A-01

ROS に基づく自動運転用ソフトウェアと MATLAB/Simulink の統合開発フレームワーク

Integrated Development Framework for ROS-based Autonomous Driving System

徳永 翔太 †1　堀田 勇樹 †2　小田 裕弘 †3　安積 卓也 †1
Shota Tokunaga　Yuki Horita　Yasuhiro Oda　Takuya Azumi

1. はじめに

近年，自動運転技術は急速に進歩している [1]．自動運転車はカメラ，LIDAR，GNSS，高性能制御装置といった様々なハードウェアで構成されており，それらのシステムは，ROS（Robot Operating System）[2] を用いて開発されている [3]．ROS はロボットアプリケーション開発に適したオープンソースのミドルウェアフレームワークである．ROS は，個々のプロセスを表すノードと，ノードの入出力データを保持するトピックを用いたノード間通信により，分散処理が可能である．そのため，各ロボットのコンポーネントやアルゴリズムの再利用性が高い．加えて，ROS は様々なライブラリや視覚化ツールも提供しており，それらを用いることで開発効率が向上する．それゆえ，ROS はロボット開発において広く普及している．

ROS に基づく自動運転システムに Autoware [4] [5]がある．Autoware は自動運転のためのソフトウェアフレームワークであり，自動運転車のシミュレーション及び操作が可能である．Autoware では，Runtime Manager と呼ばれる GUI（Graphical User Interface）ツールを用いてノードの立ち上げや，データの読み込みを行うことができる．加えて，Autoware では実データに基づくシミュレーションも行うことができる．

一方で，自動車産業では，経路計画，画像処理，機械学習といった自動運転システムの設計に，MATLAB/Simulink [6] を用いている．MATLAB/Simulink には，モデリング，シミュレーション，解析をするための機能があり，これら機能により，作成されたモデルはすぐにその動作を検証できるため，モデルの評価及び改善を容易にする．モデルの改善を設計の早期の段階で繰り返すことにより，モデルの生産性及び質を向上させる．そのため，Autoware 開発においても MATLAB/Simulink が利用されている．

しかしながら，現在採用されているフレームワークでは，MATLAB/Simulink で作成されたモデルを Autoware で直接利用することはできない．MATLAB/Simulink で作成されたモデルを Autoware で利用するためには，そのモデルの C++ コードを作成し，Autoware に手動で組み込む必要がある．さらには，MATLAB/Simulink では仮想データによるシミュレーションしかできないため，Autoware に組み込みこんでも実データで正常に動作しない恐れがある．これらの問題を解消するために，Autoware と MATLAB/Simulink の統合開発を可能にするフレームワークを提案する（図１）．提案フレームワークは，ROS と MATLAB/Simulink のインタフェースを提供する Robotics System Toolbox [7]を用いることで，MATLAB/Simulink で作成されたモデルが Autoware と接続可能になる．そのため，作成されたモデルを，C++ コード生成して Autoware に組み込まず MATLAB/ Simulink コードのままで利用可能なため，開発効率が改善する．

本研究の主な貢献は次の通りである：

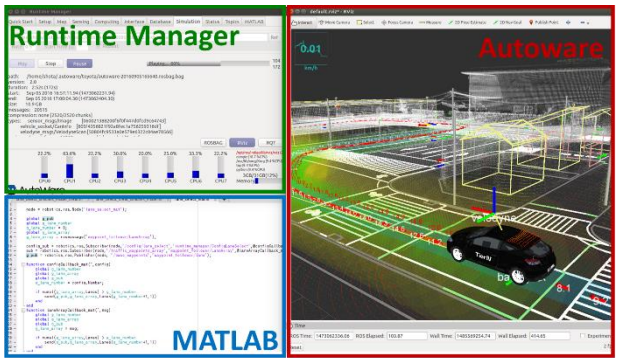

図 1　提案フレームワークを用いて Runtime Manager から MATLAB/Simulink を操作し，Autoware とコシミュレーションしている様子．

- ROS と MATLAB/Simulink におけるデータの転送時間及び処理能力を比較し，この手法の実用性を検証する．その結果から，MATLAB/Simulink で作成されたモデルはコシミュレーション環境において適用可能であることを示す．
- 提案フレームワークは，MATLAB テンプレートスクリプト及び Simulink テンプレートモデルを生成するため開発効率を改善する．テンプレートは，MATLAB/Simulink で Autoware のノードを作成する手助けをする．
- Runtime Manager を拡張し，MATLAB/Simulink 用の操作を可能にすることで，ユーザビリティを向上させる．　加えて，提案フレームワークが提供する他の機能(例えばテンプレート生成機能)も Runtime Manager 上から実行可能にする．

構成： ２章では，システムモデルを説明し，そのシステムモデルを理解するために必要な事前知識を与える．３章では，提案フレームワークの設計及び実装について説明する．４章では，提案フレームワークの実用性，効率，ユーザビリティについて評価する．５章では，提案手法を関連研究と比較する．６章では，結論を与え，今後の展望について述べる．

2. システムモデル

提案手法に基づくシステムモデルを図２に示す．Robotics System Toolbox によって提供された関数を用いてモデルを作成することで，コード生成をして Autoware に組み込むことなく MATLAB/Simulink で作成されたモデルが Autoware と連携可能になる．これにより，MATLAB/Simulink モデルに対しても，実データによるシミュレーションが可能なため，シミュレーションの質及び効率が向上する．加えて，このモデルは自動運転車を用いた実証実験

