

ハイブリッド制約処理系 HyLaGI の並列化

伊藤 剛史† 松本 翔太† 上田 和紀‡

† 早稲田大学大学院基幹理工学研究所 ‡ 早稲田大学理工学術院

1 はじめに

時間の変化に伴う連続的な変化と離散的な変化の両方を含む動的システムをハイブリッドシステム [1] と呼ぶ。ハイブリッドシステムは制御工学などの幅広い分野への応用が可能である。HydLa[2] は微分方程式を含む制約によりハイブリッドシステムをモデリングする宣言型言語である。HydLa の処理系である HyLaGI[3] は、パラメータを含むシステムの高信頼記号実行シミュレーションを特徴とする。Acumen, Flow*などの、他のハイブリッドシステムのシミュレータと比較して、記号実行などの優位性を持つ。

本研究は、HyLaGI がより複雑な問題に対応できるように、シミュレーション処理の並列化により高速化することを目的とする。多くの条件式を含む問題の条件判定の並列化やパラメータにより解軌道が分岐した場合の並列化を目指す。

2 HyLaGI

HyLaGI は数式処理を利用した HydLa の処理系である。開発言語は C++ で、現在のコード行数は 22,000 行程である。他のハイブリッドシステムのシミュレータと比較して、HyLaGI は以下の特徴を持つ。

- 数値計算ではなく数式処理を用いることで、計算誤差を排除している。
- 記号実行により、パラメータを含む、定性的に異なる軌道解を全解探索できる。

他のシミュレータと比較したデメリットとして、変数値を表す数式はシミュレーションが進むにつれて長くなる場合が多いため、シミュレーション速度では数値計算を利用したシミュレータに劣る。この点は、精度保証付き数値計算と数式処理の連携による対策を行っている。

3 提案方式

本研究では、2 種類の並列化を行い、HyLaGI を高速化することを目指す。

1 つ目はガード判定と呼ばれる、制約の有効・無効を判定するための処理の並列化である。各制約には、ガードと呼ばれる式を記述することができ、この式が満たされた場合にのみ後件の制約が有効となる。我々が既に行った拡張により、変数の連続性を利用した判定処理の省略 [4] が行われている。これらのガード判定はオブジェクトが多数あるような問題で、シミュレーション時間全体の 9 割以上を占める場合もある。この判定は各々のガード同士に依存関係が無いため、並列に行うことができる。このガード判定のアルゴリズムを図 1 に示す。図中の guards は、判定の対象となるガードの集合であり、constraints store は現在成立している制約(前提条件)の集合である。

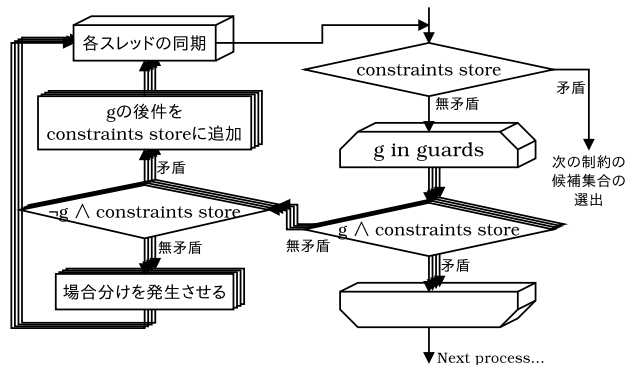


図 1 ガード判定のアルゴリズム

2 つ目は、パラメータにより解軌道が分岐した場合の、それぞれの軌道のシミュレーションの並列実行である。分岐の原因の 1 つは、図 1 において $g \wedge \text{constraints store}$ と $\neg g \wedge \text{constraints store}$ の両方が無矛盾となった場合、 g が成立する場合としない場合とで場合分けが発生することによる。HyLaGI は深さ優先探索により各解軌道を求める。これらの軌道のシミュレーションの並列化により、分岐が大量に発生するモデルのシミュレーションを高速化する。

並列化した HyLaGI の構成図を図 2 に示す。図において、1 つ目の並列化は Phase Simulator から複数の Backend を使用して判定処理を行うことを示し、2 つ目の並列化での深さ優先探索は Sequential Simulator で行う。Mathematica Engine は別のプロセスである。Sequential Simulator において、各 Phase Simulator に割り当てる

Parallelization of HyLaGI: a hybrid constraint processing system
 †Takeshi ITO †Shota MATSUMOTO ‡Kazunori UEDA
 †Graduate School of Fundamental Science and Engineering, Waseda University
 ‡Faculty of Science and Engineering, Waseda University

Backend を管理する.

