卒業研究報告

AUTOSAR Adaptive Platform向け分散シミュレーション環境の構築

2020年2月

牧野 晃汰

概要

　〇〇・・・・・・・

目次

0.概要  
1.はじめに

* 1.1 研究背景
* 1.2 研究目的と概要
* 1.3 本論文の構成

2.（既存の）ラズパイマウスシミュレータ

* 2.1 シミュレータの構造

・GazeboやROS, AUTOSARなども含む

* 2.2 使用方法
* 2.3 シミュレータの課題

3.提案手法

* 3.1 求められる要件
* 3.2 要件への対応
* 3.3 使用方法

4.評価

* 4.1 評価環境
* 4.2 シミュレータ使用時の負荷評価

・CPU, GPU, メモリ使用率等

* 4.3 シミュレータの性能評価

・遅延やシミュレータ時間と実時間のずれ

5.おわりに

* 5.1 まとめ
* 5.2 今後の課題

図目次

表目次

第1章

はじめに

* 1. 研究背景

システムの開発において，開発したシステムが期待通りに動作するか，構築したシステムが仕様書通りの機能や性能要件を満たしているかどうか確認するために，システムテストが行われる．近年のシステムの複雑化に伴って，システムレベルのテストも複雑化しており，現在ソフトウェアの開発においては，システムレベルのテストのみで平均して全体の工数のうち約18％を占めている[1]．このテストの規模の増大により，テストのコストが増加するという問題がある．

このような問題を軽減するために，組込みアプリのシステムレベルのテストでは，シミュレーションが使われる機会が増えている．例えば，UGVのシステム開発の分野ではシミュレーション環境は欠かせないツールとなりつつある[2]．

しかし，シミュレータを動かすには専用の環境の構築を行わなければいけない．また，シミュレータの描画には多くの計算が必要で，動かすには一定以上のスペックのコンピュータが必要になる，これらが原因でシミュレータを利用する際に必要なコストが大きくなっているという問題がある．

* 1. 研究目的と概要

　本研究では，既存のシミュレータをターゲットとして，1.1節で述べた，シミュレータを利用する際のコストの大きさを軽減することを目的とする．その目的を達成するために，満たすべき要件を以下のように定める（要件？言い回し検討すべき）．

* + スペックの低いコンピュータでもシミュレータの利用を可能にする

（スペックの低いを厳密に定めるべき？例えばラズパイでも動きが滑らかに描写できるなど）

* + コンピュータ１台のみでシミュレータの利用を可能にする

前者の条件を満たすために，シミュレーション環境の一部をクラウドに移行する．また，後者の条件を満たすために，ターゲットとするシミュレーション環境に必要なRaspberry Pi（以下，ラズパイ）を仮想環境で実現する．また，これらの実装に伴う，コンピュータへの負荷の変化とシミュレータへの影響の評価を行う．

* 1. 本論文の構成

　本論文は本章を含めて全５章で構成される．第2章でターゲットとするシミュレーション環境の構成や課題について述べる．第3章では，第2章で触れたシミュレーション環境の課題を解決するために必要な要件を挙げ，その要件を満たす提案手法について述べる．第4章では，第3章で提案した手法を用いたときのコンピュータへの負荷やシミュレータへの影響について述べる．第5章ではまとめと今後の課題について述べる．

第2章

ラズパイマウスシミュレータ

2.1 シミュレータの構造

　本研究で扱うシミュレータは，アールティ社製のRaspberry Pi Mouse（以下，ラズパイマウス）をシミュレートする．ラズパイマウスとは，メインボードにラズパイを使った左右独立二輪方式のプラットフォームロボットである[3]．このシミュレータで使用する機器を次に示す

* + PC
  + ラズパイ

PCはROSノードやGazeboなどシミュレータの中心部分の役割や，キーボードからの文字入力やシミュレート結果の画面表示などユーザインタフェースの役割を担う．ラズパイでは，PCがキーボードから受け取った情報をもとにラズパイマウスの進行方向を決めて，その情報をPCへ送っている．これらの機器は同一LAN内になければならない．

### 図2.1を作り直すのでそれに伴い以下の用語も要変更 ######

このシミュレータの構成は図2.1のようである．TerminalはLinuxのターミナルで，ラズパイへSSH接続を行う．これにより，ターミナル経由でキーボードへ入力した情報をラズパイに送信する．InputとControlはAUTOSAR APのアプリケーションである，Inputではキーボードから受け取った情報を取得し，Controlへ送信する．ここでのControlへの送信には，AUTOSARの機能を用いる．Controlでは，Inputから受け取った情報やシミュレータから受け取った情報をもとにラズパイの進行方向を決める．その情報はPC上のcontrol\_monitorに送られる．control\_monitorはROSのノードで，GazeboとAUTOSAR APのアプリケーションの橋渡しを行う．Controlから進行方向の情報の受信に加えて，Gazebo上のシミュレータの情報のControlへの送信も行う．そのデータのやり取りにもAUTOSAR APの機能を用いる．Gazeboはシミュレータの計算，結果の表示を行う．control\_monitorとのデータのやり取りには，ROSのPublish．Subscribe型の通信を利用している．この通信はGazeboの特徴であるROSとの連携機能が充実していることにより可能になっている．

