

卒業論文

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

22TC074 河部 修輔

指導教員 宮澤 元

2026年1月

南山大学 理工学部 電子情報工学科

Title Foo and Bar

2022TC074 KAWABE Shusuke

Supervisor MIYAZAWA Hajime

January 2026

Department of Electronics and Communication Techonology Nanzan University

要約

本研究では、通信品質が不安定な環境においても自律移動ロボットが安定して自己位置推定および環境地図生成を行えるシステムの構築を目的とする。提案システムでは、ロボット内部で完結するローカル処理構成と、外部計算資源を活用するクラウド処理構成を用意し、通信状態に応じて自動的に処理構成を切り替える機構を設計・実装した。切り替え判断には、通信遅延や通信の安定性を示す静的・動的指標を組み合わせ、過剰な切り替えを防止する工夫を取り入れた。評価実験では、通信品質が変動する状況下で提案方式の有効性を検証し、ローカル・クラウドの処理切り替えによって自己位置推定精度の維持およびシステム全体の安定性向上が確認された。本研究により、通信依存性の高い遠隔操作ロボットにおいて、安定した運用を可能とする制御方式の設計手法が示された。

Abstract

This study aims to develop a mobile robot system capable of maintaining stable self-localization and environment mapping under unstable communication conditions. The proposed system employs two processing configurations: a local configuration that executes all computations onboard, and a cloud configuration that leverages external computational resources. An automatic switching mechanism between these configurations was designed and implemented based on communication quality indicators, including static and dynamic metrics for latency and stability, to prevent excessive switching. Experimental evaluations under variable communication conditions demonstrated the effectiveness of the proposed approach, showing that the switching mechanism preserves localization accuracy and enhances overall system stability. This research provides a practical design methodology for remote-controlled robotic systems that require reliable operation in communication-dependent environments.

目次

第 1 章	はじめに	1
1.1	研究の概要	1
1.2	研究の目的	2
1.3	本研究の貢献	2
1.4	本論文の構成	2
第 2 章	背景	3
2.1	自律移動ロボットを取り巻く社会的背景	3
2.2	自律移動ロボットにおける環境認識	3
2.3	計算資源とクラウド技術	4
2.4	通信環境とその課題	4
2.5	処理方式切替に関する考え方	5
2.6	既存研究と本研究の位置付け	5
第 3 章	設計	7
3.1	システム設計	7
3.2	切り替え機構の設計思想	7
3.3	通信品質指標に基づく切り替え設計	8
3.4	ローカル構成の設計	8
3.5	クラウド構成の設計	9
第 4 章	実装	10
4.1	システム実装	10
4.2	切り替え機構の実装方針	11
4.3	通信品質指標に基づく切り替え実装	11
4.4	ローカル構成の実装	12
4.5	クラウド構成の実装	12
第 5 章	実験	15
5.1	実験環境	15
5.2	通信品質変動下における切り替え動作の検証	16
5.3	動的 RTT 評価の有効性検証	17
5.4	クラウド SLAM 基盤の自動復旧性能評価	17
5.5	実験結果のまとめ	18
第 6 章	考察	19
6.1	通信品質に応じた処理構成切り替えの有効性	19
6.2	切り替え頻度とシステム安定性の関係	19

6.3	動的 RTT 評価方式の設計思想	19
6.4	k3s を用いたクラウド SLAM 基盤の信頼性	20
6.5	実運用を想定したシステム全体の評価	20
6.6	既存研究との比較	20
6.7	実運用環境を想定した評価結果の解釈	21
6.8	設計選択に関する考察	21
6.9	リアルタイム性と通信依存性のトレードオフ	21
6.10	オンボード計算資源制約の再評価	21
第 7 章	まとめ	22
7.1	研究全体の総括	22
7.2	卒業研究としての到達点	22
7.3	本研究で得られた知見	22
7.4	研究の意義と位置づけ	22
7.5	工学的意義と実用可能性	22
7.6	本研究の課題	23
7.7	今後の展望	23
参考文献		24

第1章

はじめに

本章では、本研究の背景および目的を概観し、研究全体の位置付けを明らかにする。近年の自律移動ロボットを取り巻く社会的・技術的状況を踏まえ、ロボットが周囲環境を認識しながら安定して動作するために求められる要件について整理する。

また、計算資源の制約および通信環境の不確実性が、自律移動ロボットのシステム設計にどのような影響を与えるかを示し、これらの課題に対して本研究がどのようなアプローチを取るのかを述べる。さらに、本研究の目的および貢献を明確化し、本論文全体の構成について説明する。

1.1 研究の概要

近年、物流倉庫や工場、医療・福祉分野、さらには災害対応現場など、さまざまな分野において自律的に移動可能なロボットの活用が進んでいる。これらの分野では、人手不足の深刻化や作業効率向上への要求を背景として、ロボットが周囲環境を認識し、人の介入を最小限に抑えながら移動・作業を行うことが強く求められている。

このような自律移動ロボットにおいては、周囲環境を把握しつつ、自身の位置を推定しながら行動する能力が不可欠である。そのためには、ロボットが搭載するセンサから得られる情報を用いて、自己の位置推定と環境情報の構築を同時に行う処理が重要な役割を担う。この処理は、ロボットの移動制御や経路計画の基盤となるため、多くの自律移動システムにおいて中核的に利用されている。

一方で、自己位置推定および環境地図生成を同時に行う処理は、計算量が大きく、リアルタイム性が強く求められる処理である。特に、高精度な推定を行う場合や、悪天候やガラス張りのような複雑な環境を対象とする場合には、演算負荷が増大し、ロボットに搭載された計算機の性能がシステム全体の性能を制限する要因となる。その結果、処理遅延や推定精度の低下が生じる可能性がある。

この問題に対する一つの解決策として、ロボット外部に存在する高性能な計算資源を活用し、計算処理の一部または全部をネットワーク越しに実行する方式が提案されている。この方式では、ロボット側の計算負荷を軽減しつつ、高度な計算処理を実現できる点が利点である。しかしながら、このような外部計算資源の利用は、通信品質に強く依存するという課題を抱えている。

通信遅延やパケットロスが発生した場合、処理結果の到達が遅延したり、処理自体が失敗したりする可能性がある。その結果、自己位置推定の不安定化や、ロボットの挙動の乱れにつながる恐れがある。特に、無線通信を利用する環境では、通信品質が時間的に変動することが多く、この影響を無視することはできない。

本研究では、このような背景を踏まえ、通信品質が変動する環境においても、自律移動ロボットが安定して自己位置推定および環境地図生成を継続可能なシステムの構築を目指す。具体的には、ロボット内部で処理を完結させる構成と、外部計算資源を利用する構成の二つの処理配置を用意し、通信状態に応じて適切な処理構成を選択する方式を提案する。

1.2 研究の目的

本研究の目的は、通信品質が時間的に変動する環境においても、自律移動ロボットが自己位置推定および環境地図生成を安定して継続可能なシステムを実現することである。

従来の外部計算資源を活用した方式では、通信状態が良好な場合には高い処理性能を得られる一方で、通信遅延や通信断が発生した場合には、処理結果の遅延や欠落が生じ、ロボットの動作が不安定になるという問題がある。一方で、すべての処理をロボット内部で行う構成では、通信に依存しない安定性を確保できるものの、計算資源の制約により、処理精度やリアルタイム性に限界が生じる。

これらの課題を踏まえ、本研究では、通信状態が良好な場合には外部計算資源を積極的に活用し、通信品質が低下した場合にはロボット内部で処理を完結させる構成へ切り替える方式を採用する。これにより、両者の利点を活かしつつ、それぞれの欠点を補完することを目指す。

特に本研究では、通信品質の一時的な変動に対して過剰に反応しない切り替え機構の設計に重点を置く。通信遅延やパケットロスは、短時間に発生・回復する場合が多く、瞬間的な通信状態のみを基準とした場合、不要な処理構成の切り替えが頻発する恐れがある。

