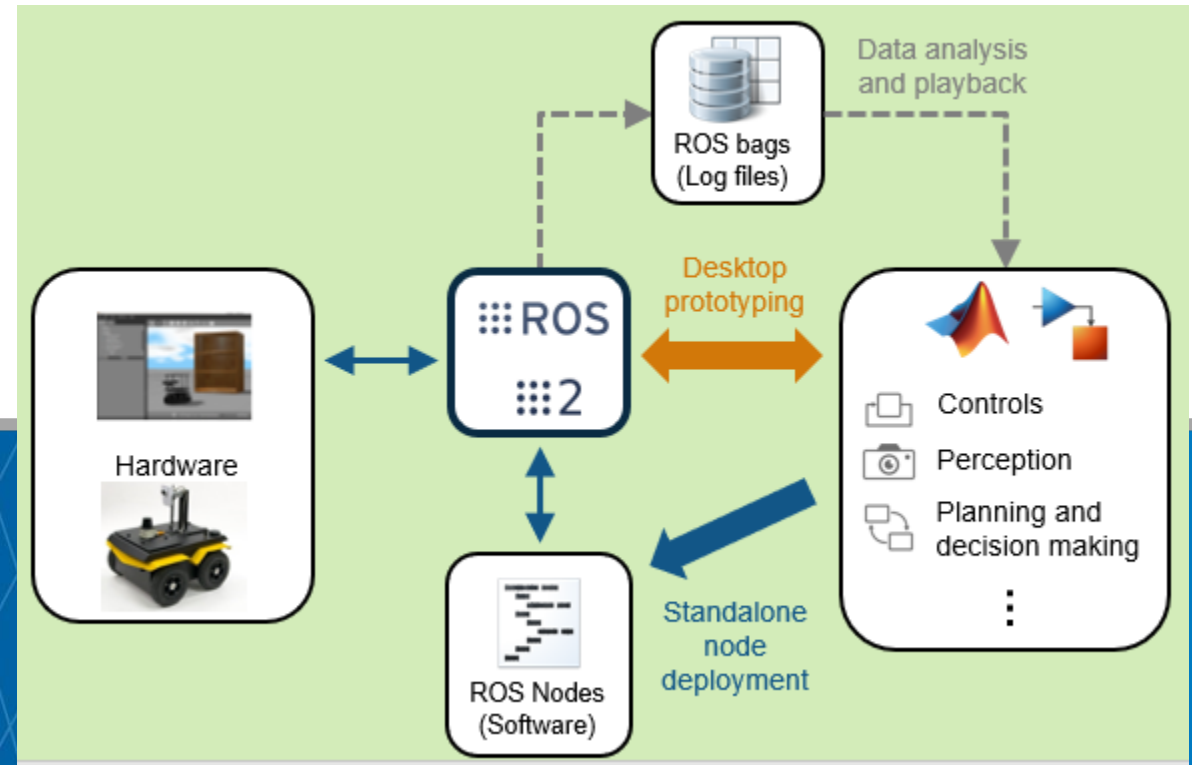


ROS Toolbox による ROS / ROS 2 連携

MathWorks Japan

アプリケーションエンジニアリング部

木川田 亘



ROS 2による移動ロボットの標識追従

The image displays a Simulink Stateflow chart for "Sign Tracking Logic" and a Gazebo simulation environment. The Stateflow chart includes logic for scanning for objects, checking for minimum angles, turning, and tracking the sign. The Gazebo environment shows a robot in a maze following a sign.

