

複数ロボットによる SLAM シミュレーションの性能評価

大野 雄杜[†] 福井 誠人[†] 菅谷 みどり[†] 大川 猛[‡]

東海大学情報通信学部組込みソフトウェア工学科[†]

芝浦工業大学工学部情報工学科[‡]

1 はじめに

SLAM (Simultaneous Localization and Mapping: 自己位置推定マッピング) シミュレーションは、実際のロボットの振る舞いを理解し、より知的なロボットを実現するための手段として普及している。しかし SLAM 処理の計算量の多さによりシミュレーションが長時間化する問題がある。文献[1]ではこの問題に対し、ボトルネックとなるスキャンマッチング (SLAM の自己位置推定手法) 処理を FPGA に上実装し、ソフトウェア単体と比較し 1.7 倍から 2.8 倍の性能向上を達成した。

本研究では、文献[1]のような FPGA 等専用の集積回路を用いずエッジ計算機環境 (Ubuntu を搭載した SLAM 処理を行う PC) を用いて SLAM シミュレーションの性能評価を行い、複数ロボットに必要なエッジ計算機環境の処理性能に関する指針を得ることを目的とする。

本稿では、複数のエッジ計算機環境での 3 次元 SLAM 処理の時間計測結果について報告する。

2 SLAM シミュレーションの構成

本稿での 3 次元 SLAM シミュレーションは、ロボットとセンサのデータを取得しやすく連帯に優れた Gazebo (3 次元ロボットシミュレータ) を基軸にロボット用ソフトウェアプラットフォームである ROS (Robot Operating System) と ROS アプリケーションから構成した。SLAM アルゴリズムには未知環境から自己位置推定を行えるアルゴリズムの中でも応用例が多く、扱いやすい GMapping (Grid Mapping) [2] を使用することとした。

GMapping を利用した応用例として文献 [3] では、LiDAR、全天球カメラ (360 度の撮影が可能なカメラ)、3D スキャニング装置を用いて GMapping による 2D マップから強化学習 (現状からさらなる最善の行動を行う機械学習の手法) を行い実世界

の 3D コンテンツを作成している。また、Gmapping に関して、作成地図および CPU の消費率に絞った性能評価をする研究はされている [4] が、処理性能を分析したものは見当たらなかった。

シミュレーションする自立型ロボットには本研究に応用可能なサンプルプログラムを含むロボットプラットフォームである TurtleBot3 を使用した。なお SLAM に使用するセンサデータは TurtleBot3 の自立型ロボット上の LiDAR (レーザー照射により物体の距離、性質を分析する技術) センサから取得する。

Gazebo の画面で自律走行ロボットと自律走行ロボットが存在するワールドマップをシミュレーションする。ロボットが取得したデータから GMapping による SLAM 処理が行われ、その結果を視覚化するために ROS のアプリケーションである rviz を使用した。以上の 3 次元 SLAM シミュレーションの構成を以下の図 1 に示す。

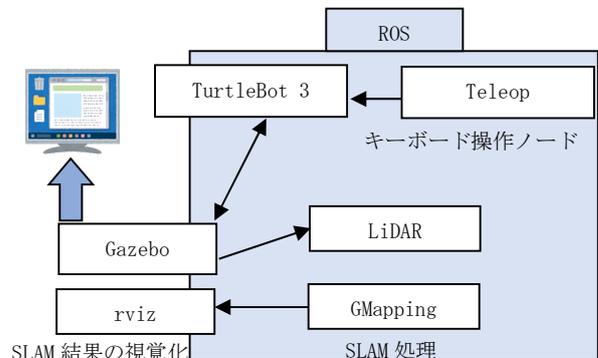


図 1: 3次元 SLAM シミュレーション構成図

3 評価

3 次元 SLAM シミュレーションの性能を評価するにあたり、GMapping のソース上から addScan コールバック関数に着目し、addScan 関数の処理時間を計測することにより毎回の SLAM シミュレーションにかかる時間を計測した。時間計測にはシミュレーションの正確な時間を計測するために ROS が提供する `ros::Time` 関数を使用した。

複数ロボットの SLAM シミュレーションに必要な処理性能の指針を得ることを目的とし、処理時間の計測による性能評価を行った。性能評価にあたり、表 1 に示す主に個人で使う PC 計算機環境 (A)、主にデータセンターで用いられる計算機環

Performance Evaluation of SLAM Simulation by Multiple Robots[†]Yuto Ohno, Masato Fukui, Midori Sugaya, Takeshi Ohkawa[‡]Department of Embedded Technology, Graduate School of Information and Telecommunication Engineering, Tokai University, Department of Computer Science and Engineering, Graduate School of Engineering, Shibaura Institute of Technology

境(B)を用意した。

表 1: 計算機環境のスペック

	計算機(A)	計算機(B)
CPU	Intel Core i9 10900K @2.80GHz 10Core 20Thread	Intel Xeon Gold 6230 @2.10GHz 20Core 40Thread ×2 基搭載
メモリ 容量	DDR4 128GB	DDR4 188GB
GPU	NVIDIA GEFORCE RTX2080	オンボード
OS	Ubuntu 18.04.5	Ubuntu 18.04.5

