamics simulation and motion generation for multiple human figures interacting with each others and the environment. Our previously proposed algorithms for computing the dynamics of structure-varying kinematic chains are parallelized in several levels to maximize the effect of parallel computation. The algorithms are implemented using message passing interface (MPI) and run on a PC cluster connected by high-speed network Myrinet.

Key Words— Dynamics Computation, Parallel Computation, PC Cluster, High Speed Network.

1 はじめに

人間型のリンク構造を持ったロボットの動力学計算は、ヒューマノイドやコンピュータグラフィックス中のヒューマンフィギュアの運動生成などへの応用が期待されている。しかし、通常の計算法では自由度の大きいヒューマンフィギュアの動力学計算には非常に長い時間がかかり、特にインタラクティブ性が要求される運動生成には適用することができない。筆者らはすでにリアルタイムでの運動生成に応用することを目指した動力学計算法を開発し [1]、さらにそれを用いた運動生成ツール“力学フィルタ”を提案している [2] が、計算時間の問題からリアルタイムでの運動生成は難しい。

本論文では、動力学計算をさらに高速化することを目的とした並列計算について述べる。動力学計算の並列化は、パラレルメカニズムの制御 [3] や動力学シミュレーション [4] を目的とした研究が行われているが、ここでは Fig.1 のように多数のフィギュアが互いに干渉し合いながら運動する系を対象に、アルゴリズムの並列化および実装法を提案する。

2 動力学計算の並列化

文献 [1] の動力学計算で中心となるのは以下の 2 つの計算である。

1. d'Alembert の仮想仕事の原理を用いた拘束力の計算
2. 単位ベクトル法による慣性行列の計算

このうち 2. の計算は慣性行列の各列で独立に行うことができるのは明らかであり、容易に並列化できる。しかし、複数のフィギュアが干渉したときは慣性行列が非常に大きくなり、その逆行列の計算量が膨大になるという問題があった。一方、1. の計算は干渉しているフィギュア群でシリアルに行わざるを得ず、並列化は難しい。本節では、フィギュア毎に並列に計算できる部分をできる限り広げ、並列計算の効果を高めたアルゴリズムを提案する。

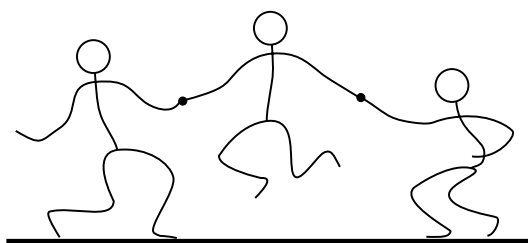


Fig.1: Multiple human figures with constraints

文献 [1] で述べた方法に従いフィギュア間の干渉を仮想リンクにより表現する。 k 番目の仮想リンクがフィギュア L_{k1} と L_{k2} のリンクを接続しているとする。フィギュア数を N_{fig} としたとき、フィギュア i ($i = 1, 2, \dots, N_{fig}$) の運動方程式は以下ようになる。

$$A_i \ddot{\theta}_{Gi} + b_i = H_{Ji}^T \tau_{Ji} + \sum_{k=1}^{N_{col}} H_{Cki}^T \tau_{Ck} \quad (1)$$

また、仮想リンク k における運動学的拘束条件は

$$H_{CkL_{k1}} \ddot{\theta}_{GL_{k1}} + H_{CkL_{k2}} \ddot{\theta}_{GL_{k2}} + \dot{H}_{CkL_{k1}} \dot{\theta}_{GL_{k1}} + \dot{H}_{CkL_{k2}} \dot{\theta}_{GL_{k2}} = \ddot{\theta}_{Ck} \quad (2)$$

- A_i : 慣性行列
- b_i : コリオリ力, 重力, 遠心力
- θ_{Gi} : 一般化座標
- τ_{Ji} : 関節トルク
- $H_{Ji} \triangleq \partial \theta_{Ji} / \partial \theta_{Gi}$
- θ_{Ji} : 関節値
- τ_{Ck} : 干渉点における拘束力
- $H_{Cki} \triangleq \partial \theta_{Ck} / \partial \theta_{Gi}$
- θ_{Ck} : 仮想リンクの関節値
- N_{col} : 仮想リンク数

これを全てのフィギュアについてまとめると以下の式が導かれる．

$$\begin{pmatrix} \mathbf{A} & -\mathbf{H}_C^T \\ \mathbf{H}_C & \mathbf{O} \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\theta}}_G \\ \boldsymbol{\tau}_C \end{pmatrix} = \begin{pmatrix} \mathbf{H}_J^T \boldsymbol{\tau}_J - \mathbf{b} \\ \ddot{\boldsymbol{\theta}}_C - \dot{\mathbf{H}}_C \dot{\boldsymbol{\theta}}_G \end{pmatrix} \quad (3)$$

$$\begin{aligned} \mathbf{A} &\triangleq \text{diag}(\mathbf{A}_i) \\ \boldsymbol{\theta}_G &\triangleq (\boldsymbol{\theta}_{G1}^T \boldsymbol{\theta}_{G2}^T \dots \boldsymbol{\theta}_{GN_{fig}}^T)^T \\ \boldsymbol{\tau}_C &\triangleq (\boldsymbol{\tau}_{C1}^T \boldsymbol{\tau}_{C2}^T \dots \boldsymbol{\tau}_{CN_{col}}^T)^T \end{aligned}$$

