解 説

ロボットシミュレータ開発

Development of Robot Simulator

金 広 文 男^{*1} 中 岡 慎一郎^{*1} 杉 原 知 道^{*2} 石 上 玄 也^{*3} 尾 崎 伸 吾^{*4}^{*1}産業技術総合研究所 ^{*2}プリファード・ネットワークス ^{*3}慶應義塾大学 ^{*4}横浜国立大学Fumio Kanehiro^{*1}, Shin'ichiro Nakaoka^{*1}, Tomomichi Sugihara^{*2}, Genya Ishigami^{*3} and Singo Ozaki^{*4}^{*1}National Institute of Advanced Industrial Science and Technology (AIST) ^{*2}Preferred Networks, Inc. ^{*3}Keio University^{*4}Yokohama National University

1. シミュレータとロボット開発

ロボット開発においてシミュレータを活用することには、(1) 試作の回数を削減し、試作のコストや試作に要する時間を削減できる、(2) 多数の実機がなくても多数の開発者やオペレータ要員が同時にプログラミングやトレーニングを行うことができる、(3) 開発途上のプログラムに含まれるバグや未熟なオペレータの操作によってロボットやその周囲に被害を与えることを防ぐことができる、(4) 実機で試すことができないような極限的な状況でのトレーニングを行うことができる、(5) ロボットによる作業計画を事前に検証できる、(6) 多くの試行回数を必要とする学習を多数のシミュレーションを並列実行することで効率的に行えるといった多くの利点がある。

一方で (1) シミュレーションのアルゴリズム上の近似やシミュレータに与えるロボットや環境のモデルに含まれる誤差といった要因によりシミュレーション結果が現実世界と 100%一致するとは限らない、(2) 環境やロボットのシミュレーションモデルが複雑であったり、物体の変形や流体の挙動等を正確にシミュレーションしようとしたらするとシミュレーションの計算速度が大幅に低下し、インタラクティブ性が必要とされる遠隔操作ロボットのシミュレーションには適さなくなる等の制限も存在する。従って上記の利点をうまく活かせる場面でシミュレータを活用することが重要である。

2. シミュレーションソフトウェア

シミュレータの中核となるのはユーザプログラムから指示された力やトルクをシミュレーション世界に存在する物体に適用し、シミュレーション世界の状況がどのように変化するかを計算する物理計算エンジンである。筆者らがシミュレータの開発を始めた 2000 年頃には DADS や ADAMS [1] な

どの商用の機構解析ソフトウェアに組み込まれた物理計算エンジンを使用するか、自前で開発する以外の選択肢は非常に少なかった。今日では無償・有償で使用可能な選択肢が多数存在する。オープンソースで公開され、無償で利用が可能な物理計算エンジンとして、ODE [2] や Bullet [3], PhysX [4], DART [5], Unreal Engine [6] を、商用の物理計算エンジンとして AgX [7] や Vortex [8] を挙げることができる。Bullet や Unreal Engine の様にゲーム用途を想定して精度よりは速度を優先したものや、産業用途を想定し精度を優先したものなどがあり、エンジンによって性格が異なる。また、どのエンジンも剛体のシミュレーションを基本機能として持っているが、そこに、ワイヤ、クローラ、キャラクターの皮膚の変形、飛行体等通常の剛体シミュレーションでは取り扱えない要素のシミュレーション機能を追加することで付加価値を出している。またこれらの物理計算エンジンに、シミュレーション世界の可視化機能やユーザプログラムとの接続機能を加えた統合シミュレーション環境として商用の V-REP [9] (教育用途は無償) や無償で利用可能な Isaac [10], Unity [11] (有償版もあり)、オープンソースの gazebo [12] やその後継の Ignition [13], choreonoid [14] が存在し、ユーザはシミュレーション世界を構成するロボットや環境のシミュレーションモデルとロボットを制御するためのソフトウェアを用意すればシミュレータの利用を開始することができる。ただしシミュレーションモデルの記述方式やソフトウェアを接続するためのミドルウェアについては、ロボット用ミドルウェアである ROS [15] およびその記述方式がデファクトにはなりつつあるものの、統一されていないのが現状である。ユーザは物理計算のエンジンの特徴や、既存のソフトウェア資産との相性によって使用する物理計算エンジン/統合シミュレーション環境を選択することとなる。