Stateflow Chart Logic:

- ScanForObject:**
 - Condition: $scanDir = 1, stopRobot = false$
 - Action: $v = 0, w = 0.25 * scanDir$
 - Logic: $\% Track blobs larger than minimum size [minBlobSize == 200]$
 - Logic: $\% Scan for blobs larger than minimum size [xBlobSize < 200]$
- CheckAngle:**
 - Condition: $\% Check if the robot has turned a minimum angle [abs(angDiff) > pi/2]$
- TurnPlace:**
 - Action: $rotAng = posRot, v = 0, w = 0.25 * scanDir$
 - Logic: $\% Stop tracking sign when: \% speed commands are zero [w == 0 && v == 0]$
 - Logic: $\% Stop if the biggest object was not [dir == 1]$
- Stop:**
 - Action: $w = 0, v = 0, stopRobot = true$
- TrackObject:**
 - Condition: $[blobSize] = max(blobSize), blob = blob(blob)$
 - Logic: $\% Turn right if green [scanDir = -1, \% Turn left otherwise scanDir = 1, \% Turn left otherwise scanDir = 1]$
 - Logic: $\% Turn left otherwise scanDir = 1$
 - Logic: $\% Turn right if green scanDir = -1$
 - Logic: $\% Turn left otherwise scanDir = 1$

Gazebo Simulation:

- Environment: ROS Melodic Dashing Gazebo v9 - VMware Workstation 15 Player
- Time: Thu 22:38
- Real Time Factor: 0.10
- Sim Time: 00 00:26:09.288
- Performance: 20x faster

アジェンダ

- ROS Toolboxの概要
- ユーザー事例
- ROS / ROS2とは？
- ROS / ROS2 連携のワークフロー
- ROS / ROS2 連携アプリケーションの例
- まとめ

アジェンダ

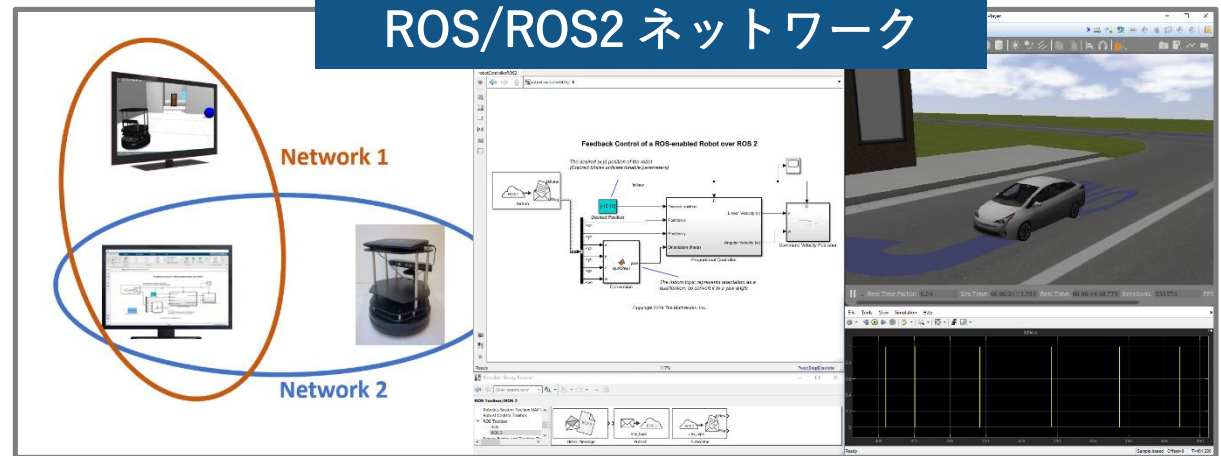
- ROS Toolboxの概要
- ユーザー事例
- ROS / ROS2とは？
- ROS / ROS2 連携のワークフロー
- ROS / ROS2 連携アプリケーションの例
- まとめ

