

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、ロボット制御における高負荷な作業はリアルタイム性の観点からローカルで実行されてきた。しかし、ローカルで高負荷な作業を行うには高精度なサービスの利用に必要なリソースが足りないことも考えられる。そこで、通信状況に応じて高負荷な作業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。

- 通信状態によって高負荷な作業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

2 背景

近年の自律走行ロボットは、リアルタイム性からローカルでほとんどの処理が行われることが多い。ただ、リソース運用の面やマルチタスク管理の面でサーバーを用いた場合がある。例えば、高負荷な作業をオフロードすることでより高精度なロボット制御を行うことや解析、複数のロボット制御等。しかし、ロボット制御にはどうしても高精度とリアルタイム性の両方を担保することが求められている。しかし、現実のサーバーでの運用環境では以下のような課題が考えられる。

- 通信の乱れによるセンサデータやロボット制御の不安定化
- 通信断時のノードクラッシュや再初期化

こういう理由から、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版のサービスとサーバーでの高性能サービス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・ローカル間でのサービスに用いる地図データの共有・維持についても提案したいと考えている。

3 設計

• サーバー

ROS 内で高負荷なサービスを構築し動作させる。ここでは、ローカルの自律走行用のサービスをサーバーにオフロードするさせることで、ローカルにかかる負荷の軽減することが目的である。更に、高負荷なサービスの使用のためのリソースの供給が可能のため、ローカルで動作させるより高精度なサービスの動作が可能である。

• ローカル

ROS を構築し ROS 内でロボット制御用のノードや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、データをそのまま自律走行用のサービスを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 とパケットロスをもと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自律走行用のサービスに変更する。

• 地図生成

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。通信安定時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。通信不安定時は、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サーバー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構成図である。

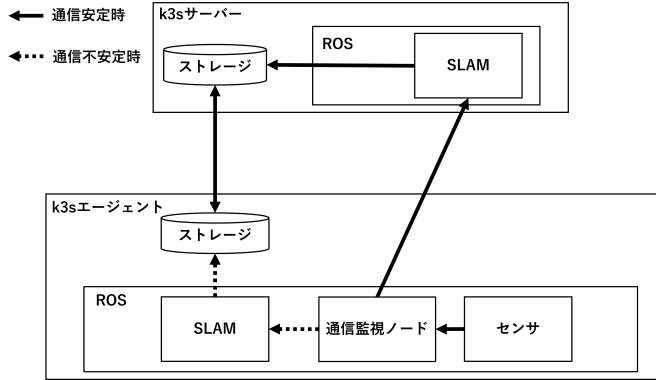


図 1. システム構成図

4 実装

サーバーとローカルの構成には、Kubernetes ディストリビューションの一つであり、軽量化されておりエッジデバイスでも使いやすい k3s を用いて、負荷分散や pod の自己修復機能等の役割を担う。更に、自律走行用のサービスとして今回は SLAM を用いて自立制御を行う。SLAM は SLAM Toolbox を用いることでサーバーとローカルで軽量版と高精度版で分けることが可能である。その k3s 上に ROS を構築し、ROS 上に SLAM Toolbox を構築する形にする。k3s サーバーと k3s エージェントは共にストレージをサーバー構築用の SQLite と地図を保管するための PersistentVolume を永続化したものを使うものとする。

5 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。実際の実験環境を以下に示す。

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

6 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ーラビリティの拡張がされ障害時の迅速リカバリやリソースの最適化の構成を検討した。現在は、まだ実装段階なので今後は、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。最終的には、切り替え機構の最適化にまで踏み込めたらと考えている。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.