そこで、通信品質を評価する指標として、通信遅延および通信の安定性を用い、それぞれの時間的特性を考慮した処理構成選択方式を提案する。これにより、処理の安定性とリアルタイム性の両立を図ることを本研究の目的とする。

1.3 本研究の貢献

本研究の主な貢献は、以下の三点にまとめられる。

第一に、通信品質の時間的特性を考慮した処理構成選択機構を設計し、自律移動ロボットシステムに適用した点である。通信遅延を動的に変化する指標として扱い、通信の安定性を補助的な指標として用いることで、一時的な通信変動と継続的な通信劣化を区別可能な切り替え方式を提案した。

第二に、ロボット内部で完結する処理構成と、外部計算資源を利用する処理構成を明確に分離したシステム構成を設計・実装し、通信品質に応じた柔軟な処理配置の変更を実験的に検証した点である。これにより、通信環境が不安定な状況においても、自己位置推定処理を継続可能であることを示した。

第三に、外部計算資源側において、処理を分散実行可能な基盤を構築し、処理の自動再起動や継続実行を可能とした点である。これにより、外部計算資源を用いる場合においても、システム全体の信頼性向上に寄与した。

1.4 本論文の構成

本論文の構成を以下に示す。

第2章では、自律移動ロボットおよびSLAMに関する背景技術について述べ、本研究の位置付けを明確にする。

第3章では、本研究で提案するローカル構成およびクラウド構成を切り替える自律移動ロボットシステムの設計について説明する。

第4章では、第3章で述べた設計に基づき、システムの実装方法について詳述する。

第5章では、提案システムの有効性を検証するために行った各種実験の内容および結果を示す。

第6章では、実験結果に基づく考察を行い、提案手法の有効性および課題について議論する。

第7章では、本研究のまとめと今後の課題について述べる。

第2章

背景

本章では、自律移動ロボットを取り巻く社会的背景および技術的背景について整理し、本研究が対象とする課題の位置付けを明確にする。まず、自律移動ロボットが社会的に求められている背景を概観した上で、ロボットが安定して自律的に動作するために必要となる環境認識および計算処理の要件について述べる。

次に、ロボットに搭載される計算資源の制約と、外部計算資源を活用するアプローチの可能性について整理し、それらが通信環境に依存するという課題を明らかにする。さらに、通信品質の変動がロボットシステムに与える影響を整理し、処理方式を切り替えるという考え方の有用性と、既存研究における位置付けについて論じる。

2.1 自律移動ロボットを取り巻く社会的背景

2.1.1 自律移動ロボットの社会的背景

近年、物流倉庫や工場、医療・福祉、さらには災害現場等において、自律移動ロボットの活用が急速に拡大している。これらの分野では、人手不足の深刻化や作業効率向上への要求を背景として、人が行ってきた作業をロボットに代替させる動きが進んでいる。特に、搬送や巡回といった反復的な作業においては、ロボットによる自動化が高い効果を発揮すると期待されている。

また、少子高齢化が進む社会においては、将来的な労働力不足が懸念されており、ロボット技術の活用は重要な解決策の一つと考えられている。このような背景のもと、自律移動ロボットには、限られた人手による管理で長時間運用できることや、予期せぬ状況においても安定して動作することが求められている。そのため、ロボットが人の介入に依存せず、自律的に環境を認識し判断できる能力の重要性が高まっている。

2.2 自律移動ロボットにおける環境認識

2.2.1 自律移動ロボットにおける認識の重要性

自律移動ロボットが人と共存する環境で安全に動作するためには、周囲の状況を正確に把握し、それに基づいて適切な行動を選択する必要がある。ロボットは自身の位置を誤って認識した場合、目的地への到達に失敗するだけでなく、障害物との衝突や危険な挙動を引き起こす可能性がある。そのため、環境認識と自己位置推定は、自律移動ロボットにとって最も基本的かつ重要な機能の一つである。

このような環境認識を実現する技術として、自己位置推定と環境地図作成を同時にを行う SLAM が広く用いられている。SLAM は、ロボットが未知環境を移動しながら、自身の位置を推定しつつ地図を生成するための枠組みであり、多くの自律移動ロボットシステムにおいて中核的な役割を果たしている。SLAM の結果は、経路計画や障害物回避などの上位機能にも利用されるため、その精度と安定性はシステム全体の性能に大きな影響を及ぼす。

2.3 計算資源とクラウド技術

2.3.1 ロボットにおける計算資源の制約

一方で,SLAM を含む環境認識処理は,多量のセンサデータをリアルタイムで処理する必要があり,高い計算負荷を伴うことが知られている. 実際の運用を想定した移動ロボットでは,小型化や軽量化が求められることが多く,搭載可能な計算機の性能には制約がある. また,バッテリ駆動で長時間稼働する必要があるため,消費電力の観点からも高性能な計算機の搭載は容易ではない.

このような制約の下では,ロボット単体で高度な処理を実行することが難しくなる場合がある. 計算資源が不足すると,処理速度の低下や精度の劣化が生じ,結果としてロボットの行動に悪影響を及ぼす可能性がある. そのため,ロボットのハードウェア制約を考慮しつつ,必要な処理性能をどのように確保するかは,自律移動ロボットの設計における重要な課題である.

2.3.2 クラウド技術の活用と期待

この課題に対する解決策の一つとして,近年ではクラウド技術をロボット分野に応用するクラウドロボティクスが注目されている. クラウド環境では,高性能な計算資源を柔軟に利用できるため,ロボット単体では困難な処理を外部で実行することが可能となる. これにより,ロボットのハードウェア構成を簡素化しつつ,高度な機能を実現できる可能性がある.

特に SLAM のような計算負荷の高い処理においては,クラウド側で処理を行うことで,高精度な結果を得られることが期待されている. また,複数のロボットから得られた情報を集約することで,より広範な環境認識が可能になるなど,クラウド利用ならではの利点も指摘されている. このように,クラウド技術は自律移動ロボットの性能向上に寄与する重要な要素と考えられている.

2.4 通信環境とその課題

2.4.1 通信環境の不確実性

しかしながら,クラウドを利用した処理は通信環境に強く依存するという課題を抱えている. 移動ロボットは多くの場合,無線通信を用いて外部と接続されるため,通信遅延や通信速度の変動,さらには通信断が発生する可能性がある. これらの問題は,屋内外の環境差や電波干渉,アクセスポイントとの距離など,さまざまな要因によって引き起こされる.

さらに,移動ロボットは常に位置を変えながら動作するため,通信環境も時間とともに変化する. このような通信品質の変動は,クラウドに処理を依存するシステムにとって大きな不確実要因となる. 通信が不安定な状況では,クラウド側での処理結果を適切なタイミングで受け取ることができず,ロボットの挙動に影響を及ぼす可能性がある.

2.4.2 通信品質低下が与える影響

通信品質が低下した状態でクラウド処理を継続すると,処理結果の遅延や欠落が発生する恐れがある. このような問題は,自己位置推定の精度低下や地図更新の遅れといった形で現れ,結果としてロボットの行動計画や制御に悪影響を及ぼす可能性がある. 特に,人や障害物が存在する環境では,認識の遅れが安全性の低下につながる危険性も否定できない.