†1 大阪大学大学院 基礎工学研究科
†2 日立製作所
†3 日立オートモティブシステムズ

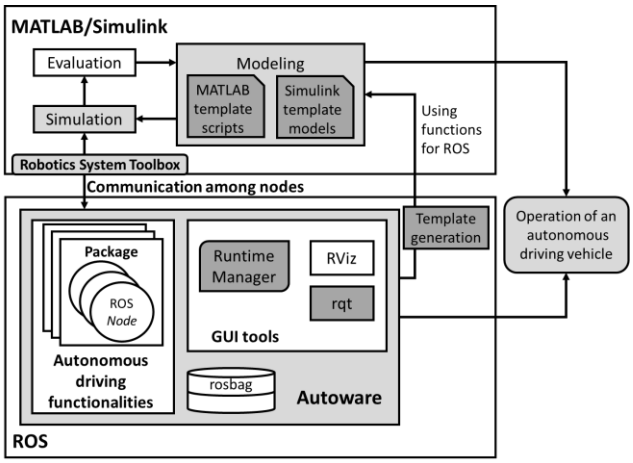

図 2 自動運転システムの開発における提案フレームワークのシステムモデル.

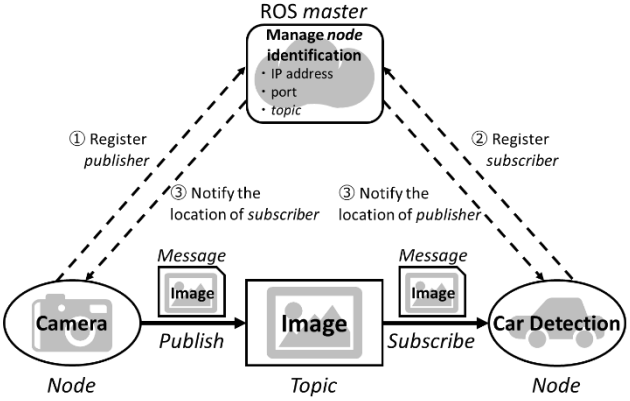

図 3 Publish/Subscribe モデルに基づいた通信の例.

でも利用可能である．さらに，提案フレームワークはテンプレート生成機能を提供するため，MATLAB/ Simulink でのモデリング効率も向上する．そして，このシミュレーションと実証実験の過程において，提案フレームワークは MATLAB/Simulink の操作を可能にするため，ユーザビリティを改善する．

このシステムモデルを理解するためには，いくつかの事前知識が必要となる．本章では，まず ROS の主な特徴について解説し，ROS の機能である RViz [8]，rqt [9]，rosbag [10] について説明する．そして，Autoware，MATLAB/ Simulink 及び Robotics System Toolbox について説明する．

2.1 Robot Operating System (ROS)

ROS は Linux 上で動作するオープンソースミドルウェアであり，ロボットアプリケーション開発に適した様々なツールやライブラリを提供する．ROS に基づくアプリケーションはノードと呼ばれる独立したプロセスで作られる．

ノード間の通信は主にメッセージを用いた Publish/ Subscribe モデルに基づいて行われる．メッセージとは C の構造体のような構造をしており，トピックは各ノードがメッセージを特定ために用いられる．各ノードは自身の位置（IP アドレスとポート番号）とそのトピックを ROS マスタに登録する．ROS マスタは登録されたトピックを確認し，各ノードに所望のノードの位置を知らせる．そして，ノードはトピックを通じてメッセージをやり取りする．図 3 の例では，まず，カメラノード（パブリッシャ）がその位置

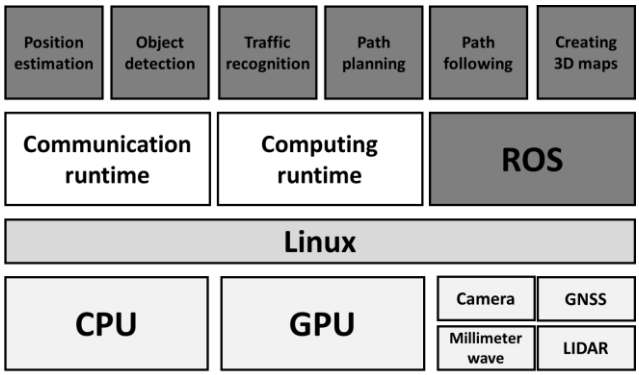

図 5 Autoware の構成

とイメージトピックを ROS マスタに登録する．次に，車検知ノード（サブスクライバ）がその位置と所望のトピック（イメージトピック）を ROS マスタに登録する．ここで，ROS マスタがトピックの一致を確認し，そのパブリッシャとサブスクライバに相手の位置を知らせる．そして，パブリッシャがサブスクライバにメッセージを送る．

この Publish/Subscribe モデルにおいて，ノードは ROS マスタの位置さえ知っていれば，ノード間通信を行うことができる．これにより，ROS では分散コンピューティングを構築することが可能となり，重たい処理を複数の機械に分散させることができる．