ROS Toolbox概要

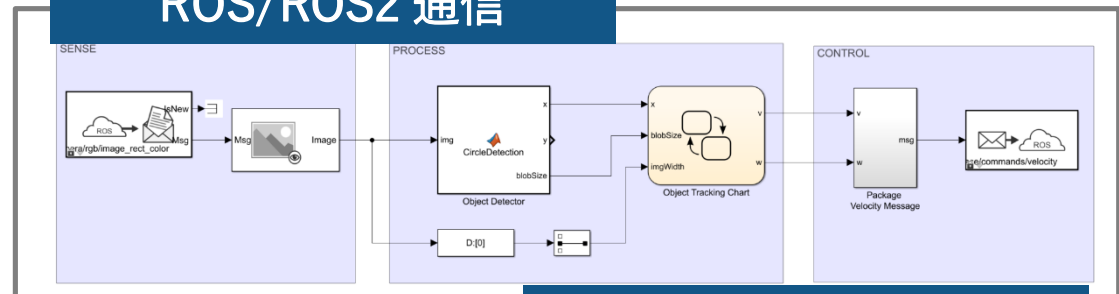
ROS/ROS2ベースのアプリケーションのデザイン、シミュレーションおよび実装

- MATLABとSimulinkをROSおよびROS2に直接接続
 - MATLAB/Simulinkの資産をROS接続
 - MATLAB/SimulinkをROSノード化
- rosbagの読み込みおよびプレイバックをサポート
 - オフラインで解析やテストが可能
- SimulinkからのROS/ROS2ノード生成に対応
 - C++でROSのコーディング不要
 - catkin/colconでのビルドに必要なファイルも自動生成

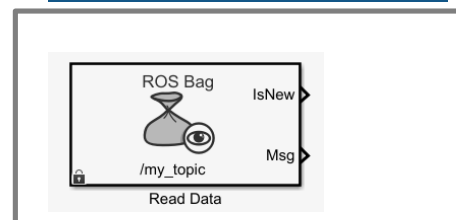
ROS/ROS2 ネットワーク



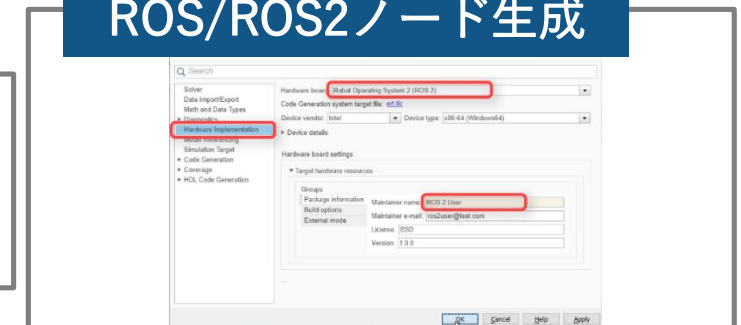
ROS/ROS2 通信



rosbag再生



ROS/ROS2ノード生成



製品分割で ROS Toolbox が誕生

R2019a 以前

Robotics System
Toolbox

ROSとロボットアルゴリズム全般

3-way split

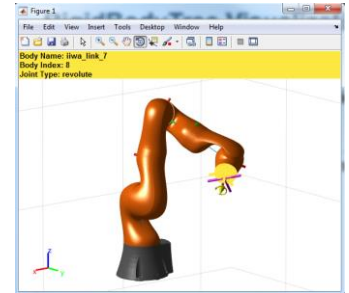
R2019b



Robotics System
Toolbox



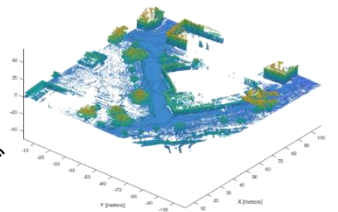
ロボットアーム
移動ロボット
UAV/ドローン
エンジニア



Navigation Toolbox



SLAM・パスプランニング
エンジニア



ROS Toolbox



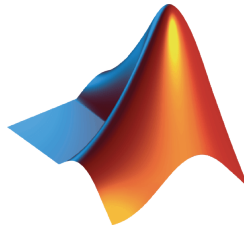
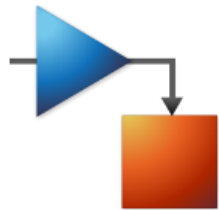


ロボティクス/自動運転
エンジニア



ROSに特化したToolboxとしてRobotics System Toolboxから分離

ROS Toolbox でサポートされている連携機能

	 ROS	 2
	<ul style="list-style-type: none"> • Topic – Publish / Subscribe • Service – Server / Client • Action – Client • Parameter – Get / Set • カスタムメッセージ (<i>add-on</i>) • rosbag 読み込み 	<ul style="list-style-type: none"> • Topic – Publish / Subscribe • カスタム messages (<i>built-in</i>)
	<ul style="list-style-type: none"> • Topic – Publish / Subscribe • Service – Call • Parameter – Get / Set • ROS Time • rosbag 再生 • ノード生成 (<i>roscpp</i>) 	<ul style="list-style-type: none"> • Topic – Publish / Subscribe • ノード生成 (<i>rclcpp</i>)
ROS バックエンド	<ul style="list-style-type: none"> • ROS Indigo 	<ul style="list-style-type: none"> • ROS2 Dashing R2020a