このため,通信環境が悪化した場合にどのように対応するかは,ロボットシステムの信頼性を左右する重要な課題である. 通信状況に応じて処理方式を切り替えるといった柔軟な対応は,ロボットの自律性を高め,安定した

運用を実現する上で有効な考え方である。

2.5 処理方式切替に関する考え方

2.5.1 処理方式切替の有用性

通信品質の変動が避けられない環境において、单一の処理方式に依存したシステム設計は、安定した動作を維持する上で限界がある。特に、通信が安定している状況と不安定な状況では、ロボットに求められる処理の在り方が大きく異なる。このため、状況に応じて処理方式を切り替えるという考え方は、合理的かつ実用的な解決策の一つであると考えられる。

例えば、通信品質が十分に良好な場合には、外部の計算資源を活用することで、高精度な処理結果を得ることが可能となる。一方で、通信品質が低下した場合には、ロボット単体で完結する処理方式に切り替えることで、通信遅延や通信断の影響を回避できる。このように、複数の処理方式を適切に使い分けることで、システム全体の信頼性を向上させることが期待される。

また、処理方式切替は、計算資源や通信資源を効率的に利用するという観点からも有用である。常に高負荷な処理をクラウドに依存するのではなく、必要に応じて処理の分担を変更することで、無駄な通信や計算を削減できる可能性がある。このような柔軟な運用は、実環境における長時間運用を想定したロボットシステムにおいて重要な要素である。

以上の理由から、通信状況に応じて処理方式を切り替えるというアプローチは、自律移動ロボットの安定運用を実現する上で有効な手段であり、本研究が着目する基本的な考え方の一つである。

2.5.2 処理方式切替に関する既存の考え方

これまでに、通信品質を指標として処理方式を切り替える手法が検討されてきた。一般的には、通信遅延や応答時間などの指標を用い、それらがあらかじめ定められた基準を超えた場合に処理方式を変更するという考え方が採用されている。このような方法は、判断基準が明確であり、システム設計が比較的容易であるという利点を持つ。

また、固定的な基準を用いることで、処理方式の切替を単純化できる点も評価されている。しかしながら、このようなアプローチは、通信環境が常に一定であることを前提としており、実際の運用環境においては必ずしもその前提が成り立たない場合がある。

2.5.3 固定的な判断基準の課題

実環境における通信品質は、短時間で大きく変動することが多い。一時的な通信遅延の増加や瞬間的な品質低下は頻繁に発生し、これらをすべて異常として扱うことは現実的ではない。固定的な判断基準を用いた場合、このような一時的な変動に過剰に反応し、不必要的処理方式の切替が頻発する可能性がある。

このような頻繁な切替は、システム全体の安定性を損なう要因となる。また、異なる環境や運用条件に対して同一の基準を適用することは、環境適応性の観点からも課題が残る。このように、固定的な基準のみに基づく判断には限界がある。

2.6 既存研究と本研究の位置付け

2.6.1 クラウドロボティクスに関する既存研究

近年、ロボット分野においてクラウド技術を活用する試みが数多く報告されている。これらは一般に「クラウドロボティクス」と総称され、ロボット単体では実行が困難な計算負荷の高い処理を、ネットワークを介して

外部計算資源に委譲することを目的としている。

代表的な研究として,C2TAM (Cloud framework for cooperative tracking and mapping) が挙げられる。[1] C2TAM では、ロボットが取得したセンサデータをクラウドに送信し、地図最適化などの計算量の大きい処理をクラウド側で実行する構成が提案されている。これにより、ロボット搭載計算機の負荷を軽減しつつ、高精度な地図生成を実現している。

しかしながら、C2TAM をはじめとする多くのクラウド SLAM 研究では、通信遅延や通信品質変動の影響については課題として言及されるにとどまり、実際に通信状態が時間的に変動する環境下でどのように対処するかについては、十分に検討されていない場合が多い。特に、通信品質の変動に応じて処理方式そのものを切り替えるといった観点は、必ずしも研究の主題とはされてこなかった。

2.6.2 オンボード処理とクラウド処理の分担に関する研究

オンボード処理とクラウド処理の役割分担に関する研究も多数存在する。これらの研究では、一部の処理のみをクラウドへオフロードすることで、通信遅延の影響を低減しつつ計算負荷を分散する構成が提案されている。

一方で、処理分担を固定的に設計した場合、通信品質が想定より悪化した際にシステム全体の性能が低下する問題がある。特に無線通信を用いる移動ロボットでは、運用環境や移動経路によって通信品質が大きく変動するため、固定的な処理分担では実環境への適応性に限界がある。

2.6.3 通信品質を考慮したロボットシステム設計

通信品質を考慮したロボットシステムの設計に関する研究では、通信遅延や帯域制限がロボット制御に与える影響が議論されている。これらの研究では、通信遅延が制御安定性や応答性に悪影響を及ぼすことが示されており、通信環境を考慮した設計の重要性が指摘されている。

しかし、多くの研究は通信品質を「前提条件」として扱うことが多く、通信状態が時間的に変動する状況下でシステム構成そのものを動的に変更するという観点は十分に扱われていない。

2.6.4 本研究の位置付け

以上の既存研究を踏まえると、クラウドを活用した SLAM 処理による計算負荷軽減の有効性は示されている一方で、通信品質変動に対する実践的な対応手法は未だ十分に確立されていない。

本研究は、

- 通信品質が時間的に変動する実環境を前提とする点
- ローカル構成とクラウド構成を明確に分離し、動的に切り替える点
- 通信品質評価に基づき、切り替え回数そのものを抑制する設計を行っている点

に特徴がある。

特に、本研究では SLAM 手法そのものの改良を目的とするのではなく、SLAM を「どこで実行するか」というシステム構成の観点から問題を捉え直している点に独自性がある。

第3章

設計

本章では、本研究で提案する自律移動ロボットシステムの全体設計について述べる。本研究は、通信品質が時間的に変動する実環境を前提とし、そのような環境下においても自己位置推定および環境地図生成を安定して継続可能なシステム構成を目指すものである。

特に本章では、処理をロボット内部で完結させる構成と、外部計算資源を利用する構成という二つの処理方式をどのように設計し、どのような基準に基づいて切り替えるかについて、設計思想および判断指標の観点から整理する。

また、通信品質の変動に対して過剰に反応しないための切り替え機構の設計方針を示し、静的指標および動的指標を組み合わせた通信品質評価の考え方について説明する。さらに、外部計算資源を用いる構成においては、処理基盤自体の信頼性も重要であることから、クラウド側実行基盤の設計方針についても述べる。

3.1 システム設計

本研究では、ローカル構成およびクラウド構成の2つの処理構成を切り替えて利用する自律移動ロボットシステムを提案する。本システムは、ロボットが取得したセンサデータを用いて自己位置推定および環境地図生成を行う点では共通しているが、処理を実行する計算資源の違いによりローカル構成とクラウド構成に分かれている。

クラウド構成は高精度なSLAM処理を可能とする一方で、通信品質に強く依存する。そのため、通信状態が不安定な状況で頻繁に処理構成を切り替えることは、システムの安定性やリアルタイム性を低下させる要因となる。

本研究では、通信状態を表す指標を用いて処理構成を自動的に切り替える機構を設計し、不必要的切り替えを抑制しつつ、安定性と精度の両立を目指す。

3.2 切り替え機構の設計思想

本研究における切り替え機構の設計において最も重視する点は、通信状態の一時的な変動によって不必要的構成切り替えが発生することを防ぐことである。

通信環境は時間とともに変動するため、瞬間的な通信遅延の増加のみを基準として処理構成を切り替えると頻繁な切り替えが発生し、システム全体の挙動が不安定となる可能性がある。特にSLAM処理では、構成の切り替えに伴う処理状態の変化が、自己位置推定の精度低下を招く恐れがある。

そこで本研究では、通信品質を表す指標を時間特性の異なる複数の観点から評価し、切り替え判断に用いる。具体的には、通信状態の長期的な安定性を表す静的指標と、通信遅延の短期的な変動を捉える動的指標を組み合わせることで、不必要的切り替えを抑制する設計とする。

さらに、本研究では、切り替え判断そのものを目的とするのではなく、切り替え回数を必要最小限に抑えることを設計上の重要な指針とする。構成切り替えは、SLAM処理の再初期化や内部状態の変化を伴うため、頻繁に

発生すると自己位置推定の連続性が損なわれる可能性がある。

そのため、本研究では、通信状態が「悪化し始めた瞬間」ではなく、「一定時間以上にわたり悪化が継続している状態」を検出することを重視する。動的 RTT を時間的に評価する設計は、このような状態判定を可能とし、結果として不必要的構成切り替えの抑制につながる。