2.1.1 RViz

RViz は ROS 用の 3 次元視覚化ツールであり，GUI で操作可能である．RViz ではロボットやセンサに加え，カメラやレーダから得たデータのトピックの表示も行うことができる．プログラムの開発時では，RViz を用いたシミュレーションで事前に動作を確認できる．

2.1.2 rqt

rqt は Qt [11] に基づく ROS 用のソフトウェアフレームワークであり，プラグイン形式で GUI ツールの実装と開発が可能である．rqt のすべての GUI ツールはドッキング可能なウィンドウとして実行でき，複数の視覚化ウィンドウを同時に簡単に管理できる．

2.1.3 rosbag

rosbag パッケージはトピックの記録と再生を行うコマンドラインツールを提供する．rosbag は自動運転車を使って実証実験で取得したトピックのメッセージとそのメッセージの転送時間を記録できる．時間も記録しておくことで，メッセージを記録したときと同じ状況を再生できるため，シミュレーションの質を向上させ，問題解決に役に立つ．

2.2 Autoware

Autoware は，ROS に基づく自動運転システムの研究開発をサポートするために開発された，オープンソースソフトウェアである．Autoware には，自己位置推定，物体検出，信号認識，経路計画，経路追従，3D マップ作成などの一連の自動運転機能が備わっている（図４）．これらの機能は主に，ROS，通信ランタイム，コンピューティングランタイムの下で動作し，C++で書かれたノードを立ち上げることで使うことができる．これらのノードは，Runtime

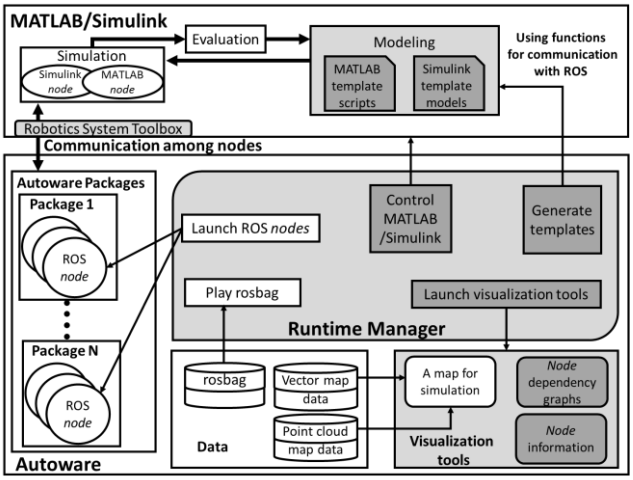

図 4 提案フレームワークによって提供される機能の概要.

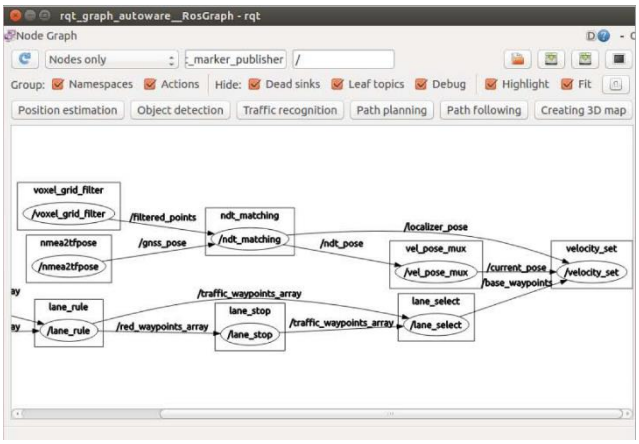

図 5 rqt_graph_autoware プラグインのスクリーンショット．経路計画のノード依存関係グラフを描画している．

Manager と呼ばれる Autoware の開発用の GUI ツールを用いて起動できる．Runtime Manager はファイルの読み書き，ノードの実行，rosbag の記録・再生を GUI で行うことができる．GUI 操作により，Autoware の開発者が ROS や自動運転システムに詳しくなくても簡単に利用できる．

2.3 MATLAB/Simulink

MATLAB/Simulink は自動車産業において，モデルの設計・開発に広く使用されており [12]，提供機能によりモデリングの効率と質を向上させる．そのため，Autoware のモデルを作成するときにも MATLAB/Simulink は利用されている．

MATLAB/Simulink で作成されたモデルが Autoware と通信するためには，Robotics System Toolbox を用いる必要がある．Robotics System Toolbox は ROS と MATLAB/Simulink のインタフェースとして働き，ROS に基づくロボットの開発に用いられている [13]．Robotics System Toolbox によって提供された関数を使ってモデルを作成することで，そのモデルを ROS マスタに登録でき，Publish/ Subscribe モデルに基づいた通信が可能となる．

3. 設計・実装

提案フレームワークによって提供される機能は Autoware と MATLAB/Simulink の統合開発を促進する．主な機能は次のようになる（図５）：

- MATLAB テンプレートスクリプトと Simulink テンプレートモデルの生成，及びテンプレート生成機能を支援する視覚化ツールの提供
- Runtime Manager 上で MATLAB/Simulink の操作，ノード情報の表示，及び提供された機能を利用可能にする

