

# 仮想物理世界上で動く論理回路の実装と高速化の検証

電気通信大学 情報・通信工学科 成見研究室

1011114 瀬戸口 幸寿

## 1 はじめに

近年、「分子コンピューティング」とよばれる DNA などを用いて論理回路を設計し、計算機として用いる試みが話題を集めている。分子コンピューティングによって実装された分子コンピュータは、従来の計算機では解くことが困難だった問題を解くことができると期待される一方で、現状では動作が遅い、制御が困難といった課題が存在する。また全てが分子で出来ているコンピュータでは、予測不可能な問題が発生する可能性がある。これらのことは、理論的に分子コンピュータの挙動を議論することができ、実験から理論の真偽を確かめることはできても、分子コンピュータが計算機として実際に駆動する様子を直接確認し続けることが困難であることに由来する。本研究の目指す仮想物理世界でコンピュータを動かすことは、分子コンピュータで発生しうる問題を事前に洗い出し、研究に寄与する可能性がある。

別の話題として人工生命がある。既存の人工生命は、設計者によって定められた世界が存在し、その枠を超えた挙動を示すことはない。この事実は、人工生命のモデルを複雑に定義する方針を取る限り、生物らしい挙動を示すことに限界が存在することを意味する。そこで別の方針として、人工生命のモデルではなく人工生命の存在する環境をより複雑に、よりリアルに定義することで、環境の微妙な差異による影響を受けて各個体が挙動に再現性を持たない、より生命的な人工生命が生まれる可能性がある。

将来前述のような分野に役立つことを目的として、本研究では仮想物理世界上で駆動するコンピュータの最も基本的な部品である機械式論理回路を、ゲームエンジンである Unity を用いた仮想物理環境上で動作させた。また、コンピュータとしてより高速な駆動を目指す為に、仮想物理世界で機械式論理回路を動かすのに不可欠な剛体シミュレーションを行う独自の物理エンジンを自作し、同条件の仮想物理世界下で Unity と計算速度を比較した。

## 2 既存研究

原らは物理エンジン「PhysX」による仮想物理世界の構築を行い、AND 回路、OR 回路、NOT 回路、および NAND 回路と配線パーツを設計して基礎的な回路を実装した

[1]。作成した回路による実験から、より複雑な構造を構築するために、論理回路の高速化、および安定性を改善するのに論理回路の設計を見直す必要性を提唱した。また PhysX は GPU によるハードウェアアクセラレートに対応しているが、剛体シミュレーションにおいては無効になることを明らかにした。さらにロボットアプリケーション作成用のツールである「Microsoft Robotics Developer Studio(MSRDS)」の導入により高速で安定な論理回路が実装できる可能性を示した。本研究はゲームエンジン「Unity」[2]を用いて、壊れることなく高速に駆動する回路を作成し、より大規模な論理回路を作成する点、自作の物理エンジンを用いた場合と Unity を用いた場合での剛体シミュレーションの速度比較を行う点において、新規性を主張する。

## 3 研究手法

### 3.1 機械式論理回路の実装

NAND ゲートの実装を図 1 のように行った。入力を受け取る青と赤の剛体を入力ロッド A または B、出力を行う緑色の剛体を出力ロッドとする。入力ロッドは押し込まれた状態が入力 1、そうでない場合は入力 0 である。出力ロッドは突き出た状態が出力 1、引っ込んだ状態が出力 0 である。入力ロッドはバネのような一定の力が下向きに働いていて、押されない限りは入力 0 になっている。黄色い剛体は入力ロッドより小さい力が上向きに働いており、普段は入力ロッドに抑えられている。入力が共に 1 になると黄色い剛体は上向きに働く力によって青緑色のセンサに触れて出力ロッドに下向きの力が働く。即ち、出力が 0 になる。

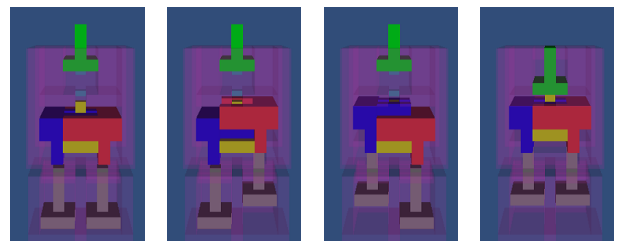


図 1 NAND 回路

次に NAND 回路を用いて全加算機を実装した。図 2 は機械式的全加算機である。また実装した全加算機を 2 つ用いて 2bit 加算機を実装することにより、ある程度大規模な論理回路の駆動が出来ることを確認すると同時に、3bit 以上の論理回路を実装しようとする動作が急