3.3 通信品質指標に基づく切り替え設計

本研究では、ローカル構成とクラウド構成の切り替え判断に用いる通信品質指標として、パケットロス率および RTT を採用する。これらの指標はそれぞれ異なる時間特性を持ち、切り替え設計において異なる役割を担う。

3.3.1 静的指標としてのパケットロス率

パケットロス率は、一定期間における通信の成功率を表す指標であり、通信経路の安定性を評価するために用いる。パケットロスが継続的に発生する環境では、クラウド構成におけるセンサデータや処理結果の欠落が生じやすく、SLAM 処理の信頼性が低下する。

そのため本研究では、パケットロス率を通信環境の良否を判断するための静的な指標として位置付け、一定期間の統計値に基づいて評価を行う。これにより、一時的な通信変動ではなく、通信環境そのものがクラウド利用に適しているかを判断する。

3.3.2 動的指標としての RTT

RTT は、ロボットとクラウド間の通信遅延を表す指標であり、ネットワーク負荷や混雑状況に応じて短時間で変動する特徴を持つ。

本研究では、RTT を切り替え判断における動的な指標として用いる。ただし、瞬間的な RTT の増加のみを基準として構成を切り替えると、通信状態の一時的な揺らぎに反応して不必要的切り替えが頻発する可能性がある。

そこで、RTT を時間的に評価し、一定期間における変動傾向を基に切り替えを判断する設計とする。これにより、一過性の遅延増加では切り替えを行わず、通信状態が継続的に悪化した場合のみローカル構成へ移行することが可能となる。

このように、RTT を動的に扱うことで、システムの安定性を維持しつつ、不要な構成切り替えを抑制することを本研究の目的とする。

本研究では、短時間で変動する RTT を主指標として扱い、通信環境そのものの良否を示すパケットロス率を補助的な指標として用いる点に特徴がある。

3.4 ローカル構成の設計

ローカル構成は、通信品質の低下が検出された場合においても、ロボットが自律的に動作を継続できるよう設計する。本構成では、センサデータの取得、SLAM 処理、および自己位置推定結果の利用をすべてロボット上で完結させる。

動的 RTT の評価により、通信遅延が継続的に増加していると判断された場合、ローカル構成へ切り替えることで、処理遅延による影響を回避する。

3.5 クラウド構成の設計

クラウド構成は、通信環境が安定している状況において、高精度な SLAM 処理を実現することを目的として設計する。パケットロス率が低く、かつ RTT が安定して低い場合に、クラウド構成を選択する。

動的 RTT を用いた評価により、通信遅延が一時的に増加した場合でも、即座に構成を切り替えることなく、クラウド構成を維持できる。これにより、不要な切り替えを抑制しつつ、高精度な処理を継続可能とする。

3.5.1 クラウド実行基盤の設計

本研究におけるクラウド構成では、SLAM 処理を単一の外部計算機に依存させるのではなく、クラウド実行基盤上のサービスとして提供することを設計方針とした。

通信品質が良好であっても、クラウド側の処理ノードが停止した場合には自己位置推定処理が継続できなくなる。このような事態を避けるため、クラウド構成においては、処理ノード自体の障害に対しても耐性を持つ構成が求められる。

そこで本研究では、コンテナ技術を用いてクラウド SLAM 処理を構築し、コンテナの起動・停止・再起動を自動的に管理可能なクラウド実行基盤を前提とした設計を行う。

このような設計により、クラウド側の処理ノードが一時的に異常終了した場合でも、処理の自動復旧が可能となり、ロボットから見た自己位置推定サービスの可用性向上が期待できる。

本研究では、通信品質に応じたローカル構成とクラウド構成の切り替えに加え、クラウド内部における処理基盤の安定性も考慮することで、より実運用を想定した自律移動ロボットシステムの実現を目指す。

第4章

実装

本章では、設計章で述べたシステム設計および切り替え機構の設計思想に基づき、実際に構築した自律移動ロボットシステムの実装内容について詳述する。

本研究の特徴は、通信品質の変動に応じてローカル構成とクラウド構成を動的に切り替える点にあり、これを実現するためには、通信監視、切り替え制御、SLAM処理といった複数の機能を協調的に動作させる必要がある。そのため、本章では、個々の機能の実装方法だけでなく、それらがどのように連携してシステム全体として動作するかに着目して説明を行う。

まず、システム全体の実装構成を示し、ローカル構成およびクラウド構成がどのように分離・統合されているかを説明する。次に、通信品質指標の取得方法および切り替え機構の具体的な実装方針について述べる。さらに、ローカル構成およびクラウド構成それぞれの実装内容を示し、特にクラウド構成においては、k3sを用いたクラウド実行基盤の構築と自動復旧機構について詳述する。

本章で示す実装は、単に設計を実現した一例にとどまらず、通信変動が発生する実環境においても安定して動作可能な自律移動ロボットシステムの構築を目的としている。これらの実装の有効性については、次章以降の評価章において、実験結果を通じて検証する。

4.1 システム実装

4.1.1 実装全体の構成

本研究で構築した自律移動ロボットシステムは、設計章で述べた設計に基づき、ローカル構成およびクラウド構成の2つの処理構成を切り替えて利用可能な形で実装した。

本システムは、ロボット上およびクラウド上で動作する複数の処理を独立したノードとして実装し、それらを通信機構によって連携させる構成とした。各ノードは明確な役割分担を持ち、センサデータ取得、通信状態監視、自己位置推定および環境地図生成、ならびに処理構成の制御といった機能を担っている。

ローカル構成およびクラウド構成は、自己位置推定および環境地図生成という処理目的は共通であるが、処理を実行する計算資源の配置が異なる点に特徴がある。この差異を吸収するため、処理構成の切り替えは個々の処理内容を変更するのではなく、ノードの起動・停止および通信経路の切り替えによって実現する方式を採用した。

この構成により、上位の制御処理やナビゲーション処理は、現在どの構成が選択されているかを意識することなく動作可能となっている。

4.1.2 ローカル構成とクラウド構成の分離実装

ローカル構成とクラウド構成は、切り替え時の影響を最小限に抑えるため、同一の入出力インターフェースを持つように実装した。

具体的には、自己位置推定結果および環境地図の出力形式を統一し、処理構成の変更によって上位処理の接続先が変化しない設計とした。

ローカル構成では、ロボット上で完結する処理ノードのみを起動し、外部との通信を必要としない形で動作する。これにより、通信品質が低下した状況においても、ロボットは自己位置推定を継続可能である。

一方、クラウド構成では、ロボットから送信されたセンサデータを外部計算資源上の処理ノードが受信し、処理結果をロボット側へ返送する構成とした。ロボット側では、処理結果の受信および利用に専念することで、計算負荷の軽減を図っている。

このように、処理内容そのものではなく処理の配置を切り替える実装とすることで、設計章で述べた柔軟な処理構成の切り替えを可能としている。

4.2 切り替え機構の実装方針

4.2.1 一時的変動を考慮した切り替え制御

切り替え機構の実装においては、通信状態の一時的な変動によって不必要的構成切り替えが発生しないよう配慮した。

通信品質は時間とともに変動するため、瞬間的な通信遅延や通信失敗を直接的な切り替え条件として用いると、頻繁な構成切り替えが発生する恐れがある。これを避けるため、一定時間の通信状態を観測し、その傾向に基づいて切り替え判断を行う方式とした。

この方式により、短時間の通信揺らぎに対しては構成を維持しつつ、通信品質の悪化が継続している場合にのみ処理構成を変更することが可能となる。

4.2.2 切り替え状態管理の実装

処理構成の切り替えは、専用の制御ノードによって一元的に管理される。

本ノードでは、現在選択されている処理構成を内部状態として保持し、通信品質評価結果に応じて状態遷移を行う。同一の条件下で同一構成への切り替え命令が繰り返し発行されることを防ぐことで、システム全体の安定性を確保している。