である．また， $\mathbf{H}_C$ は縦 N_{col} 個，横 N_{fig} 個のブロックからなる行列で，その (i, j) ブロックは $\mathbf{H}_{Cij}$ である．式 (3) の各構成要素はフィギュアまたは仮想リンク毎に並列に計算できる．式 (3) を $\boldsymbol{\tau}_C$ について解くと

$$\begin{aligned} \boldsymbol{\tau}_C &= -\mathbf{S}^{-1}(\mathbf{H}_C \mathbf{A}^{-1}(\mathbf{H}_J^T \boldsymbol{\tau}_J - \mathbf{b}) + \ddot{\boldsymbol{\theta}}_C - \dot{\mathbf{H}}_C \dot{\boldsymbol{\theta}}_G) \\ \mathbf{S} &\triangleq \mathbf{H}_C \mathbf{A}^{-1} \mathbf{H}_C^T \end{aligned} \quad (4)$$

が得られる．ここで $\mathbf{A}$ はブロック対角行列であるから，その逆行列は各ブロック $\mathbf{A}_i$ の逆行列を並列に計算することにより求められる．式 (4) から各仮想リンクで働く拘束力が計算されれば，各フィギュアの一般化加速度は式 (1) により並列に計算できる．

以上より，次の 2 つのレベルで動力学計算を並列化することができることがわかる．

1. $\mathbf{A}$ を計算する際の，自由度レベルでの並列計算
2. 運動方程式を解く際のフィギュアレベルでの並列計算

3 実装

並列アルゴリズムを実装する際の問題点として，プロセス (スレッド) 間の通信・待機によるオーバーヘッドがある．待機によるオーバーヘッドはスケジューリングを最適化することで減らすことができるが，通信のオーバーヘッドの改善にはハードウェア・ソフトウェア両面でのサポートが必要である．

近年，G-Bit Ethernet, Myrinet などに代表されるような高速ネットワークシステムが開発されている．さらに，ソフトウェア面でも message passing interface (MPI) と呼ばれるプロセス間通信ライブラリが整備されてきた．これらを用いて，多数の計算機で構成されたクラスタによる並列計算がさまざまな分野で行われている．本研究では，Myrinet で接続された PC クラスタにおいて，MPI を用いた並列計算を適用する．

MPI によるプロセス間通信を考慮したスケジュールを Fig.2 に示す．ここで灰色の四角形は MPI による通信を表す．全プロセスは 3 種類に分類される．唯一の Main process が全プロセスを管理し，シリアルに行う $\boldsymbol{\tau}_C$ を求める計算を担当する．Figure leader はフィギュア数だけあり，フィギュア毎の計算を並列に行う．その他のプロセスは各フィギュアの自由度数に応じて配分され，慣性行列の列を計算する．

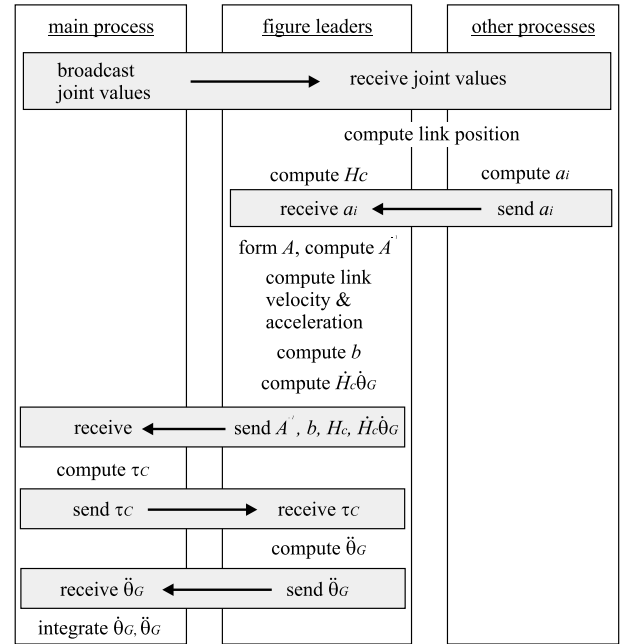


Fig.2: Sequence of computation using MPI; the gray boxes indicate MPI protocols

4 おわりに

本研究で得られた結果は以下の通りである．

- 従来の動力学計算法 [1] の並列性を改善したアルゴリズムを提案した．
- 提案したアルゴリズムを MPI により実装するためのプロセスのスケジューリングを行った．

本研究は (財) 製品科学研究センターからの受託研究「人間協調・共存型ロボットシステムの研究開発」の一環として行われた．

参考文献

- [1] 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.124–131, 1998.
- [2] 山根, 中村: “力学フィルタによるヒューマノイドの運動生成—HRP 仮想プラットホームへの適用—,” 第 17 回日本ロボット学会学術講演会予稿集, vol.3, pp.1213–1214, 1999.
- [3] K. Yamane, M. Okada, N. Komine, and Y. Nakamura: “Parallel Dynamics Computation and H_∞ Acceleration Control of Parallel Manipulators for Acceleration Display,” In *Proceedings of 1998 IEEE International Conference on Robotics and Automation*, pp.2301–2308, 1998.
- [4] A. Fijany, I. Sharf, and G.M.T. D’Eleuterio: “Parallel $O(\log N)$ Algorithms for Computation of Manipulator Forward Dynamics,” *IEEE Transactions on Robotics and Automation*, vol.11, no.3, pp.389–400, 1995.