##### DiagはAUTOSAR APの機能？ #####

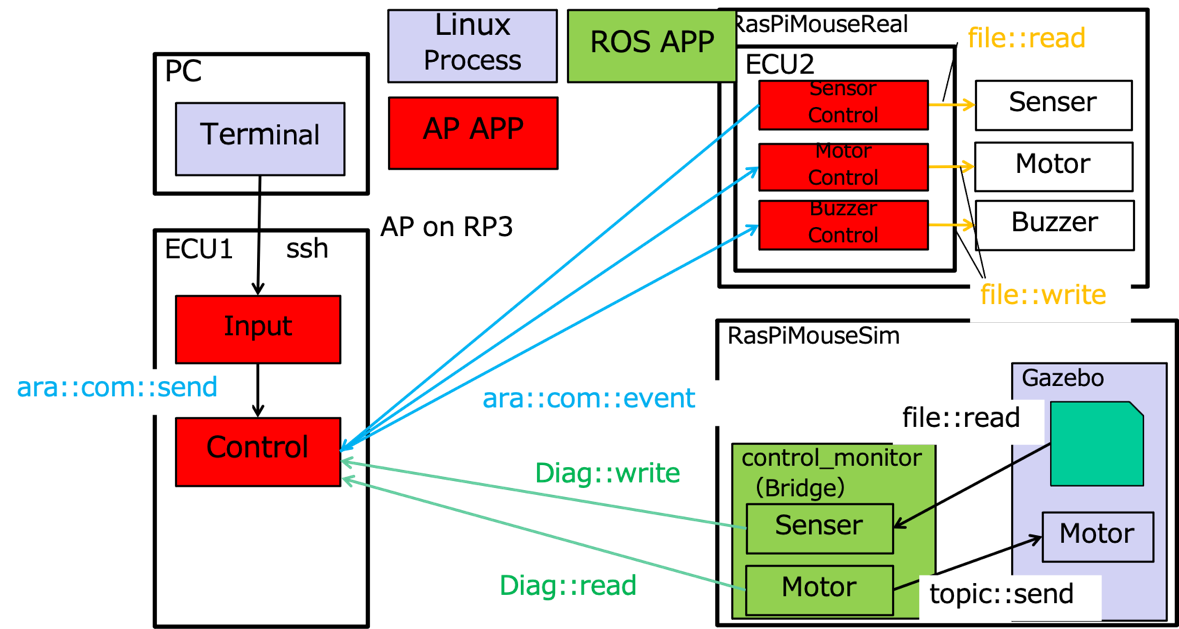


図2.1 シミュレータの構造図（仮，ちゃんと自分で作る）

2.X Gazebo

　　ターゲットとするシミュレータはGazebo[4]を使用する．Gazeboとは，Player/Stage projectsで開発された[6]，[7]，[8]オープンソースの3Dロボットシミュレータである． Gazeboでは，シミュレートする対象物はそれぞれ質量，速度，摩擦係数を初めとする多くのパラメータを保持しており，それにより現実世界の忠実な再現を可能としている．さらに．Gazeboには以下のような特徴がある[5]

* + 3次元レンダリングエンジンによる高品質なグラフィックス処理機能を有する
  + カメラ，レーザーレンジファインダ，力センサーなど，多様なセンサーをシミュレーションできる
  + プロセス間はTCP/IP接続されており，リモートでもシミュレーションを実行できる
  + ROSと同様にOSRFのサポートを受けており，ROSとの連携機能が充実している．