本評価では、二台のエッジ計算機環境上の Gazebo 画面(図 1 左側)で、同じ速度でロボットを操作し前方の壁までの SLAM 処理が行われた上で、処理の負荷が安定するまでの時間を計測した。SLAM 処理が行われた結果は図 1 右側の rviz 上に表示される。3 次元処理時間の測定結果を図 2 に示す。

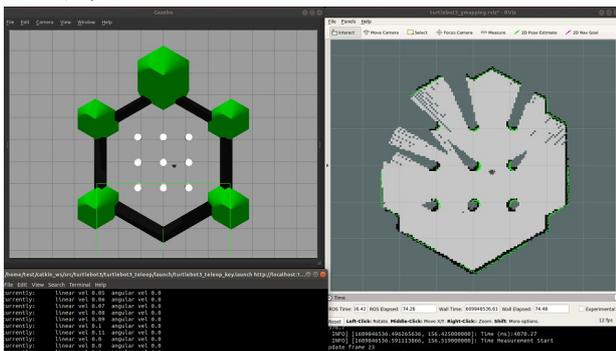
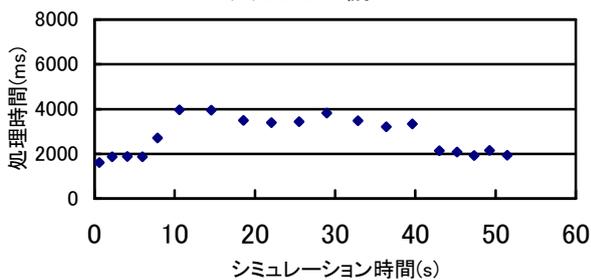


図 1: SLAM シミュレーションを行う計算機画面

(A)Core i9機



(B)Xeon Gold機

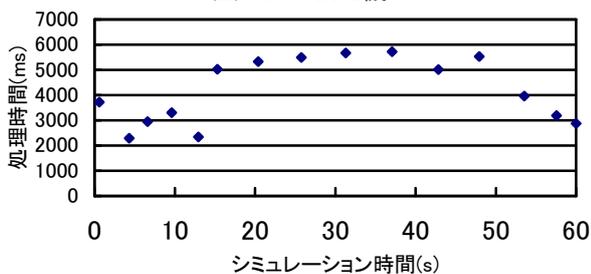


図 2: SLAM 処理時間の比較

(A)は、開始後しばらく処理時間は 2,000ms 程度であるが、10s 程度経過すると 4,000ms 程度に増加する。更に 40s 以降では、処理時間は 2,000ms 程度となる。これは、SLAM 処理において新たな特徴点が追加されなくなり処理量が減少したためと考えられる。

一方(B)は、開始時の処理時間が 3,000~4,000ms 程度であり、(A)の 1.5~2 倍程度である。そのためシミュレーション経過に伴い SLAM 処理を行う回数自体も(A)と比較して少ない(22 回→15 回)。また、15s 以降の処理時間は 6,000ms 程度と(A)の 1.5 倍程度である。

5 おわりに

本稿は、SLAM シミュレーションの性能評価から、複数ロボットの SLAM シミュレーションに必要な計算機環境の処理性能に関する指針を得ることを目的とし、複数のエッジ計算機環境での SLAM 処理時間の計測結果について述べた。

結果として、2 つの計算機環境で SLAM 処理時間を計測したところ、1.5 倍から 2 倍程度の SLAM 処理時間の違いがあった。具体的には、Core i9 を搭載した計算機の方が、Xeon を搭載した計算機よりも処理時間が短いことが分かった。このことから、SLAM シミュレーションのエッジ計算機環境を検討する際において、単一の SLAM 処理の場合 CPU のシングルスレッド性能に重点が置かれるが、複数の SLAM 処理を検討する際にはマルチスレッド性能が重要になると推測する。

なお、3 次元空間シミュレーションと SLAM シミュレーションの処理を、複数台のエッジ計算機環境で切り離した際のパフォーマンス評価は今後の課題である。

謝辞

本研究は、JST, CREST, JPMJCR19K1 の支援を受けたものです。

参考文献

- [1] 杉浦 圭祐, 松谷 宏紀, “2 次元 LiDAR SLAM アルゴリズムの FPGA による高速化”, IPSJ 研究報告 2020-SLDM-190, pp. 1-6, 2020.
- [2] https://github.com/ros-perception/slam_gmapping
- [3] 楊 夢龍, 長尾 確, “強化学習に基づく大規模屋内環境の 3D コンテンツの自動生成”, IPSJ 第 81 回全国大会講演論文集, pp. 229-230, 2019.
- [4] M. Rojas-Fernández, et al., “Performance comparison of 2D SLAM techniques available in ROS using a differential drive robot,” 2018 International Conference on Electronics, Communications and Computers (CONIELECOMP), pp. 50-58, Feb. 2018.