3. ImPACT-TRC におけるシミュレータ開発

ImPACT-TRC [16] では 5 種のプラットフォームロボットの開発が行われたが、前記のシミュレータの利点および制約を考慮し、ロボットが大型で開発コストが高く、ロボットが誤動作した場合に自身および周囲に与える被害が大きくなることが予測される脚ロボット [17] および建設ロボット [18] を主な利用対象としてシミュレータの開発を行った。災害現場で活動する災害対応ロボットのシミュレーションには以下のような特徴がある。

原稿受付 2019 年 7 月 11 日

キーワード: Simulator, Tough Robotics Challenge, Choreonoid

^{*1}〒305-8560 つくば市梅園 1-1-1^{*2}〒100-0004 千代田区大手町 1-6-1^{*3}〒223-8522 横浜市港北区日吉 3-14-1^{*4}〒240-8501 横浜市保土ヶ谷常盤台 79-5^{*1}Tsukuba-shi, Ibaraki^{*2}Chiyoda-ku, Tokyo^{*3}Kohoku-ku, Yokohama-shi, Kanagawa^{*4}Hodogaya-ku, Yokohama-shi, Kanagawa

図1 シミュレーション世界における相互侵入(左)と現実世界における相互圧縮(右)

- (1) 複雑な形状を有するロボットと瓦礫の接触計算を正しく行わなければならない
 - (2) ロボットの移動に伴う路面の変形や掘削など、剛体のみならず大きく変形する物体を取り扱わなければならない
 - (3) 雨や霧、煙、炎など災害対応ロボットの視覚機能に影響を与える様々な自然現象を再現しなければならない
- これらの課題に対応するため、(1) 立体積を用いた接触力計算[19]、(2) ミクロ・マクロアプローチによる掘削シミュレーション[20]、(3) GLSLを用いた自然現象シミュレーション[21]の開発を行った。以下にその概要を紹介する。

3.1 立体積を用いた接触力計算

既存の多くの物理計算エンジンでは、物体形状を三角形ポリゴンのメッシュとして表現し、干渉チェックライブラリがメッシュ間の干渉を検出、干渉点、干渉深さ、干渉法線を計算し、それらを用いて Projected Gauss Sidel 法などを用いて接触力を計算する方法がとられてきた。しかしこの方法では不正確な干渉法線が計算され、その結果として物体が不正確な挙動を示す場合があった。

本計算手法では、シミュレーション世界において物体同士が干渉している際の相互侵入が現実世界では相互圧縮に対応すると考え(図1参照)、この相互圧縮を補償するための速度を発生する力積が物体間に生じると考える。立体積の各切片の法線ベクトルを面積比で重み付き平均化し、これを物体間のノミナル接触平面の法線ベクトルとする。次に相互圧縮を補償するため、すなわち立体積の各切片をノミナル平面上に押し戻すために必要な切片の速度をノミナル接触平面上で積分し、必要な物体間の相対速度、角速度を決定する。この相対速度を発生するために必要な力積を最小二乗推定によって求め、接触力を決定する。

図2に本計算方法の適用例を示す。青矢印が法線方向を表しており、従来手法では干渉法線方向が誤判定された結果不正確な接触力が計算されている。一方で本計算手法は正しい法線方向が計算されていることが確認できる。

3.2 ミクロ・マクロアプローチによる掘削シミュレーション

掘削シミュレーションを行う手法にはマクロアプローチ、ミクロアプローチという二つのアプローチがある。マクロアプローチとは、掘削を行う機械要素を離散化し、その各領

図2 従来手法(左)と立体積を用いた手法(右)で計算された法線方向(青矢印の方向)の比較。従来手法で計算された方向は不正確でその結果不正確な接触力がロボットに作用するが、立体積を用いた手法では正しい方向が計算されている