アジェンダ

- ROS Toolboxの概要
- ユーザー事例
- ROS / ROS2とは？
- ROS / ROS2 連携のワークフロー
- ROS / ROS2 連携アプリケーションの例
- まとめ

Voyageがモデル予測制御とROSによるタクシーの自動運転

課題

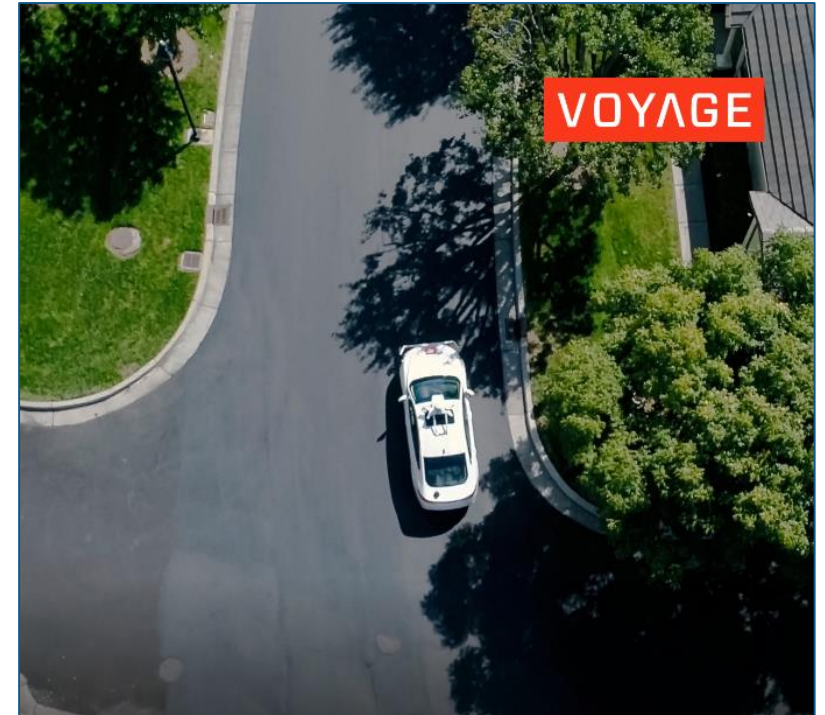
自動運転のための障害物から安全な距離を取りつつ、目標速度に追従するコントローラの開発

ソリューション

Simulink®を使用して、縦方向モデル予測コントローラを設計
MATLAB にインポートされた実験データに基づいてパラメータを調整
Robotics System Toolbox™を使用して
ROS ノードとしてコントローラを展開
Simulink Coder™を使用してソースコードを生成し、
Dockerコンテナとしてパッケージ化

結果

- 開発速度が3倍になった
- オープンソースソフトウェアと容易に統合
- Simulinkアルゴリズムを製品ソフトウェアとして活用



Voyage's self driving car in San Jose, California.

"We were searching for a prototyping solution that was fast for development and robust for production. We decided to go with Simulink for controller development and code generation, while using MATLAB to automate development tasks."

- Alan Mond, Voyage

Clearpath Roboticsが産業用ロボットのアルゴリズム開発を加速

課題

産業用ロボットのレーザーベースの認識、コンピュータービジョン、走行管理、制御アルゴリズムを短期間で開発する

ソリューション

MATLABを、ROSデータの可視化と解析、プロトタイピング、最新のロボット研究のアルゴリズムの適用に活用

結果

- データ解析時間が最大50%短縮
- 顧客とのコミュニケーションを効率化
- 最新の自動運転アルゴリズムを素早く組み込み



An OTTO self-driving vehicle from Clearpath Robotics.