このような利点から，本シミュレータではGazeboが用いられている．

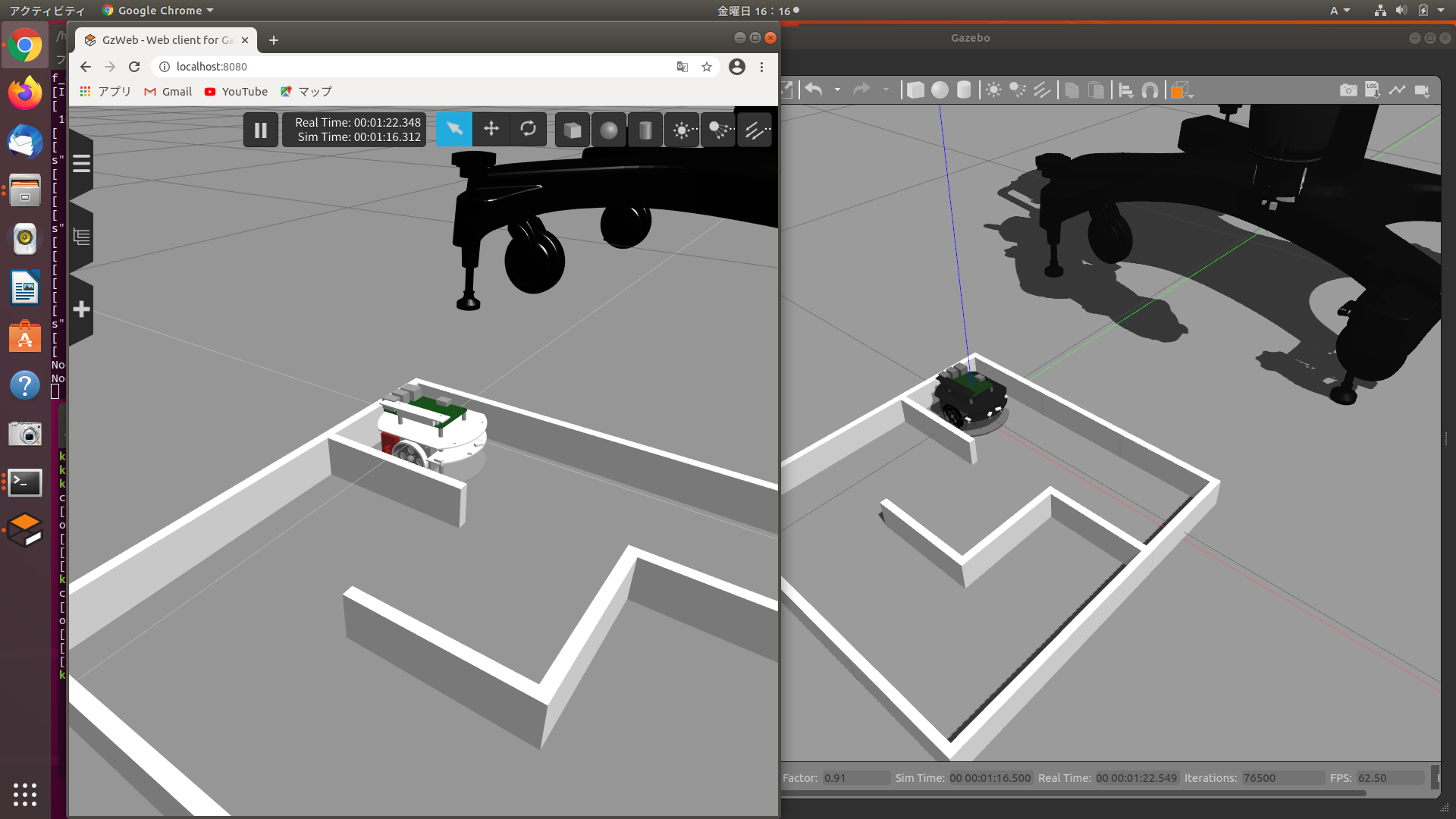


図2.X Gazebo使用時の画面（仮，後でもっとグラフィカルなものを撮る）

2.Y AUTOSAR AP

　AUTOSAR[10]，[9]，[11]とはAutomotive Open System Architectureの略称で，2003年に設立された車載ソフトウェアプラットフォームの仕様の標準化を進める団体である．AUTOSARが開発した標準化ソフトウェアプラットフォームには，Classic Platform（以下，CP）と，Adaptive Platform（以下，AP）の2種類がある．今回のシミュレータで使用されているのはAPである．APはCPよりも複雑な計算が必要なアプリケーション，またそれを動かす環境をターゲットに作られている． また，CPが既に実用段階にあるのに対し，APは開発中期段階である．

2.2 使用方法

　まず，環境の構築を行う．PC，ラズパイはそれぞれ以下のものを使用した．

* + PC　###調査後記入###
  + ラズパイ ###調査後記入###

PC

参考文献

###後で形式を整える，番号順もバラバラになっているので注意###

[1] ソフトウェア開発データ白書，p179

[2] paper3.pdf

[3] ラズパイマウスホームページ

[4] Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator

[5] ROS/Gazeboを用いたロボットシミュレーション

[6] [4]のAbstractを参照

[7] 〃

[8] 〃

[9] Achievements and exploitation of the AUTOSAR development partnership

[10] <https://www.autosar.org/>

[11] Achieving Determinism in Adaptive AUTOSAR