次に，この各機能の設計及び実装について述べていく．

3.1 テンプレート生成

MATLAB/Simulink を使って Autoware 用のノードを作成する際，ノード名，Publish/Subscribe するトピック，及びそのトピックのメッセージ型の情報が必要となる．これらの情報は，Autoware のソースコードを解析するか，ROS のコマンドを入力することによって得ることができるが，これらの解析作業は開発者，特に ROS に詳しくない開発者に取って負担になる．そこで，これらの必要不可欠な情報を含んだ MATLAB テンプレートスクリプトと Simulink テンプレートモデルを自動生成する機能を提供する．さらに，このテンプレート生成機能を補助する 2 つの視覚化ツールも作成した．一つは rqt_graph_autoware プラグインである (図６)．これは，rqt_graph [14] の機能に加え，Autoware の，自己位置推定，物体検出，信号認識，経路計画，経路追従，3D マップ作成といった機能ごとにノードの依存関係を描画する機能を持つ．もう一つのツールは，現在立ち上がっているノードのリストを描画し，そのリストから選択されたノードの情報を描画する機能である．これらの機能の設計及び実装方法について述べていく．

上記で述べたように，所望のノードのテンプレート生成には，そのノードの情報を得る必要がある．そこで，提案フレームワークでは，すべての Autoware のノードの情報を含んだ yaml ファイルを提供している．yaml ファイルとは，構造化データ及びオブジェクトを文字列にシリアライズするために使用されるデータファイルである．MATLAB テンプレートスクリプトまたは Simulink テンプレートモデルを生成する際，実行コードが yaml ファイルを読み込むことで，ノードの情報を取得する．実行コードのアルゴリズムは次のようになる：

1. 引数として生成したいノード名を与える
2. yaml ファイルを読み込み，引数と一致するノードの情報を取得する
3. 取得した情報に基づいてテンプレートを生成する

MATLAB テンプレートスクリプトと Simulink テンプレートモデルのどちらを生成する場合でも，このアルゴリズムに基づいている．開発者はこの機能によって生成されたテンプレートを利用して MATLAB/Simulink で Autoware 用のモデルを作成できる．

rqt_graph_autoware プラグインを実装するために，Autoware の機能ごとのノードの依存関係グラフを描画するための dot ファイルを作成した．さらに，rqt_graph_ autoware プラグインの GUI 設計には，Qt デザイナーを用いた．Qt デザイナーでボタンを作成し，そのボタンをクリックすることでグラフが表示されるように設定した．この機能により，開発者が Autoware の機能ごとに含まれたノードを視覚的に知ることができる．

ノード情報の表示には，rosnode コマンドラインツール [15] を利用した．このコマンドラインツールには，ノードの情報を表示するための rosnode list コマンドと rosnode

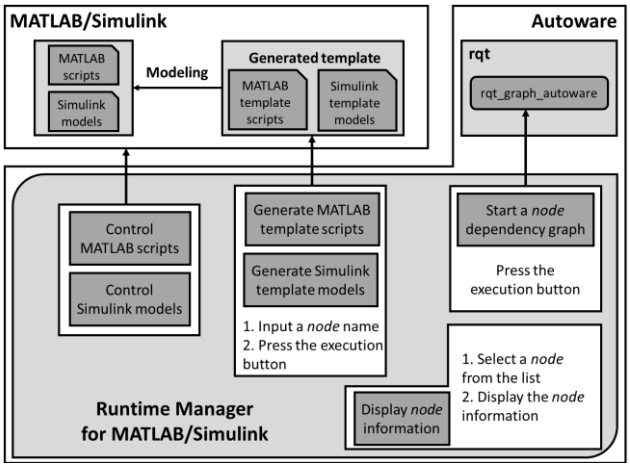

図 6 Runtime Manager に追加した機能.

info node_name コマンドがある．rosnode list コマンドは現在立ち上がっているノードのリストを表示し，rosnode info node_name コマンドは指定したノードによって Publish/Subscribe されているトピックの情報を表示する．そこで，Runtime Manager 上にこれらのコマンドの結果を表示することでノードの情報を取得できるようにした．この Runtime Manager 上への表示方法は，3.2 章で述べる．

3.2 MATLAB/Simulink 用の Runtime Manager

MATLAB/Simulink の操作には様々な方法がある．例えば，MATLAB を Linux 上で起動するためには，システムプロンプトで，matlab とコマンドを打つ．MATLAB/Simulink で作成されたモデルを実行するためには，MATLAB/Simulink 上の GUI から操作するか，実行用のコマンドを打つ．これらの方法は，Runtime Manager で自動運転システムを管理する Autoware の操作方法とは異なる．Runtime Manager では GUI 操作でノードの起動，データファイルのロード，rosbag の再生等を行う．これにより，MATLAB/Simulink を使う開発者が ROS や自動運転システムに詳しくなくても Autoware を容易に利用できる．しかしながら，現在のバージョンの Runtime Manager には MATLAB/Simulink 用の機能がないため，Autoware と MATLAB/Simulink を組み合わせて使用すると，操作方法が統一されていない．そこで，本研究では Runtime Manager に GUI を追加し，MATLAB/Simulink の操作と提案フレームワークで提供した機能を利用可能にした．これにより利用可能な機能は次のようである（図. 7）：

- MATLAB 及び Simulink の制御
- MATLAB テンプレートスクリプトと Simulink テンプレートモデルの自動生成
- ノード情報の表示

開発者はこれらの機能を Runtime Manager 内の他の機能と同様の方法で利用できる．この操作方法の統合により，MATLAB/Simulink の操作と提供機能の利用が容易になる．次に，この MATLAB/Simulink 用の Runtime Manager の設計及び実装方法を述べる．

Runtime Manager は wxPython ツールキット [16] を使って設計されている．そこで，マウス操作で GUI の配置を設計し wxPython のコードを出力する wxGlade [17]を使って追加機能用の GUI を設計した．ここで作成した GUI は各機能を実行するためのボタンとパネルである．

作成した GUI に機能を設定するために Runtime Manager の実行コードを修正した．実行コードでは wxGlade で生成

表 1 評価環境 .