“ROS is good for robotics research and development, but not for data analysis. MATLAB, on the other hand, is not only a data analysis tool, it’s a data visualization and hardware interface tool as well, so it’s an excellent complement to ROS in many ways.”

- Ilia Baranov, Clearpath Robotics

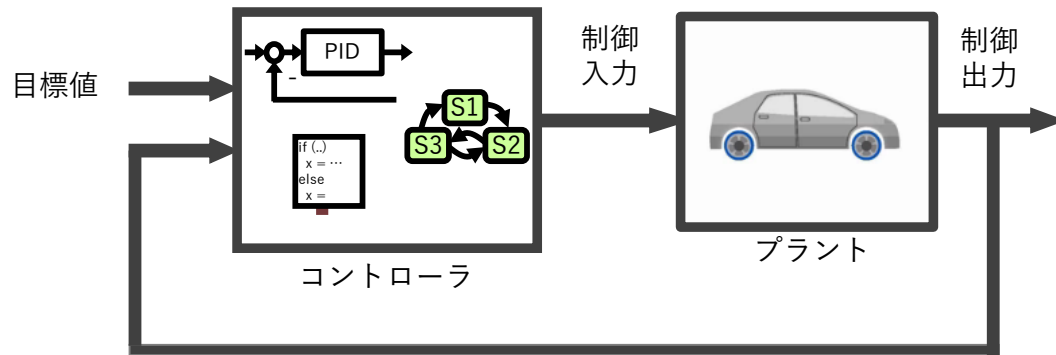
アジェンダ

- ROS Toolboxの概要
- ユーザー事例
- ROS / ROS2とは？
- ROS / ROS2 連携のワークフロー
- ROS / ROS2 連携アプリケーションの例
- まとめ

自律型システム開発とその課題

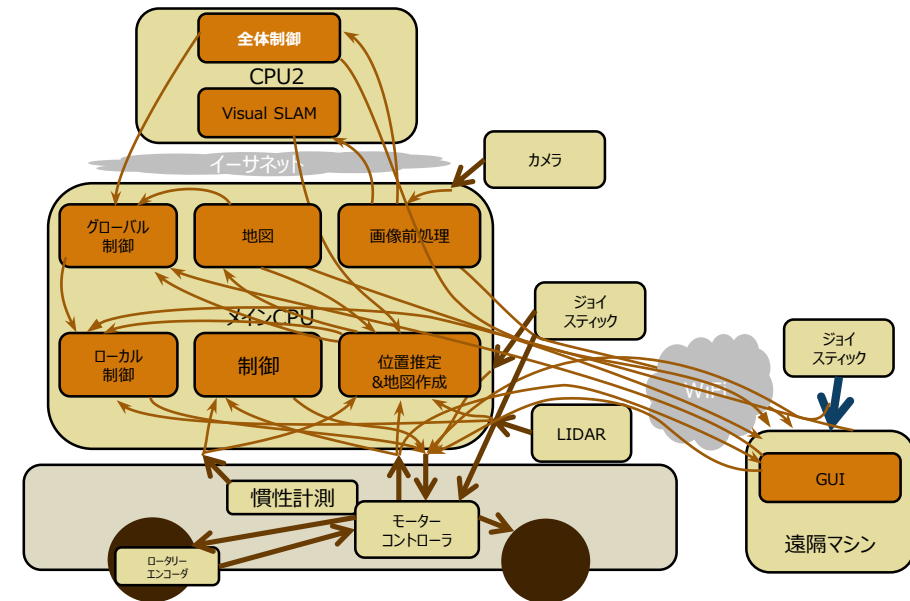
従来の制御システム

- シングルループの同期処理
- 高いリアルタイム性
- センサーは時系列の1次元
- 単一のアクチュエータ



自律型システム

- 複数周期の同期処理・非同期処理の混在
- マルチコア/GPUの並列分散処理が不可欠
- 高解像度/複数レート of センサーを連動
- 複合的なアクチュエーター(車輪とアームなど)



複雑な自律分散型システムを開発のするためのフレームワークが必要

ROS とは?