また、構成切り替え時には、関連する処理ノードの起動・停止を順序立てて行うことで、処理の中止や競合が発生しないよう配慮した。

4.3 通信品質指標に基づく切り替え実装

4.3.1 通信品質指標の取得方法

通信品質の取得は、ロボット側で動作する通信監視ノードによって行う。

本ノードは、ロボットとクラウド間の通信状態を一定周期で計測し、通信遅延および通信成功率を取得する。取得したデータは、そのまま切り替え判断に用いるのではなく、一定時間分蓄積した後、統計的に処理される。

この処理により、瞬間的な異常値の影響を低減し、通信状態をより安定的に評価することが可能となる。

4.3.2 切り替え判定周期と応答性の関係

処理構成切り替えにおいては、通信品質評価を行う周期の設定が重要である。判定周期が短すぎる場合、通信品質の一時的な揺らぎに過剰に反応し、不必要的切り替えが発生する恐れがある。

一方で、判定周期が長すぎる場合、通信品質が著しく悪化しているにもかかわらず、切り替えが遅延する可能性がある。このような遅延は、クラウド構成における処理結果の遅れや欠落を招き、ロボットの挙動に悪影響を

及ぼす。

本研究では、これらのトレードオフを考慮し、一定時間分の通信品質データを蓄積した上で切り替え判断を行う方式を採用した。この設計により、応答性と安定性の両立を図っている。

4.3.3 静的指標としてのパケットロス率の実装

パケットロス率は、一定期間における通信成功率を表す指標として実装した。

通信監視ノードでは、観測区間内の通信試行回数および成功回数を記録し、それらの比率からパケットロス率を算出する。この値は、短時間で大きく変動する指標ではないため、通信環境全体の安定性を表す静的な指標として扱う。

本研究では、このパケットロス率をクラウド構成の利用可否を判断する際の補助的な指標として用い、通信経路そのものがクラウド処理に適しているかを評価する。

4.3.4 動的指標としての RTT 評価実装

RTT は、ロボットとクラウド間の通信遅延を表す指標として取得する。

本研究では、RTT の瞬間値ではなく時間的な変動傾向を評価対象とするため、一定時間分の RTT を蓄積し、その推移を基に通信状態を判定する方式を採用した。

これにより、一過性の遅延増加では切り替えを行わず、通信遅延が継続的に増加している場合のみローカル構成への移行が行われる。この実装は、設計章で述べた動的指標を用いた切り替え思想を具体的に反映したものである。

4.4 ローカル構成の実装

4.4.1 ローカル構成の動作概要

ローカル構成は、通信品質の低下が検出された場合においても、ロボットが自律的に動作を継続できるよう実装した。

本構成では、センサデータの取得から自己位置推定および環境地図生成、それらの結果の利用までをすべてロボット上で完結させる。

4.4.2 ローカル構成への切り替え処理

通信監視結果に基づき、通信遅延の増加が一定時間継続していると判断された場合、制御ノードはローカル構成への切り替えを実行する。

切り替え時には、クラウド通信を伴う処理ノードを停止し、ローカル構成に必要な処理ノードを起動することで、処理の連続性を確保する。

4.5 クラウド構成の実装

4.5.1 クラウド構成の動作概要

クラウド構成は、通信環境が安定している状況において、外部計算資源を活用することを目的として実装した。

ロボット側では、必要最小限の処理のみを行い、計算負荷の高い処理はクラウド側の処理ノードに委ねる構成とした。

4.5.2 クラウド構成の維持制御

動的 RTT の評価結果を用いることで、通信遅延が一時的に増加した場合でも、即座にローカル構成へ切り替えることなくクラウド構成を維持する。

これにより、高精度な処理を継続しつつ、不要な構成切り替えを抑制するという設計章で述べた方針を実装として実現している。

4.5.3 k3s を用いたクラウド基盤の構築

本研究では、クラウド SLAM 処理を実行する基盤として、軽量 Kubernetes 実装である k3s を用いてクラウド環境を構築した。

k3s は、通常の Kubernetes と比較して構成が簡素であり、限られた計算資源でも動作可能であることから、研究用途やエッジ環境に適している。この特性は、実験環境においてクラウド SLAM 処理を安定して運用する上で有効である。

クラウド SLAM 処理は、Docker コンテナとして実装され、k3s 上に Pod としてデプロイされる。これにより、SLAM 処理は特定の物理マシンに依存することなく、クラスタを構成するノード上でサービスとして実行される。

k3s が提供する Kubernetes の機能により、SLAM 処理を実行する Pod が異常終了した場合でも、Pod は自動的に再起動される。この自動復旧機構により、クラウド側の処理停止時間を最小限に抑え、ロボットとの通信を再開することが可能となる。

このように、k3s を用いたクラウド基盤上で SLAM 処理を運用することで、クラウド構成においても高い可用性と運用安定性を確保している。

4.5.4 k3s とフル Kubernetes の比較

クラウド基盤の構築にあたり、本研究ではフル機能の Kubernetes ではなく、軽量実装である k3s を採用した。フル Kubernetes は高機能である一方、構成や運用が複雑であり、研究用途においては過剰な場合がある。

k3s は、不要な機能を削減することで低リソース環境でも動作可能であり、セットアップが容易である。この特性は、限られた計算資源を用いた研究環境において大きな利点となる。

また、k3s は Kubernetes と互換性を持つため、本研究で構築したクラウド SLAM 基盤は、将来的にフル Kubernetes 環境へ移行することも可能である。この点は、研究成果の発展性という観点から重要である。

4.5.5 ROS 2 ノード分割設計の意図

本研究では、システム全体を单一の巨大なノードとして実装するのではなく、機能ごとに独立した ROS 2 ノードとして分割する設計を採用した。この設計は、処理構成の切り替えを前提とする本研究の目的に強く適合している。

SLAM 処理、通信品質監視、切り替え制御といった機能は、それぞれ役割や処理周期が異なるため、单一ノード内で統合すると実装が複雑化し、保守性や拡張性の低下を招く恐れがある。特に、通信品質に基づいて処理構成を切り替える場合、通信監視処理と SLAM 処理を明確に分離することが重要である。

ノード分割設計を採用することで、各機能は独立して開発・検証が可能となり、不具合発生時の原因特定が容易になる。また、将来的に通信品質評価指標を変更する場合や、異なる SLAM 手法を導入する場合においても、該当ノードのみを差し替えることで対応可能となる。

このような設計は、研究用途における試行錯誤の容易さだけでなく、実運用システムとしての再利用性や拡張性の観点からも有効である。

4.5.6 处理構成切り替えが上位処理に与える影響

本研究では、処理構成の切り替えがナビゲーションや制御といった上位処理に影響を与えないことを重視して実装を行った。

ローカル構成とクラウド構成では、自己位置推定および地図生成の処理場所が異なるが、それらの出力形式は共通化されている。これにより、上位処理は、現在どの構成が選択されているかを意識することなく、常に同一の情報を利用できる。

この抽象化は、処理構成切り替えを安全に行う上で重要な役割を果たす。もし構成ごとに出力形式が異なれば、切り替えのたびに上位処理の再設定や再起動が必要となり、システム全体の安定性を損なう可能性がある。

本研究の実装では、処理構成の違いを下位層に閉じ込めてることで、切り替えによる影響範囲を最小限に抑えている。この設計方針は、複雑なシステムを安定して運用するための基本的な工学的原則に基づくものである。

4.5.7 クラウド基盤の自動復旧機構と切り替え制御の関係

クラウド構成における Pod の自動復旧は、通信品質に基づく処理構成切り替え機構と相補的な関係にある。

通信品質が良好であるにもかかわらずクラウド処理が停止した場合、k3s による自動復旧によりクラウド構成を維持することが可能である。一方で、通信品質そのものが悪化した場合には、自動復旧では問題を解決できないため、ローカル構成への切り替えが必要となる。