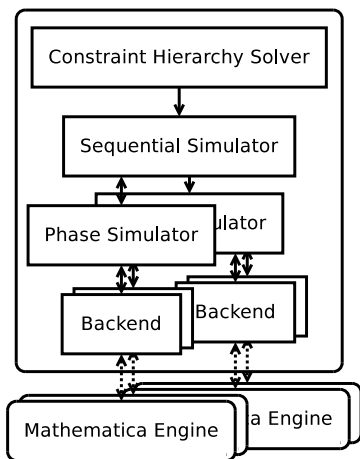


図 2 並列化した HyLaGI の構成図

4 評価と考察

現状、実装が完了しているのはガード判定の並列化である。この効果を測定するため、ベンチマークとして一次元ビリヤードのモデルを使用した。このモデルでは、一次元空間上にボールが 300 個並んでいる。最初のフェーズでは、任意の 2 個のボールについて衝突の判定が行われるため、ボールの数を n とすると $O(n^2)$ 回の判定が行われる。この判定は全体の実行時間の 8 割を占める。100 フェーズまで実行した場合のガード判定のベンチマーク結果を図 3 に示す。

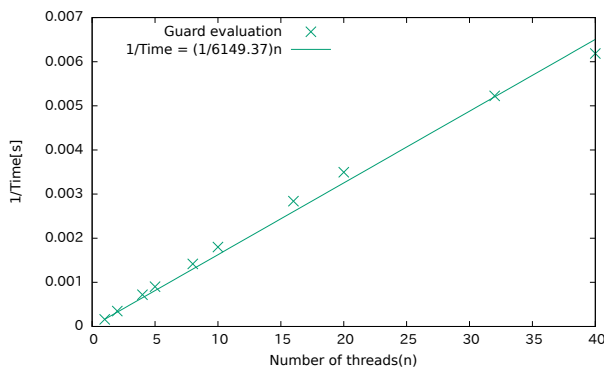


図 3 ビリヤードのモデルの速度向上

図 3 はガード判定にかかる時間の逆数と、スレッド数との関係を示しており、概ね直線となっていることが分かる。このことから、ガード判定の並列化について、ガードの個数が十分に大きい場合は期待通りスレッド数倍の性能を発揮していると言える。

次に、複数のモデルについてベンチマークをとり、ガード判定が 1 スレッドのときと比べて何倍速くなったかを計測した結果を図 4 に示す。

Cars は 20 台の車が距離を保とうとブレーキ・アクセ

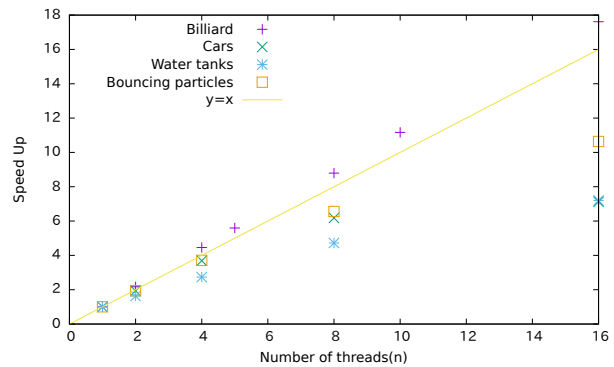


図 4 複数のモデルの 1 スレッド時に対する速度向上

ルを制御するモデル、Water tanks は水が流出する 10 個のタンクに対し水量が一定以下になったタンク 1 つに注水を行っていくモデル [5]、Bouncing particles は 100 個の質点が互いに干渉せずに床に対し弾んでいるモデルである。ビリヤードのモデルと比較するとガード条件の個数が少なく、他のモデルではスレッド数を増やしても速度の伸びは鈍化するが、高速化されていることが確認できる。

5 まとめ

ハイブリッドシステムをモデリングする言語 HydLa の処理系 HyLaGI の並列化について、ガード判定とケース分岐の並列化という 2 つの手法を提案した。ガード判定の並列化については実装を行い、その効果を確認した。分岐した解軌道のシミュレーションの並列実行は実装中である。

参考文献

- [1] Lunze, J. Handbook of Hybrid Systems Control: Theory, Tools, Applications, Cambridge University Press, 2009.
- [2] 上田和紀, 石井大輔, 細部博史. 制約概念に基づくハイブリッドシステムモデリング言語 HydLa, SSV2008 (第 5 回システム検証の科学技術シンポジウム), 2008.
- [3] 松本翔太, 上田和紀. ハイブリッド制約言語 HydLa の記号実行シミュレータ Hyrose, コンピュータソフトウェア, Vol.30, No.4, pp.18-35, 2013.
- [4] 河野 文彦, 小林 輝哉, 松本 翔太, 上田 和紀. 数式処理に基づくハイブリッドシステムシミュレータ Hyrose の大規模モデルシミュレーションに向けた拡張, 日本ソフトウェア科学会第 31 回大会, 2014.
- [5] John Lygeros, Claire Tomlin, and Shankar Sastry. Hybrid Systems: Modeling, Analysis and Control., 2008. <http://inst.cs.berkeley.edu/~ee291e/sp09/handouts/book.pdf>.