

通信の不安定性を考慮した遠隔操作ロボット制御方式の実装と評価

2022TC074 河部修輔

指導教員 宮澤元

1 はじめに

本研究は、通信の不安定性を考慮した遠隔操作ロボットの制御方式の実装と評価をする。従来、SLAMとよばれる自己位置推定と地図生成を行うシステムがありリアルタイム性の制限から主にローカルで実行されてきたが、通信状況に応じて SLAM の稼働場所をサーバーとローカルで動的に切り替えることにより、リアルタイム性と計算負荷の分散の両立が可能だと考える。本研究では、ROS2Jazzy と軽量 Kubernetes である k3s を用いて以下を検討する。

- 通信状態によって SLAM ノードの実行場所をローカルとサーバーで動的に切り替える機構
- 切り替え時および平常時における地図データの同期と更新データの共有方式
- 切り替えによる自己位置推定精度や遅延、リソース消費の影響等の評価

中間発表時点では基本的な切り替えプロトコルを構築済みであり、本要旨では現状の設計・実装を示す。

2 背景

SLAM と呼ばれる自己位置推定と地図生成を行う技術がある。近年の自律走行ロボットは、リソース運用の面やマルチタスク管理の面でクラウド環境下で SLAM の高精度かつリアルタイム性を求められている。しかし、現実の運用環境では以下のような課題が考えられる。

- 通信の乱れによる地図生成/自己位置推定の不安定化

安定化

- 通信断時のノードクラッシュや再初期化
- クラウド経由地図合成時のデータ競合や整合性低下

こうした状況下での地図の断絶やロボット制御の安定性に課題があった。この課題に対して本研究では、通信状況に応じたローカルでの軽量 SLAM とサーバーでの高性能 SLAM を通信状況に応じた切り替え機構を提案する。更にクラウド・ローカル間での地図データの共有・維持についても提案したいと考えている。

3 設計/実装

- k3s サーバー
k3s 内で ROS を、ROS 内では SLAM の高性能版を構築し動作させる。ここでは、ローカルである k3s エージェントの SLAM を k3s サーバーにオフロードするさせることで、k3s エージェントにかかる負荷の軽減と高精度な SLAM の使用のためのリソースの供給を可能にする。
- k3s エージェント (ローカル)
ROS を構築し ROS 内でロボット制御用のノードや SLAM、通信監視ノードを動作させる。通信が安定している時のシステムの動作はサーバーと通信させるため、Lidar を起動し通信監視ノードでサーバーとの通信を確認後取得データをサーバーに送信する。サーバー側で通信を受け取った後、データをそのまま

SLAM に通して地図生成をする. その地図をもとにローカルのロボットを制御するというものである. 通信が不安定になった場合は, 通信監視ノードが RTT とパケットロスを中心に基準を下回った時, 通信する場所をクラウドからローカル内の SLAM に変更する.

- 地図生成

地図データの共有に関しては, 最初はサーバーから地図生成を行うためサーバーからローカルへ全地図データを送信する. その後はサーバーで元の地図データとの差分だけを更新して新たな地図データのを生成して更新部分の地図データだけをローカルにも送信する. 通信が安定している時は, このような地図データの共有をする. 通信が不安定な場合は, ローカルの地図生成だけが進むためローカルの地図データの更新部分をサーバーに送信する. 加えて, 一定間隔で全地図データをローカルからクラウドに送信することで地図データの整合性を担保する.

以下がこのシステムの設計図である.

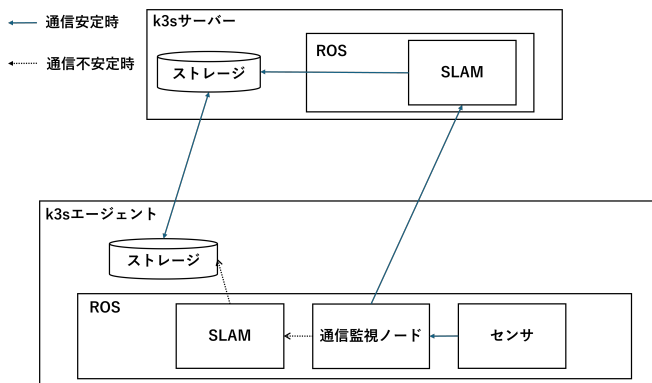


図 1. システム構成図

4 実験評価

本研究の実験については, 実際にシステムを稼働し linux コマンドを用いて RTT やパケットロス疑似的に下げることやサーバー側から離れた位置での通信等を想定している. その際の評価指標としては自己位置推定の揺らぎや地図データの誤差, 異常検知から SLAM ノード切り替え時間, リソース消費等を使

いたいと考えている.

表 1. 実験環境

項目	内容
OS バージョン	ubuntu24.04
ROS バージョン	Jazzy
SLAM	slam_toolbox
クラスタ	k3s (サーバ/エージェント)
エッジデバイス	raspberrypi5
ロボット	MentorPi

5 終わりに

本研究は, ROS2 と軽量 Kubernetes である k3s を組み合わせて, 通信状況に応じて SLAM 処理の位置を動的に切り替える分散型自律ロボットシステムとそのための効率的な地図同期戦略を提案した. 更には, k3s の導入によりリソースの効率デプロイやスケラビリティの拡張がされ障害児の迅速リカバリやリソースの最適化の構成を検討した.

今後は, システムを最後まで構築し実験を繰り返すことでこのシステムの有用性をはっきりとしたいと考えている. 以下の参考資料を参照した.

参考文献

- [1] Rancher Labs, "k3s: Lightweight Kubernetes", <https://k3s.io/>, 2019 年.