図3 ミクロ・マクロアプローチ併用による軽量・高精度な掘削シミュレーション

域に加わる応力の積分から接触力を計算する手法であり、掘削によって土壌がどのように変形するかよりも掘削によって機械要素がどのような力を受けるかを計算することに焦点をおいた手法である。そのため、掘削によって土壌が不連続に変形するような現象の模擬はできないが、機械要素に作用する力を小さい計算コストで計算できるという利点がある。一方ミクロアプローチとは解析対象である土壌を離散粒子の集合体としてモデル化し、粒子同士の接触や滑動を含めた粒子の運動を追跡する手法であり、個別要素法(Distinct Element Method, DEM)とよばれる。不連続な変形を含めた土壌の変形を精密に再現することができるが、計算コストが非常に大きいという問題がある。

これら一長一短の二つのアプローチと実験装置を併用することで計算コストを抑えながら精度良く掘削現象を再現する仕組みを構築した。その構成を図3に示す。実験装置で取得したデータを用い、DEMモデルやマクロモデルの検証、パラメータ調整を行う。実験装置では検証できる機械要素や土壌のパリエーションに限りがあるが、ミクロアプローチを用いれば様々な条件でのバーチャルテストが可能である。実験データによって検証されたDEMモデルを用いることで、実験データが存在しない場合のバーチャルテストの結果も現実に近い値が得られると期待できるため、その結果をマクロモデルの調整に用いることで様々な状況で精度良くかつ計算コストの小さなマクロモデルを用いたシミュレーションを可能としている。図4に実験装置で検証したモデルを用いて、機械要素や土質を変更した場合で

図4 実験装置で検証したモデル(左上)を元に、機械要素を変更した例(右上)、土質を変更した例(左下)、両方を変更した例(右下)に展開

シミュレーションを行った結果のスナップショットを示す。

3.3 GLSL を用いた自然現象シミュレーション

GLSL (OpenGL Shading Language) を用いた描画エンジンを開発し、エンジン内のプログラマブルシェーダに GLSL で記述したプログラムを与えることで様々な自然現象シミュレーションを可能とした。

炎や煙等を再現する方法としてパーティクルシステムとよばれる方法を用いている。この方法は、画像テクスチャを貼り付けた点などの描画プリミティブを、与えられた法則に従って位置やサイズ、透明度等を変化させながら多数描画することで炎や煙等を表現する方法である。このような描画を CPU が行うことは非常に大きな負荷となるため、GPU を用いて描画を行うことは非常に有効であり、数千、数万のパーティクルを CPU に負荷をかけずに描画することが可能である。

パーティクルシステムの実装方法としてステートレスとステートフルがある。前者はパーティクルの挙動を数式で表す方法であり、初期状態と時刻の情報のみから任意の時刻のパーティクルの状態を計算できる。後者の方法は、パーティクルの全状態を記憶し、時刻の進展とともにそれらの状態を逐次的に更新する方法であり、より複雑なパーティクルの挙動を表現することが可能である。シミュレーションにおいては、シミュレーション結果を時刻を不連続に前後させながら確認したい場合がある。このような用途に対応するためには、ステートフルの実装方法の場合シミュレーション中の全時刻における全パーティクルの状態を記録しておくか、表示したいフレームまでの計算を再度行う必要が生じる。前者はメモリやストレージにかかる負担が大きく、後者は計算負荷が大きい。そこで Choreonoid における実装では表現力は劣るもののステートレスの方法を用いている。

図5に自然現象のシミュレーション例を示す。複数の光源によって生じる影に加えて、煙や炎など複数の自然現象が再現されているが、この程度であればノート PC 等に使用されているチップセット内蔵の GPU 程度の処理能力で

図5 プラント内で活動するロボットのシミュレーション例(左)、視覚を阻害する炎、煙、水流や影が再現されている。屋外で活動するロボットのシミュレーション例(右)、霧と雪が再現されている

も 30FPS 以上での描画が可能である。

3.4 活用事例

以上のような機能を有するシミュレータを用いて、ImPACT-TRC では、(1) 建設ロボットに搭載するカメラ配置の検討、(2) 建設ロボットが作業する際に車両が転倒する可能性の検討、(3) 脚ロボットを遠隔操作する際の遠隔操作結果の予測シミュレーション等に活用された。また開発成果は 2018 年に開催された World Robot Challenge [22] のインフラ・災害カテゴリーにおけるトンネル事故災害対応・復旧チャレンジ種目のシミュレーション競技でも活用された。