- ROS = Robot Operating System
- 自律分散システムのためミドルウェア
 - 一般的なOSではない – ミドルウェアの一種
- メッセージング通信/分散コンピューティングのためのツール群
- アプリケーションのパッケージ化とビルドのための管理システム
- 活発なコミュニティ
 - ドライバー、シミュレーション、モーションプランニング、認知、etc.
- ROS 1を刷新したROS 2が近年リリース
 - コミュニティでもROS 2 への移行が少しずつ始まっている

ROSを活用することで自律分散システムを効率的に開発可能

ROS と ROS 2 の比較

<http://design.ros2.org/articles/changes.html>

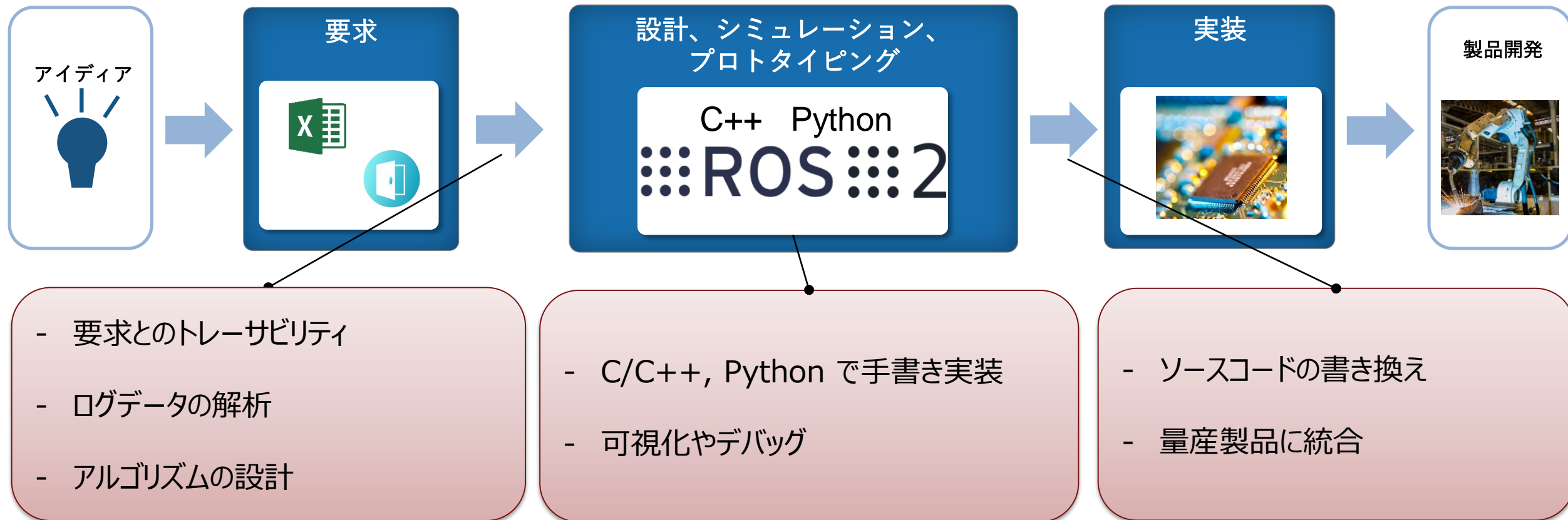
	ROS	ROS 2
最初のリリース	2010年1月22日	2017年12月8日
公式サポートしているOS	<ul style="list-style-type: none"> Ubuntu 16.04 / 18.04 	<ul style="list-style-type: none"> Ubuntu 16.04 / 18.04, OS X El Capitan / macOS Sierra Windows 10
開発言語	<ul style="list-style-type: none"> Python 2 (2019年12月サポート終了) C++03 標準 	<ul style="list-style-type: none"> Python 3.5以降 C++11 / C++14 標準
ビルドシステム	<ul style="list-style-type: none"> catkin 	<ul style="list-style-type: none"> ament (colconでcatkinとamentを双方サポート)
アーキテクチャ	<ul style="list-style-type: none"> 独自実装のメッセージ通信プロトコル マスターに接続し他のノードを参照 同一プロセス内複数ノードはオプション ノード起動 – roslaunch (XML) 	<ul style="list-style-type: none"> DDS(Data Distribution Service)ベース マスターレス、ドメインIDで区別 同一プロセス内の複数ノードを標準サポート ノード起動 – ros2 launch (Python)
用途	<ul style="list-style-type: none"> 学術研究やプロトタイピング 産業用途でも活用 	<ul style="list-style-type: none"> 学術研究、プロトタイピング、製品開発 ROS 1に加えてマルチロボット、組み込み向け、リアルタイムシステムに適用可能