このように、本研究では、通信品質に基づく構成切り替えと、クラウド基盤内部の自動復旧機構を併用することで、多様な障害要因に対応可能な自律移動ロボットシステムを実現している。

第5章

実験

本章では、前章までに述べた設計および実装に基づき構築した自律移動ロボットシステムについて、通信品質が変動する環境下における動作特性を実験的に評価する。

本研究では、ローカル構成とクラウド構成を通信品質に応じて切り替える機構を提案しており、その有効性を検証するため、以下の三つの観点から実験を行う。

まず、通信品質が時間的に変動する状況において、処理構成が意図した通りに切り替わるかを検証する。次に、切り替え判断に用いる動的 RTT 評価手法が、不必要的切り替え抑制に有効であるかを検証する。最後に、クラウド SLAM 基盤として採用した k3s による Kubernetes 環境において、障害発生時の自動復旧性能を評価する。

5.1 実験環境

項目名	データ名	結果
資料 1	こんなデータ	1.234
資料 2	あんなデータ	12.345
資料 3	こういったデータも	2.345

本研究で用いた実験環境を以下に示す。

ロボット側には、LiDAR を搭載した自律移動ロボットを用い、センサデータの取得およびローカル SLAM 処理を行う。ロボット上では、ROS 2 を用いて各種ノードを動作させ、通信品質の計測および処理構成の切り替え制御を実装した。

クラウド側には、複数ノード構成の小規模クラスタを構築し、軽量 Kubernetes ディストリビューションである k3s を用いてクラウド SLAM 基盤を構成した。SLAM 処理はコンテナ化され、Pod として管理されている。

ロボットとクラウド間は、無線ネットワークを介して接続されており、通信遅延およびパケットロスが時間的に変動する環境を想定している。

5.2 通信品質変動下における切り替え動作の検証

5.2.1 実験目的

本実験の目的は、通信品質が時間的に変動する環境下において、提案システムがローカル構成およびクラウド構成を適切に切り替えられるかを検証することである。

設計章で述べた通り、通信品質の一時的な悪化に対して即座に構成切り替えを行うことは、システムの安定性低下につながる可能性がある。そのため、通信状態の継続的な変化を捉え、必要な場合のみ構成切り替えが行われることが実環境において実現されているかを確認する。

5.2.2 実験方法

ロボットとクラウド間の通信環境において、通信遅延が時間的に変動する状況を作り出し、その際の RTT および処理構成の状態を記録した。

通信品質は、ロボット側の通信監視ノードによって一定周期で計測され、RTT およびパケットロス率が取得される。これらの値は時系列データとして保存し、後処理によって解析を行った。

実験中、ロボットは連続的に走行を行い、SLAM 処理を継続した状態で構成切り替えが発生するかを観測した。

5.2.3 実験結果

通信品質が時間的に変動する状況における処理構成の切り替え挙動を観測した結果、提案システムは通信状態の変化に応じてローカル構成およびクラウド構成を適切に切り替えて動作することが確認された。

実験開始直後、RTT が低く安定している状態では、クラウド構成が選択され、クラウド側で SLAM 処理が継続して実行された。この間、RTT は小さな変動を含みつつも、一定範囲内に収まっており、処理構成の切り替えは発生しなかった。

その後、通信遅延が段階的に増加する状況において、RTT の平均値および変動幅が継続的に大きくなる傾向が観測された。この状態が一定時間以上継続した時点で、制御ノードによりクラウド構成からローカル構成への切り替えが実行された。

切り替え後は、クラウドとの通信を伴う処理が停止され、ロボット上で完結するローカル構成の SLAM 処理が開始された。この際、自己位置推定および地図生成は途切れることなく継続されており、切り替えに伴う致命的な処理停止は確認されなかった。

さらに、通信品質が回復し、RTT が再び低下して安定した状態が一定時間継続した場合には、ローカル構成からクラウド構成への再切り替えが行われた。この切り替えも、短時間の通信回復では発生せず、安定状態が継続した場合のみ実行されることが確認された。

これらの結果から、提案システムは通信品質の時間的変化を考慮した構成切り替えを実現しており、通信遅延の増減に対して一貫した挙動を示すことが明らかとなった。

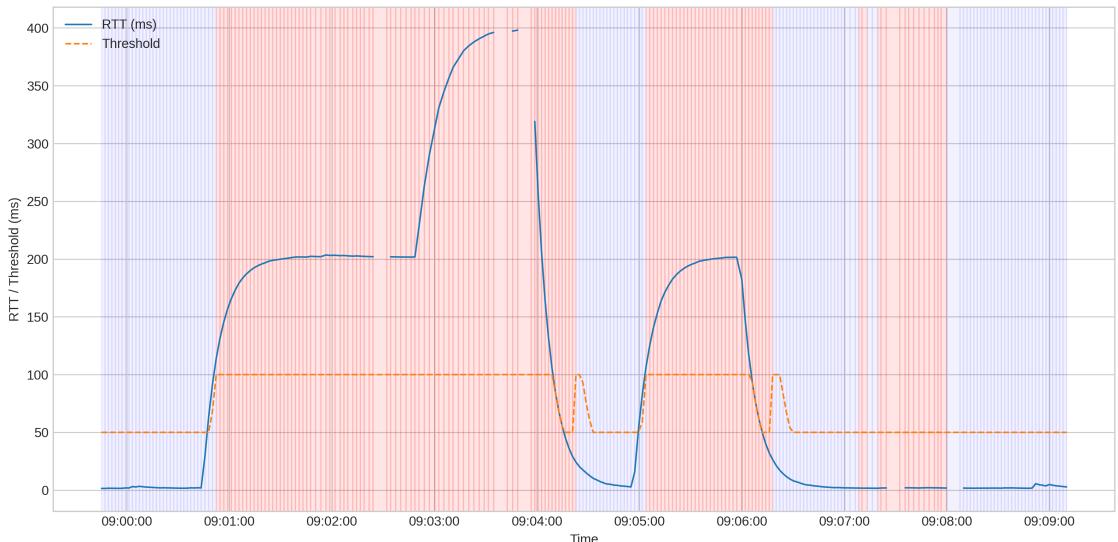


図 5.1 動的な閾値のグラフ

5.3 動的 RTT 評価の有効性検証

5.3.1 実験目的

本実験の目的は、切り替え判断において動的 RTT 評価を用いることの有効性を明らかにすることである。瞬間的な RTT 値のみを用いた場合、通信状態の一時的な変動によって不要な構成切り替えが発生する恐れがある。そこで、RTT を時間的に評価する手法が、切り替え安定性の向上に寄与するかを検証する。

5.3.2 実験方法

同一の通信環境下において、瞬間 RTT を用いた切り替え方式と、提案する動的 RTT 評価方式をそれぞれ適用し、切り替え挙動を比較した。

評価指標として、切り替え回数および切り替え発生タイミングを用い、それぞれの方式における挙動の違いを分析した。

5.3.3 実験結果

瞬間 RTT 評価方式および動的 RTT 評価方式を適用した場合の処理構成切り替え挙動を比較した結果、両者の間には明確な違いが確認された。

瞬間 RTT 評価方式を用いた場合、RTT が一時的に増加したタイミングにおいて、処理構成の切り替えが頻繁に発生した。特に、通信遅延が短時間だけ増大した場合であっても、閾値を超過すると即座にローカル構成への切り替えが行われる挙動が観測された。

この結果として、通信状態が完全には悪化していない状況でも、構成切り替えが発生し、再び通信状態が改善すると短時間でクラウド構成へ戻るといった往復的な切り替えが確認された。

一方、動的 RTT 評価方式を用いた場合、RTT の瞬間的な増加に対しては構成切り替えは行われず、RTT の平均値や変動傾向が一定時間以上悪化した場合にのみ切り替えが発生した。