4. ま と め

本稿ではシミュレータのロボット開発における利点や制約、利用可能なシミュレーションソフトウェア、ImPACT-TRC に行ったシミュレーション機能の開発について紹介した。CPU や GPU の処理能力の向上に伴い、多くのシミュレーションソフトウェアが出現し、手軽にシミュレーションを研究に活用できる土壌が整って来ている。近年の AI 研究の発展に伴い、シミュレータを学習を効率的に行う手段として活用する事例が増えている [23]。シミュレーション世界と実世界が 100% 一致することはなく、シミュレーション世界で学習しても、結果がそのまま実世界で使えないことが障害となっていたが、この Reality Gap 問題を克服する研究も行われている [24]。シミュレータの得意、不得意に注意しながら是非シミュレータを研究に役立てていただきたい。

謝 辞 本研究の一部は、ImPACT-TRC の一環として実施された。シミュレータ開発にご協力いただいた皆様に感謝する。

参 考 文 献

- [1] MSC Software Corporation: Adams—機構解析ソフトウェア, <https://www.mssoftware.com/ja/product/adams>
- [2] Open Dynamics Engine, <http://www.ode.org/>
- [3] Bullet Real-Time Physics Simulation, <https://pybullet.org/wordpress/>
- [4] NVIDIA Corporation: PhysX SDK, <https://developer.nvidia.com/physx-sdk>
- [5] DART: Dynamic Animation and Robotics Toolkit, <https://dartsim.github.io/>
- [6] Epic Games, Inc.: Unreal Engine, <https://www.unrealengine.com>

- com/
- [7] Algorix Simulation AB: AGX Dynamics, <https://www.algorix.se/products/agx-dynamics/>
 - [8] CM Labs Simulations: Vortex Studio Simulation Platform, <https://www.cm-labs.com/vortex-studio/>
 - [9] Coppel Robotics GmbH: V-REP Virtual Robot Experimentation Platform, <http://www.coppelrobotics.com/>
 - [10] NVIDIA Corporation: Isaac SDK, <https://developer.nvidia.com/isaac-sdk>
 - [11] Unity Technologies: Unity, <https://unity.com/>
 - [12] Open Source Robotics Foundation: Gazebo, <http://gazebo.sim.org/>
 - [13] Open Robotics: Ignition Robotics, <https://ignitionrobotics.org/>
 - [14] Choreonoid, <https://www.choreonoid.org>
 - [15] Open Source Robotics Foundation: ROS Robot Operating System, <http://ros.org/>
 - [16] S. Tadokoro: "Overview of the ImPACT Tough Robotics Challenge and Strategy for Disruptive Innovation in Safety and Security, Disaster Robotics—Results from the ImPACT Tough Robotics Challenge," S. Tadokoro ed., Springer Tracts in Advanced Robotics 128, pp.3–22, DOI:10.1007/978-3-030-05321-5_1, 2019.
 - [17] K. Hashimoto, S. Kimura, N. Sakai, S. Hamamoto, A. Koizumi, X. Sun, T. Matsuzawa, T. Teramachi, Y. Yoshida, A. Imai, K. Kumagai, T. Matsubara, K. Yamaguchi, G. Ma and A. Takanishi: WAREC-1—A Four-Limbed Robot Having High Locomotion Ability with Versatility in Locomotion Styles. Proc. of the 15th IEEE International Symposium on Safety, Security, and Rescue Robotics, pp.172–178, 2017.
 - [18] 吉瀧: "ImPACT タフ・ロボティクス・チャレンジにおける複腕建設ロボット", 日本ロボット学会誌, vol.33, no.10, pp.711–715, 2017.
 - [19] N. Wakisaka and T. Sugihara: Loosely-Constrained Volumetric Contact Force Computation for Rigid Body Simulation. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp.6428–6433, 2017.
 - [20] R. Yoneyama, T. Omura and G. Ishigami: "Modeling of Bucket-Soil Interaction Mechanics Based on Improved Resistive Force Theory," Proc. of the European-African REgional Conference of the International Society for Terrain-Vehicle Systems (ISTVS), 2017.
 - [21] 中岡, 金広: "ロボットシミュレータ Choreonoid の自然現象描画機能の開発", 日本機械学会ロボティクスメカトロニクス講演会 2017 予稿集, 2A1-Q11, 2017.
 - [22] World Robot Challenge (WRC) 2018, <https://worldrobotsummit.org/wrc2018/>
 - [23] J. Mahler and K. Goldberg: "Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences," Proc. of the 1st Annual Conference on Robot Learning (CoRL 2017), 2017.
 - [24] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff and D. Fox: "Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience," Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp.8973–8979, 2019.