ROS 2は発展途上だが今後主流となっていく

アジェンダ

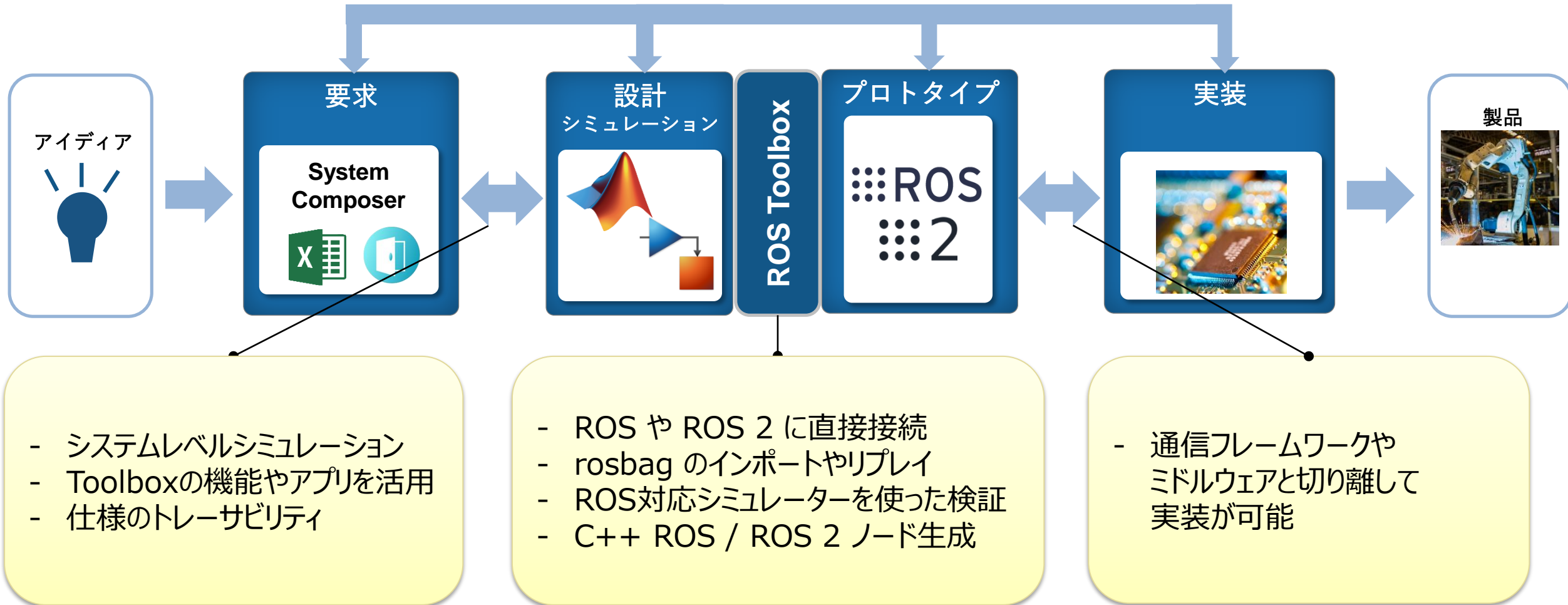
- ROS Toolboxの概要
- ユーザー事例
- ROS / ROS2とは？
- **ROS / ROS2 連携のワークフロー**
- ROS / ROS2 連携アプリケーションの例
- まとめ