CPU	Model number	Intel Core i7-6700K
	Cores	4
	Threads	8
	Frequency	4.00 GHz
Memory		32 GB
ROS		Indigo
MATLAB/Simulink		R2016b
OS		Ubuntu 14.04.5 LIT

されたファイルや yaml ファイルなどを含んだ様々なモジュールをインポートしている．この実行コード内で，yaml ファイルを読み込み，特定のボタンに特定の機能なら設定可能な関数が定義されていた．そこで，MATLAB/Simulink 用の yaml ファイルを作成し，この関数を用いて MATLAB 及び Simulink の起動ができるように各ボタンに設定した．

Runtime Manager で MATLAB スクリプトと Simulink モデルを実行するために，次のように複数の GUI の作成及び設定をした：

- ファイル選択用ダイアログを開くためのボタン
- 選択したファイルまでの絶対パスを表示するパネル
- パネルに表示されたファイルを実行するためのボタン

この実行ボタンは，選択されたファイルが MATLAB /Simulink ファイル（m ファイルまたは slx ファイル）であれば実行されるように設計されている．

次に，MATLAB テンプレートスクリプトと Simulink テンプレートを生成するために，ノード名を入力するためのパネルと，実行コードを実行するためのボタンを設定した．実行コードはパネルに入力された文字列を引数としている．

最後に，ノード情報の表示のために，2 つのパネルを設計した．一方のパネルには rosnode list コマンドの実行結果を表示し，そのリストからノードを選択することで，もう一方のパネルに rosnode info node_name の出力を表示した．

4. 評価

提案フレームワークの主な目的は実用性・効率・ユーザビリティの改善し開発効率を向上させることである．コシミュレーションや自動運転車を使った実証実験において，提案フレームワークの実用性を評価するために，ROS 内のデータ間通信時間と ROS と MATLAB/Simulink 間の通信時間を比較した．モデリング効率は生成されたテンプレートの量で評価した．ユーザビリティの評価では，Robotics System Toolbox のみを利用した場合と提案フレームワークを使った場合の開発環境を比較した．表 1 に，本評価実験に用いたソフトウェア・ハードウェア環境を示す.

4.1 実用性

MATLAB/Simulink は，Autoware の経路計画や画像処理のモデルを作成するのに用いられる．この作成したモデルを Autoware に組み込むことなく，MATLAB/Simulink モデルのまま利用可能なことを示せれば，開発効率を改善できる．この点での実用性を示すために，ROS と MATLAB/Simulink を次のように比較する：

- メッセージを ROS または MATLAB を経由して転送した場合の転送時間とデータサイズのトレードオフ
- ROS と MATLAB に同一の処理を行わせた場合の処理能力

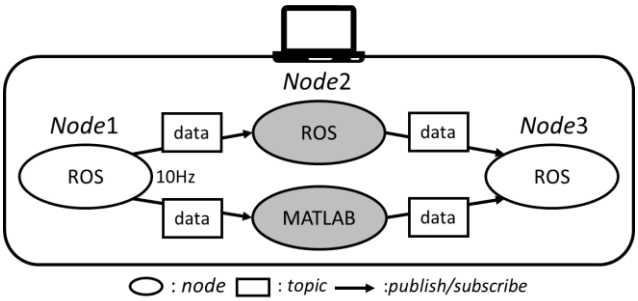

図 7 転送時間の計測.

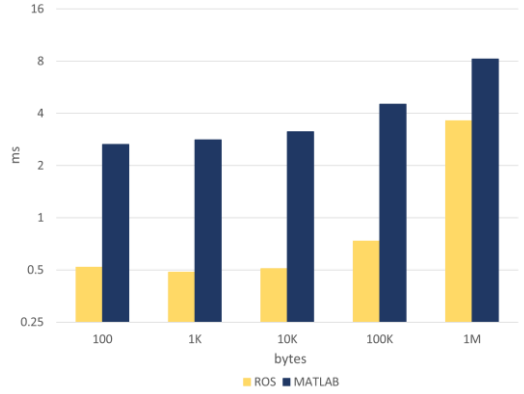

図 9 ROS または MATLAB/Simulink を経由した場合のメッセージのデータサイズに応じた平均転送時間.

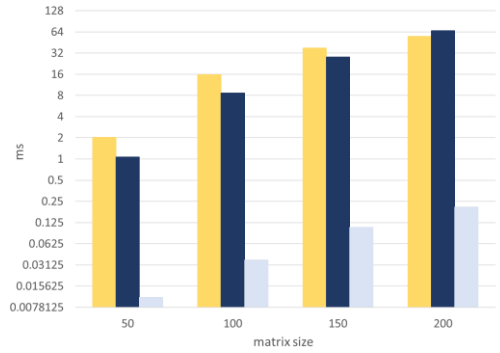

図 10 ROS または MATLAB/Simulink における行列サイズに応じた平均処理時間.