その結果、切り替え回数は瞬間 RTT 評価方式と比較して明らかに少なくなり、処理構成が長時間にわたり安定して維持されることが確認された。

また、通信遅延が緩やかに増加する状況においては、動的 RTT 評価方式では切り替えタイミングが遅延し、通信状態の悪化が明確になった時点でのみ構成変更が行われる挙動が観測された。

これらの結果から、動的 RTT 評価方式は、通信状態の一時的な変動と継続的な悪化を区別できることが実験的に示された。

5.4 クラウド SLAM 基盤の自動復旧性能評価

5.4.1 実験目的

本実験の目的は、クラウド SLAM 基盤として採用した k3s による Kubernetes 環境において、障害発生時の自動復旧機能が有効に機能するかを検証することである。

クラウド構成では、外部計算資源に処理を依存するため、障害耐性の確保が重要となる。Kubernetes の自動復旧機能が、SLAM 基盤の信頼性向上に寄与するかを評価する。

5.4.2 実験方法

クラウド上で動作する SLAM 用 Pod を意図的に停止させ、その後の Pod 再生成および処理再開までの時間を計測した。

この操作を複数回繰り返し、復旧時間のばらつきおよび安定性を評価した。

5.4.3 実験結果

クラウド上で動作する SLAM 用 Pod を意図的に停止させた場合の挙動を観測した結果、k3s による Kubernetes 環境において、自動復旧機構が正常に機能することが確認された。

Pod 停止直後、クラウド側では該当 Pod が異常終了したことが検知され、一定時間後に新たな Pod が自動的に生成された。この再生成処理は人手による操作を必要とせず、Kubernetes の管理機構によって自律的に実行された。

新たに生成された Pod は、起動後に SLAM 処理を開始し、ロボット側から送信されるセンサデータの受信を再開した。この際、ロボット側では通信エラーが一時的に発生するものの、クラウド構成そのものが無効化されることはない。

同様の実験を複数回繰り返した結果、Pod 停止から処理再開までの時間は各試行において大きな差は見られず、一定範囲内に収まることが確認された。

また、Pod 再生成後には、クラウド SLAM 処理が継続して実行され、ロボット側で取得される自己位置推定結果および地図更新が再び正常に行われる事が観測された。

これらの結果から、k3s を用いたクラウド SLAM 基盤は、障害発生時においても自動的に処理を復旧させる能力を有しており、クラウド構成の継続運用が可能であることが実験的に示された。

5.5 実験結果のまとめ

本章では、通信品質が変動する環境下における提案システムの動作特性について、三つの観点から実験的に評価を行った。

まず、通信品質変動下における切り替え動作の検証では、RTT およびパケットロス率の評価結果に基づき、ローカル構成とクラウド構成が適切に切り替えられることを確認した。特に、通信品質の一時的な変動に対しては構成を維持し、通信状態の悪化が継続した場合にのみ切り替えが行われる挙動が観測された。

次に、動的 RTT 評価の有効性検証では、瞬間的な RTT 値を用いた切り替え方式と比較することで、提案する動的 RTT 評価方式が不要な構成切り替えの抑制に有効であることを示した。この結果は、通信環境が時間的に変動する実環境において、処理構成の安定性を確保する上で重要な知見である。

さらに、クラウド SLAM 基盤の自動復旧性能評価では、k3s を用いた Kubernetes 環境において、SLAM 処理 Pod が異常終了した場合でも、自動的に処理が復旧されることを確認した。これにより、クラウド構成における処理継続性が確保され、ロボットシステム全体の信頼性向上につながることが示された。

これらの実験結果から、通信品質に応じた処理構成切り替え機構と、クラウド基盤における自動復旧機構を組み合わせることで、通信障害およびクラウド側障害の双方に対して柔軟に対応可能な自律移動ロボットシステムを構築できる可能性が示唆された。

第6章

考察

本章では、第5章で示した実験結果をもとに、提案システムの特性、妥当性、および実運用における意義について、より詳細かつ多角的な観点から考察を行う。単なる結果の確認にとどまらず、その要因分析、設計上の意味、および今後の発展可能性について議論することで、本研究の位置づけを明確にする。

6.1 通信品質に応じた処理構成切り替えの有効性

第5章の実験結果より、通信品質が変動する環境下においても、ローカルSLAMとクラウドSLAMを適切に切り替えることで、自己位置推定処理を継続可能であることが確認された。

この結果は、通信品質が常に良好であることを前提としないロボットシステム設計の有効性を示している。実環境においては、無線通信の遅延や瞬断は避けられず、これらを例外的事象として扱う設計は現実的ではない。

本研究で提案した切り替え方式は、通信品質の低下を「異常」ではなく「起こりうる状態の一つ」として扱い、その状態に応じて処理構成を変化させる点に特徴がある。この考え方は、自律移動ロボットの実運用を強く意識したものであり、長時間運用を前提とするシステムにおいて特に重要である。

また、処理構成切り替えが行われた際にも、ロボットが完全に機能停止することなく、最低限の自己位置推定機能を維持できた点は、安全性の観点からも評価できる。

6.2 切り替え頻度とシステム安定性の関係

処理構成切り替えは有効な手段である一方、その頻度が過剰になると、システムの安定性を損なう可能性がある。SLAM処理は内部状態を持つため、頻繁な切り替えは地図の不整合や自己位置推定の一時的不安定化を引き起こす恐れがある。

本研究の実験結果から、動的RTT評価方式を用いた場合には、瞬間的な通信品質低下に対して切り替えが抑制される傾向が確認された。この挙動は、切り替え頻度を適切な範囲に抑える上で重要な意味を持つ。

特に、短時間の通信遅延に対して即座に処理構成を変更しない点は、実環境における通信特性を考慮した現実的な挙動であるといえる。

6.3 動的RTT評価方式の設計思想

動的RTT評価方式は、单一時点の数値に依存しないという点で、通信品質評価の信頼性向上に寄与している。RTTはネットワーク混雑や外乱の影響を受けやすく、瞬間値のみを用いた評価は誤判定を引き起こす可能性が高い。

本研究では、一定期間におけるRTTの推移を考慮することで、通信品質の傾向を把握する方式を採用した。この設計は、通信状態を時間的文脈の中で評価するという点で、より人間の判断に近いアプローチであると考えられる。

また、この方式は閾値設定の柔軟性という観点からも有効であり、異なる環境に対しても比較的容易に適用可能である。

6.4 k3s を用いたクラウド SLAM 基盤の信頼性

クラウド SLAM 基盤に k3s を採用したことにより、Kubernetes が持つ自動復旧機能を比較的低い運用コストで実現できた。Pod の異常終了時に自動的に再生成が行われることは、クラウド処理を前提とするシステムにおいて極めて重要な特性である。

この自動復旧機構により、クラウド SLAM 処理が一時的に停止した場合でも、人手による介入なしに処理が再開される。これは、ロボットシステムの可用性向上に直接的に寄与している。

さらに、コンテナ化による環境依存性の低減は、将来的なシステム拡張や再利用性の観点からも大きな利点である。

6.5 実運用を想定したシステム全体の評価

本研究で提案したシステムは、研究室内の実験環境だけでなく、実運用を想定した構成となっている。通信品質の変動、計算資源の制約、クラウド側障害といった要素を統合的に考慮している点に特徴がある。

このような設計思想は、今後の自律移動ロボットシステムにおいて重要性を増していくと考えられる。

6.6 既存研究との比較

自律移動ロボットにおける自己位置推定および環境地図生成（SLAM）は、これまでに数多くの研究が行われてきた。特に、計算機性能の向上や通信技術の発展に伴い、クラウド計算資源を活用したクラウド SLAM に関する研究も近年注目を集めている。