激に重くなることから、大規模な機械式論理回路を高速に駆動するには物理計算の見直しが必要であることを確認した。

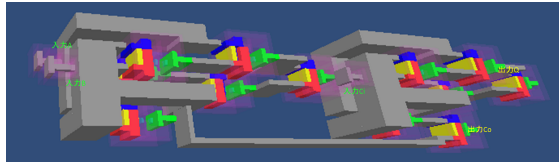


図2 全加算機

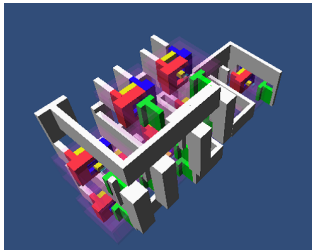


図3 リングオシレータ

原らの既存研究 [1] と比較する為、研究中で実装されたものと等価なリングオシレータを実装し、出力ロッドの変位プロットをFFTで振幅スペクトルに変換し周期が約5秒であることを確認した。既存研究で実装した機械式リングオシレータの周期は約128秒であるため、より高速な機械式論理回路を実装し、駆動出来たことを確認した。図3は実装したリングオシレータである。

### 3.2 剛体シミュレーションの高速化

機械式論理回路の大規模化と高速化の為に、剛体シミュレーションの高速化について検証を行った。論理回路を構成する剛体を球の集合による近似で表現することを想定して、本研究では球同士の剛体シミュレーションを行うシステムを実装し、Unityと比較を行う。球体の数を $n$ 個とすると、一つの球体が全ての球体と衝突判定を行うのにかかる時間は $n$ に比例し、全粒子の衝突判定を終えるには $n^2$ の時間を要する。そこで、CPUを用いて計算した場合とGPUを用いて計算した場合とで最適化を行う。CPUを用いた場合は、並進運動の計算を終えるたびに $x$ 座標でソートを行い、近い粒子とだけ衝突判定を行うようにする。GPUを用いた場合は、衝突判定を並列に処理することで高速化する。図4は粒子数を8192として自作したプログラムで剛体シミュレーションを行っている様子であり、図5はUnityで行っている様子である。図6は自作したプログラムとUnityでの計算速度の比較である。グラフから自作した剛体シミュレーション用のプログラムは計算に使用したデバイスがCPUであるか、GPUであるかに拘らずUnityによる場合より高速に計算が出来た。

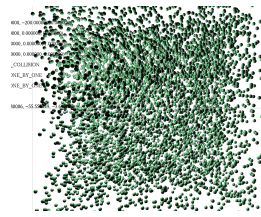


図4 自作システムによる剛体シミュレーションの様子

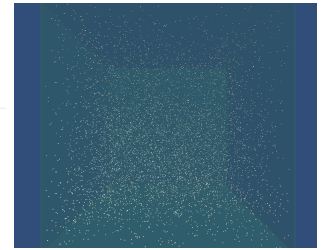


図5 Unityによる剛体シミュレーションの様子

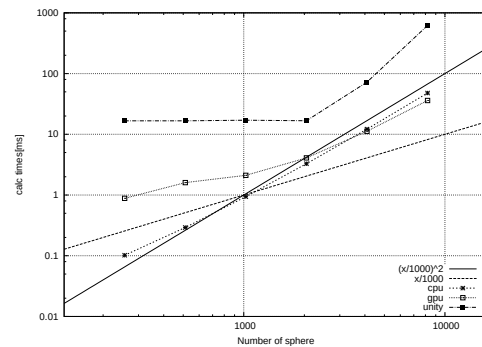


図6 速度比較

## 4 まとめ

### 4.1 達成したこと

Unityによって機械式論理回路を構築し、既存研究に対し高速かつ安定に駆動させることができた。また、球体同士の衝突シミュレーションをする自作の物理エンジンを構築し、Unityよりも高速に計算ができることを確認することで、将来より高速に機械式論理回路を駆動できる可能性を示した。

### 4.2 課題と展望

Unityで実装した機械式論理回路には、NANDゲート同士を繋ぐ配線パーツをその都度用意しており、パーツの種類が多くなる為見通しが悪い。また、回路の安定化を果たした一方で人工生命に繋がるような環境の影響を受けた挙動の変化は未だ見られない。今後は分子コンピューティングや人工生命といったニーズに則した、既存のものにとらわれない論理回路を設計する必要がある。また、剛体シミュレーションに関して、高速化が確認されたのは球体同士の場合のみであり、一般の剛体に対しても常に最適化された計算ができることを保証するものではない。今後は、一般の剛体に対しても高速化された物理エンジンを構成し、それを用いて機械式論理回路を実装する必要がある。

## 参考文献

- [1] 原 健一郎, 成見 哲, 小林 聡『高性能計算を用いた仮想物理世界での論理回路の実現』(電気通信大学2013)
- [2] Unity Technologies Japan『Unity Game Engine』  
<http://japan.unity3d.com>