図 8 に示すように，転送時間はノード 1 がメッセージをパブリッシュしてからノード 2 を経由し，ノード 3 がサブスクライブするまでの時間とする．処理能力は処理にかかる時間を測定する．本実験では，ROS と MATLAB は同一の機械で起動し，ノード 1 はメッセージを 10Hz でパブシッシュする．

4.1.1 転送時間

各トピック上のメッセージのデータサイズを 100，1K，10K，100K，1M バイトとして，ROS を経由した場合と MATLAB を経由した場合の転送時間を測定した．

図 9 は各メッセージサイズに応じた平均転送時間である．ROS と MATLAB ともにメッセージサイズが増加するにつれて転送時間が増加した．加えて，MATLAB を経由してデータを転送したほうが，より大きなオーバーヘッドがかかった．ここで，MATLAB/Simulink は経路計画といった扱うデータが小さな処理は実証実験で用いられるが，

表 2 MATLAB テンプレートスクリプトを用いることで削減される作業量

	MATLAB template scripts
Generated lines	(1) + α(2) + β((3) + 2(4))

(1): Defining node
(2): Defining publisher
(3): Defining subscriber
(4): Defining callback function
α: The number of publishers
β: The number of subscribers

表 3 Simulink テンプレートモデルを用いることで削減される作業量

	Simulink template models
Simulink blocks	α ((1) + (2) + (3)) + β ((4) + (5) + (6))
Settings	(i) + (α + β)((ii) + (iii) + (iv) + 2(v))

(1): Placing Publisher
(2): Placing Message
(3): Placing Bus Assignment
(4): Placing Subscriber
(5): Placing Bus Selector
(6): Placing Terminal
(i): Defining model name
(ii): Configuring message name
(iii): Configuring topic name
(iv): Configuring topic source
(v): Connecting blocks
α : The number of publishers
β : The number of subscribers

Autoware の最大周波数である 20Hz より十分に小さいため実用可能である．

4.1.2 処理能力

MATLAB/Simulink の処理能力を評価するために ROS と MATLAB における正方行列の乗算にかかった時間を計測した．MATLAB/Simulink の時間計算量に応じた処理能力と，提供されている関数の処理能力を測定するために，MATLAB スクリプトは，ROS コードと同様に行列サイズに応じた時間計算量になるように実装したものと，MATLAB/Simulink によって提供されている関数を用いて実装したものを用意した．図 10 は，各行列サイズの行列計算を行う ROS コード，MATLAB スクリプト，MATLAB/Simulink で提供されている関数を用いた MATLAB スクリプトの平均処理時間である．ROS と MATLAB は同様の方法でコードを記述した場合，処理時間はほとんど同じ結果となった．一方で，MATLAB/Simulink が提供する関数を用いて MATLAB スクリプトを実装した場合，他の 2 つより処理にかかった時間は圧倒的に短かった．これは，特に指定しなくても処理が複数スレッド・複数コアで実行されているからである．ここで，処理時間を転送時間と比較したとき，提供されている関数を用いて実装した MATLAB スクリプトの処理時間は十分に小さい．従って，提供されている関数を用いて MATLAB/Simulink でモデルを作成することで，画像処理といったデータサイズの大きなデータに対しても実用できる．

4.2 効率

提案フレームワークでは，MATLAB テンプレートスクリプトと Simulink テンプレートモデルの生成が可能である．このテンプレートは，開発者が MATLAB/Simulink で Autoware 用のモデルを作成するのを補助する．ここでは，そのテンプレートを利用することで，作業量どれだけ削減されるか議論する．

MATLAB スクリプトを使ってモデルを作成する際，そのスクリプトは Robotics System Toolbox によって提供された関数を用いて実装する．MATLAB テンプレートスクリ

表 4 Robotics System Toolbox のみを利用した Autoware と提案フレームワークによって利用可能な機能.

	Communication between Autoware and MATLAB/Simulink	GUI operation for MATLAB/Simulink	Draw node dependency	Generate MATLAB /Simulink templates	Display node information
Only Robotics system toolbox	✓		✓		
The proposed framework	✓	✓	✓	✓	✓

表 5 関連研究と提案フレームワークの比較.

	Code generation	Real-time	GUI	Distributed computing	Recode and replay data	Documentation	Process manager	Target
V-REP [18]		✓		✓				Robots in general
RoboComp [19]	✓		✓		✓	✓	✓	Tool-based robots
CALIPER [20]		✓	✓	✓				Tendon-driven robots
SimTrack [21]		✓		✓		✓		Detection and tracking
TeleKyb [22]				✓		✓		Unmanned aerial vehicles
PX4 [23]		✓	✓	✓			✓	Micro aerial vehicles
ROS/ROS2				✓	✓	✓		Robots in general
MATLAB/Simulink	✓	✓	✓			✓		Model based development
Robotics System Toolbox	✓	✓	✓	✓		✓		Model based development
The proposed framework	✓	✓	✓	✓	✓	✓	✓	Autonomous driving vehicles