金広文男 (Fumio Kanehiro)

1999 年東京大学大学院工学系研究科情報工学専攻博士課程修了。1998 年より日本学術振興会特別研究員。2000 年電子技術総合研究所入所。2001 年より組織改変に伴い独立行政法人産業技術総合研究所知能システム研究部門研究員。2007 年 10 月主任研究員。2007 年 4 月より仏国 LAAS-CNRS 客員研究員。2014 年 4 月より国立研究開発法人産業技術総合研究所知能システム研究部門ヒューマノイド研究グループ長、現在に至る。博士 (工学)。ヒューマノイドロボットのシステム構成法、全身行動制御に興味がある。
(日本ロボット学会正会員)

杉原知道 (Tomomichi Sugihara)

1999 年東京大学工学部機械情報工学科卒業。2004 年同大学院情報理工学系研究科知能機械情報学専攻博士課程修了。同年同専攻学術研究支援員。2005 年より 2007 年まで同専攻助手。九州大学高等研究院特別准教授。大阪大学工学研究科知能・機能創成工学専攻准教授を経て、2019 年 4 月より現職。日本ロボット学会ヒューマノイド・ロボティクス研究専門委員会委員長および開かれた知能研究専門委員会副委員長。Co-chair of IEEE-RAS Technical Committee on Humanoid Robotics。博士 (情報理工学)。ロボットの運動学、力学、運動計画と制御、操縦、ハードウェア・ソフトウェアシステム開発、人の運動計測と解析を専門とする。日本機械学会、IEEE の会員。
(日本ロボット学会正会員)

尾崎伸吾 (Shingo Ozaki)

2005 年九州大学大学院生物資源環境科学府生産環境科学専攻博士課程修了。2005 年 4 月より東京理科大学工学部機械工学科助教。2008 年横浜国立大学大学院工学研究院システムの創生部門助教を経て 2013 年より同准教授、現在に至る。この間、マサチューセッツ工科大学客員研究員 (2014 年度)、宇宙航空研究開発機構宇宙科学研究所客員准教授 (2015～2016 年度) を務める。各種材料の構成モデルの開発や接触・摩擦問題などの非線形解析を専門とする。博士 (農学)、博士 (工学)。

中岡慎一郎 (Shin'ichiro Nakaoka)

2006 年東京大学大学院情報理工学系研究科コンピュータ科学専攻博士課程修了。博士 (情報理工学)。2006 年 4 月より産業技術総合研究所知能システム研究部門研究員。2013 年 3 月より同主任研究員。2011 年 11 月より 1 年間英国エジンバラ大学客員研究員。2019 年 4 月より株式会社コレオノイド代表取締役。ヒューマノイドロボット、ロボットシミュレータの研究開発および事業化に従事。2008 年度日本ロボット学会論文賞、IEEE/SICE SII2012 Best Paper Award (Robotics) 受賞。
(日本ロボット学会正会員)

石上玄也 (Genya Ishigami)

2008 年東北大学大学院工学研究科航空宇宙工学専攻修了。同年 Massachusetts Institute of Technology Postdoctoral Associate。2010 年より宇宙航空研究開発機構宇宙科学研究所研究員を経て、2013 年より慶應義塾大学理工学部機械工学科専任講師。2017 年より同准教授。現在に至る。宇宙探査をはじめとしたフィールドロボティクス、テラメカニクス、自律移動制御の研究開発を専門とする。IEEE RA-L Associate Editor, Journal of Terramechanics Editor, FSR2019 General chair などの学会活動に従事。博士 (工学)。
(日本ロボット学会正会員)