従来の SLAM 研究の多くは、ロボット単体で処理を完結させることを前提としており、計算資源の制約下で高精度な自己位置推定を実現することに主眼が置かれてきた。代表的な手法として、粒子フィルタを用いた Monte Carlo Localization、グラフ最適化に基づく Graph-based SLAM、および LiDAR を用いた scan matching 型 SLAM などが挙げられる。

一方、クラウド SLAM に関する既存研究では、計算負荷の高い最適化処理や地図統合処理をクラウド側で実行することで、ロボット側の計算負荷を軽減することが目的とされる場合が多い。これらの研究では、高帯域かつ低遅延な通信環境を前提としていることが多く、通信品質の変動に対する対策は十分に議論されていない場合が多い。

また、一部の研究では、通信遅延を考慮した SLAM 構成が提案されているものの、通信品質が悪化した場合の処理継続性や、処理構成の動的切り替えに関する実装・評価まで踏み込んだ例は少ない。

これに対し、本研究の特徴は、クラウド SLAM の導入を前提としつつも、通信品質の変動を不可避なものとして扱い、ローカル構成とクラウド構成を動的に切り替える点にある。特に、通信状態の瞬間値ではなく、時間的な変動傾向を考慮した動的 RTT 評価方式を切り替え判断に用いている点は、既存研究との差異として挙げられる。

さらに、本研究では、クラウド SLAM 基盤として k3s を用いた Kubernetes 環境を構築し、Pod の自動復旧機構を活用することで、クラウド側障害に対する耐性も考慮している。これは、単に SLAM アルゴリズムの性能評価にとどまらず、実運用を想定したシステム全体の信頼性に踏み込んだ点で、既存研究に対する発展的な位置づけにあるといえる。

以上より、本研究は、従来の SLAM 研究およびクラウド SLAM 研究を基盤としつつ、通信品質変動およびクラウド基盤障害という実環境特有の課題に焦点を当てた点に新規性と実用的意義がある。

6.7 実運用環境を想定した評価結果の解釈

本研究の実験は、研究室環境を中心とした限定的な条件下で実施されたものであるが、通信品質が時間的に変動する状況は実運用環境においても頻繁に発生すると考えられる。例えば、倉庫や工場内においては、無線アクセスポイントとの距離変化や遮蔽物の影響により、通信遅延やパケットロスが断続的に発生する可能性がある。

このような環境において、通信品質の瞬間値のみを基準とした制御は、過剰な構成切り替えを引き起こす恐れがある。本研究の結果から、RTTを時間的に評価することで、通信状態の一時的な揺らぎと継続的な悪化を区別できることが示唆された。

6.8 設計選択に関する考察

本研究では、通信品質評価指標として RTT とパケットロス率を採用した。これらは取得が容易であり、特別な通信機器や複雑な推定処理を必要としない点で実装上の利点がある。一方で、帯域使用率やジッタなど、他の通信指標を併用する余地も考えられる。

しかし、指標を増やすことは設計の複雑化につながるため、本研究ではあえて基本的な指標に絞ることで、実装の簡潔性と再現性を重視した。この設計判断は、研究用途のみならず、実運用システムへの展開を考慮した場合にも妥当であると考えられる。

6.9 リアルタイム性と通信依存性のトレードオフ

クラウド SLAM は、計算資源を外部に委ねることで高精度な処理を実現できる一方、通信遅延の影響を強く受ける。この点は、リアルタイム性が求められる自律移動ロボットにおいて本質的な課題である。

一方、ローカル構成では、通信に依存しない即時性が確保されるが、計算資源の制約により処理精度や更新頻度に限界が生じる。本研究は、これら二つの構成を対立するものとしてではなく、相補的なものとして捉えている。

処理構成を動的に切り替えることで、リアルタイム性と高精度処理のバランスを状況に応じて調整できる点は、本研究の重要な成果である。

6.10 オンボード計算資源制約の再評価

自律移動ロボットに搭載される計算機は、サイズ、消費電力、発熱といった制約を受ける。これらの制約は、長時間運用や安全性を考慮する上で無視できない要素である。

SLAM 処理は、センサデータ量が多く、計算負荷が高いため、オンボード計算機のみで高精度処理を行うことは現実的に困難な場合が多い。本研究の結果は、このような制約下において、クラウド計算資源を活用する意義を改めて示すものである。

第7章

終わりに

本研究では、通信品質が変動する環境下においても安定した自己位置推定を実現することを目的として、ローカル SLAM とクラウド SLAM を切り替える自律移動ロボットシステムを提案した。

7.1 研究全体の総括

本研究は、通信品質が変動するという現実的な条件を前提として、自律移動ロボットにおける自己位置推定処理の安定化を目指したものである。

設計、実装、実験、考察を通じて、単一の処理方式に依存しないシステム構成の有効性を示した点は、本研究の大きな成果である。

7.2 卒業研究としての到達点

本研究では、ロボットシステム、通信、クラウド基盤という複数分野にまたがる技術を統合し、実際に動作するシステムとして構築・評価を行った。

アルゴリズム提案にとどまらず、実装および運用を含めた検討を行った点は、卒業研究として十分な到達点であると考えられる。

7.3 本研究で得られた知見

本研究を通じて、通信品質を考慮した処理構成切り替えが、自律移動ロボットの安定運用に有効であることが示された。特に、動的 RTT 評価方式の導入により、切り替え判定の安定性が向上することが確認された。

また、k3s を用いたクラウド SLAM 基盤により、クラウド処理の信頼性を確保できることが示された。

7.4 研究の意義と位置づけ

本研究の意義は、クラウドロボティクスの利点と課題を踏まえた上で、実環境に適応可能なシステム構成を提示した点にある。通信品質変動という現実的な問題に対し、具体的な実装と評価を行った点は、今後の研究においても参考になると考えられる。

7.5 工学的意義と実用可能性

本研究で提案した処理構成切り替え方式は、特定の SLAM 手法に依存しないため、既存のロボットシステムにも適用可能である。この点は、研究成果を実システムへ応用する際の大きな利点である。

また、k3s を用いたクラウド SLAM 基盤の構築により、クラウド側の障害に対しても一定の耐性を持つシステ

ムを実現した。これは、クラウドロボティクスにおける「通信の不確実性」と「クラウド側障害」という二つの問題に同時に對処する試みであると言える。

7.6 本研究の課題

本研究では、通信品質に基づいて処理構成を切り替える自律移動ロボットシステムを提案し、一定の有効性を示した。しかしながら、本研究にはいくつかの限界が存在する。

第一に、通信品質評価指標として、RTT およびパケットロス率のみに着目している点が挙げられる。実際の通信環境においては、帯域幅の変動やジッタなど、他にも考慮すべき要素が存在する。これらを統合的に評価することで、より精緻な切り替え判断が可能になると考えられる。

第二に、実験環境が比較的限定的である点が挙げられる。本研究では、特定のネットワーク構成および実験環境において評価を行ったが、異なる通信方式や大規模環境における検証は行っていない。そのため、提案手法の汎用性については、今後さらなる検証が必要である。

第三に、処理構成切り替えが SLAM 内部状態に与える影響について、定量的な評価までは行っていない。地図の連続性や自己位置推定誤差の蓄積などについて、より詳細な解析を行うことで、切り替え方式の影響をより明確に評価できると考えられる。

最後に、本研究では單一ロボットを対象としているが、実運用では複数ロボットによる協調動作が想定される場合も多い。複数ロボット環境における通信品質変動やクラウド資源共有の影響については、本研究では扱っていない。これらの課題は、今後の重要な研究課題として位置づけられる。

7.7 今後の展望

今後の課題として、複数ロボットへの拡張、通信品質予測手法の導入、および長時間運用評価が挙げられる。これらを実現することで、より実用的なロボットシステムへと発展させることが可能である。

参考文献

- [1] L. Riazuelo, Javier Civera, J. M. M. Montiel: “C2TAM: A Cloud framework for Cooperative Tracking And Mapping”, Robotics and Autonomous Systems, Preprint submitted (2013.11.12).