プトによるテンプレート生成量を，表 2 に記す．MATLAB テンプレートスクリプトは，3.1 章に示した，ノードに必要不可欠な情報の定義と，コールバック関数の定義を行う．例えば，経路計画に必要な lane_stop ノードには，１つのパブリッシャと 5 つのサブスクライバがある．テンプレートスクリプトでは，ノード名，サブスクライバ，パブリッシャの定義に 1 行，コールバック関数の定義に 2 行のコードを生成する．そのため，lane_stop ノードでは 17 行の MATLAB テンプレートスクリプトが生成される．

Simulink モデルの実装には，Simulink ブロックの配置と設定，モデル名の定義，及びブロックをつなげる必要がある．Simulink テンプレートモデルによって設置されるブロック数と設定数を表 3 に示す．Simulink テンプレートモデルでは，モデル名の定義と必要不可欠ブロックの設定を行い，それらのブロックをつなげ合わせる．例えば，lane_stop ノードの Simulink テンプレートモデルを生成した場合，合計 18 のブロックの設置と 31 の設定を行う．

この MATLAB/Simulink テンプレート生成機能を利用しない場合，開発者はノード情報を自身で調査し，それを MATLAB スクリプトまたは Simulink モデルで定義する必要がある．一方で，この機能を使うことで，その作業を削減できる．そのため，モデリング効率を改善する．

4.3 ユーザビリティ

Robotics System Toolbox によって Autoware と MATLAB/Simulink は通信可能となる．加えて，提案フレームワークでは Autoware 内で MATLAB/Simulink の操作を可能にし，開発効率を改善する機能を提供した．ここでは，Robotics System Toolbox のみを用いた場合と，提案フレームワークを用いた場合とで使える機能を比較する（表４）．

Robotics System Toolbox を使うことで ROS マスタにアクセス可能となるため，どちらの場合でも Autoware と MATLAB/Simulink 間の通信は可能である．しかし，提案フレームワークでのみ Runtime Manager を使って MATLAB/Simulink の起動及び MATLAB/Simulink コードの実行が可能である．加えて，Robotics System Toolbox のみを利用した場合でも，Autoware で rqt_graph プラグインが利用可能なため，ノードの依存関係を描画できるが，提案フレームワークでは rqt_graph_autoware プラグインが利用可能なため，Autoware 機能ごとのノードの依存関係描画も可能である．さらに，提案フレームワークでは MATLAB/Simulink テンプレートの生成，及びノード情報の表示を Runtime Manager から行うことができる．上記に示したように，Autoware で利用可能な機能が増加したことから，ユーザビリティが向上したといえる．

5. 関連研究

ROS に基づくロボットシステムは広く開発されおり，様々なフレームワークが存在する．この章では，ROS に基づいたロボット用フレームワークに関する先行研究に関して提案フレームワークと比較する．加えて，ROS 及び ROS を拡張した ROS2 [24]，MATLAB/Simulink，Robotics System Toolbox を提案フレームワークと比較する．

V-REP [18] ： V-REP （ Virtual Robot Experimentation Platform）は，多目的でスケーラブルな MATLAB/Simulink と連携可能なシミュレーションフレームワークである．V-REP は，モデルに応じて切り替え可能な独立した機能を持っており，経路計画等に利用可能な様々な計算モジュールを適用している．一方で，提案フレームワークも自動運転システム用の V-REP のような機能を備えており，それらを GUI ツールで利用できる．

RoboComp [19]：RoboComp は，使いやすさと開発スピードに重点を置いたツールベースのロボットフレームワークである．RoboComp は，componentGecerator，managerCom，monitorComp，replayComp，loggerComp といった能を提供している．componentGecerator はコード生成機能であり，提案フレームワークも MATLAB/Simulink テンプレート生成機能を備えている．managerComp は，Runtime Manager のようにプロセスを管理するための GUI ツールである．monitorComp はコンポーネントを可視化するツールであり，RViz と同様の役割を果たす．replayComp は rosbag のように出力の記録・再生を行う．loggerComp コンポーネント分析の用の標準出力器である．

CALIPER [20]：CALIPER は，腱駆動ロボット用の普遍的でカスタマイズ可能なロボットシミュレーションフレームワークである．そのソフトウェア構造は，コンポーネントに基づいて設計されており，CORBA [25] が個々のコンポーネントの分配を可能にしている．CALIPERの主な特徴はリアルタイム性であり，提案フレームワークも実用性の評価からリアルタイム性を持つことが分かる．

SimTrack [21]： SimTrack は，リアルタイムに物体の姿勢を検出し，追跡するのに用いられるオープンソースフレームワーク [26] である．SimTrack は，マニピュレータに取り付けられた複数のカメラから複数のオブジェクトを識別し，追跡できる．この画像処理には，Autoware と同様にNVIDIA 製の GPU が用いられている．

TeleKyb [22]：TeleKyb（Tele-Operation Platform of the MPI for Biological Cybernetics）は，一般的な無人航空機の制御フレームワークである．TeleKyb では，ROS を古いバージョンの MATLAB/Simulink に対してコンパイル可能な環境を構築することで，ROS と MATLAB/Simulink 間の Publish/Subscribe モデルを用いた通信を可能にしている．しかし，Robotics System Toolbox を用いることでこのような複雑な設定や条件なしで通信可能となる．

PX4 [23]：PX4 は Publish/Subscribe モデルを用いた組み込みプラットフォーム用のオープンソースプラットフォームである．PX4 では Runtime Manager に似た GUI ツールでシステムの管理を行うことができる．

表５は，上記のフレームワーク，ROS/ROS2，MATLAB/ Simulink，Robotics System Toolbox，及び提案フレームワークの特徴，機能の有無，及び対象とするロボットまたは制御デバイスの特性を要約している．表５に示す通り，提案フレームワークは ROS/ROS2 と同様に分散コンピューティング及び rosbag の記録と再生が可能となっている．MATLAB/ Simulink と Robotics System Toolbox では，C/C++ コードの生成が可能であり，リアルタイム性を示し，各機能を GUI で操作できる．一方で，提案フレームワークでは，MATLAB/Simulink テンプレート生成機能を持ち，Runtime Manager を使って各プロセスを管理しながら，シミュレーションを行うことができる．

6. おわりに

本論文では，Autoware と MATLAB/Simulink の統合開発フレームワークの提案をした．提案フレームワークは Robotics System Toolbox を用いることで，Autoware と MATLAB/Simulink 間の通信を可能にし，開発効率を改善するための機能を提供した．まず，MATLAB/Simulink のデータ転送時間と処理能力を調べることで，提案フレームワークがコシミュレーション，及び自動運転車を用いた実証実験で実用可能なことを示した．そして，テンプレート生成機能を提供し，モデリング効率を改善した．加えて，提案フレームワークによって加えられた機能により，Runtime Manager から MATLAB/Simulink の操作を可能にし，ユーザビリティを向上させた．これらにより，提案フレームワークは開発効率を改善した．

今後，MATLAB/ Simulink モデル，自動運転車で実際に運用可能であることを示す。

参考文献

[1] C. Berger and B. Rumpe, “Autonomous driving - 5 years after the urban challenge: The anticipatory vehicle as a cyber-physical system,” CoRR, vol. abs/1409.0413, 2014.

[2] “ROS.org.” http://www.ros.org.

[3] Y. Saito, T. Azumi, S. Kato, and N. Nishio, “Priority and synchronization support for ROS,” in Proc. of the 4th IEEE International Conference on Cyber-Physical Systems, Networks, and Applications, pp. 77–82, 2016.

[4] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, “An open approach to autonomous vehicles,” IEEE Micro, vol. 35, no. 6, pp. 60–68, 2015.

[5] “Autoware/github” http://github.com/CPFL/Autoware.

[6] “MATLAB/Simulink.” http:///www.mathworks.com.

[7] “Robotics System Toolbox.” http://mathworks.com/products/robotics.html.

[8] “ROS.org/RViz.” http://wiki.ros.org/rviz.

[9] “ROS.org/rqt.” http://wiki.ros.org/rqt.

[10] “ROS.org/rosbag.” http://wiki.ros.org/rosbag.

[11] “Qt.io.” http://www.qt.io.

[12] J. Friedman, “MATLAB/Simulink for automotive systems design,” in Proc. of the Conference on Design, Automation and Test in Europe, DATE ’06, pp. 87–88, 2006.

[13] T. Bamford, K. Esmaeili, and A. P. Schoellig, “A real-time analysis of rock fragmentation using UAV technology,” CoRR, vol. abs/1607.04243, 2016.

[14] “ROS.org/rqt graph.” http://wiki.ros.org/rqt graph.

[15] “ROS.org/rosnode.” http://wiki.ros.org/rosnode.

[16] “wxpython.org.” http://wxpython.org.

[17] “wxglade.sourceforge.net.” http://wxglade.sourceforge.net.

[18] E. Rohmer, S. P. N. Singh, and M. Freese, “V-rep: A versatile and scalable robot simulation framework,” in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1321–1326, 2013.

[19] L. Manso, P. Bachiller, P. Bustos, P. N´u˜nez, R. Cintas, and L. Calderita, RoboComp: A Tool-Based Robotics Framework, pp. 251–262. Springer Berlin Heidelberg, 2010.

[20] S. Wittmeier, M. Jntsch, K. Dalamagkidis, M. Rickert, H. G. Marques, and A. Knoll, “Caliper: A universal robot simulation framework for tendon-driven robots,” in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1063–1068, 2011.

[21] K. Pauwels and D. Kragic, “Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking,” in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1300–1307, 2015.

[22] V. Grabe, M. Riedel, H. H. Blthoff, P. R. Giordano, and A. Franchi, “The telekyb framework for a modular and extendible ROS-based quadrotor control,” in Proc. of the European Conference on Mobile Robots, pp. 19–25, 2013.

[23] L. Meier, D. Honegger, and M. Pollefeys, “Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms,” in Proc. of the IEEE International Conference on Robotics and Automation, pp. 6235–6240, 2015.

[24] Y. Maruyama, S. Kato, and T. Azumi “Exploring the Performance of ROS2," in Proc, of the ACM SIGBED International Conference on Embedded Software, pp. 5:1-5:10, 2016.

[25] “CORBA” http://www.corba.org.

[26] “SimuTrack/git” http://github.com/karlpauwels/simtrack.