手書きコードによるロボット開発ワークフロー

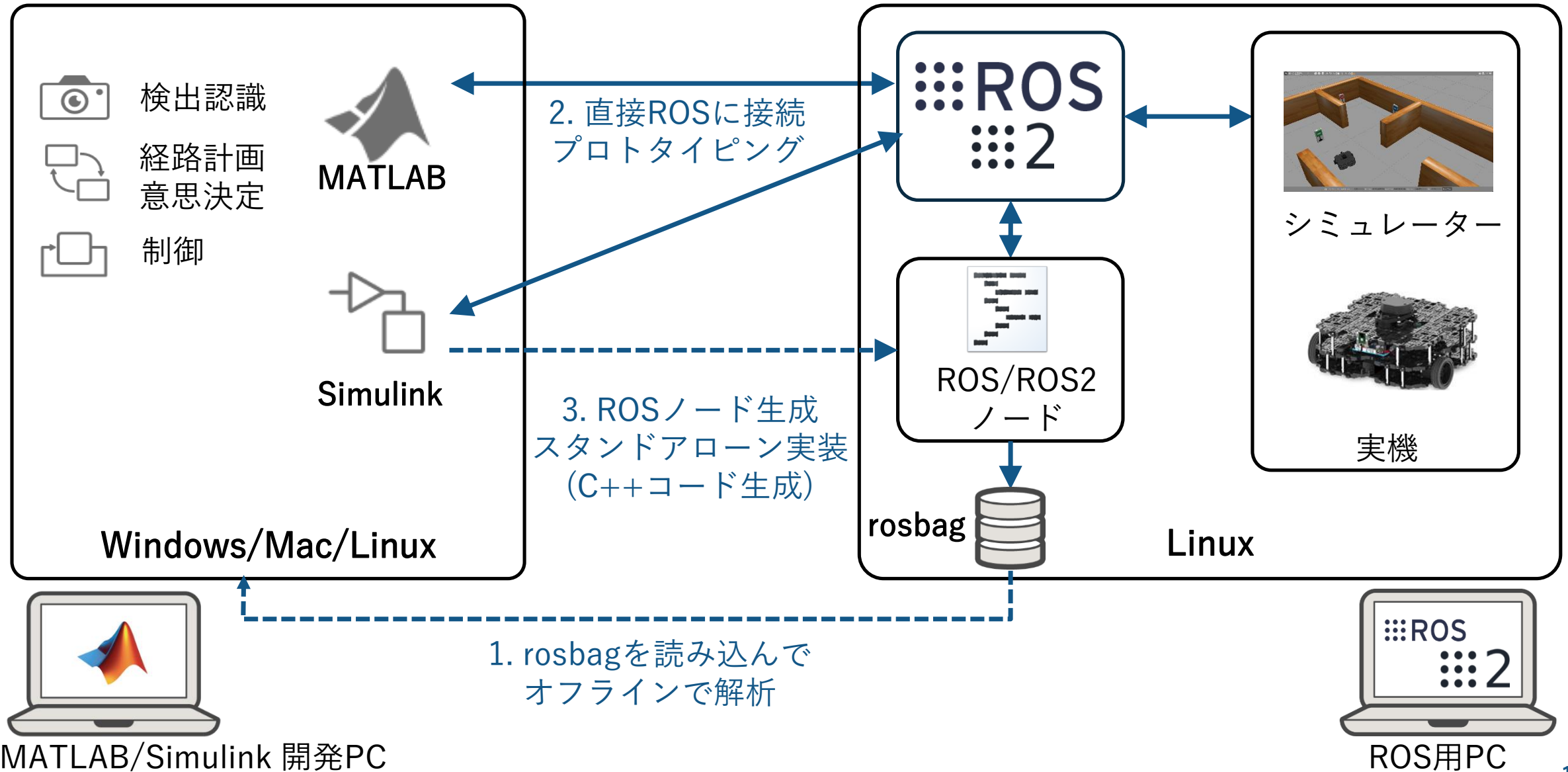


ROS Toolboxによるロボット開発ワークフロー

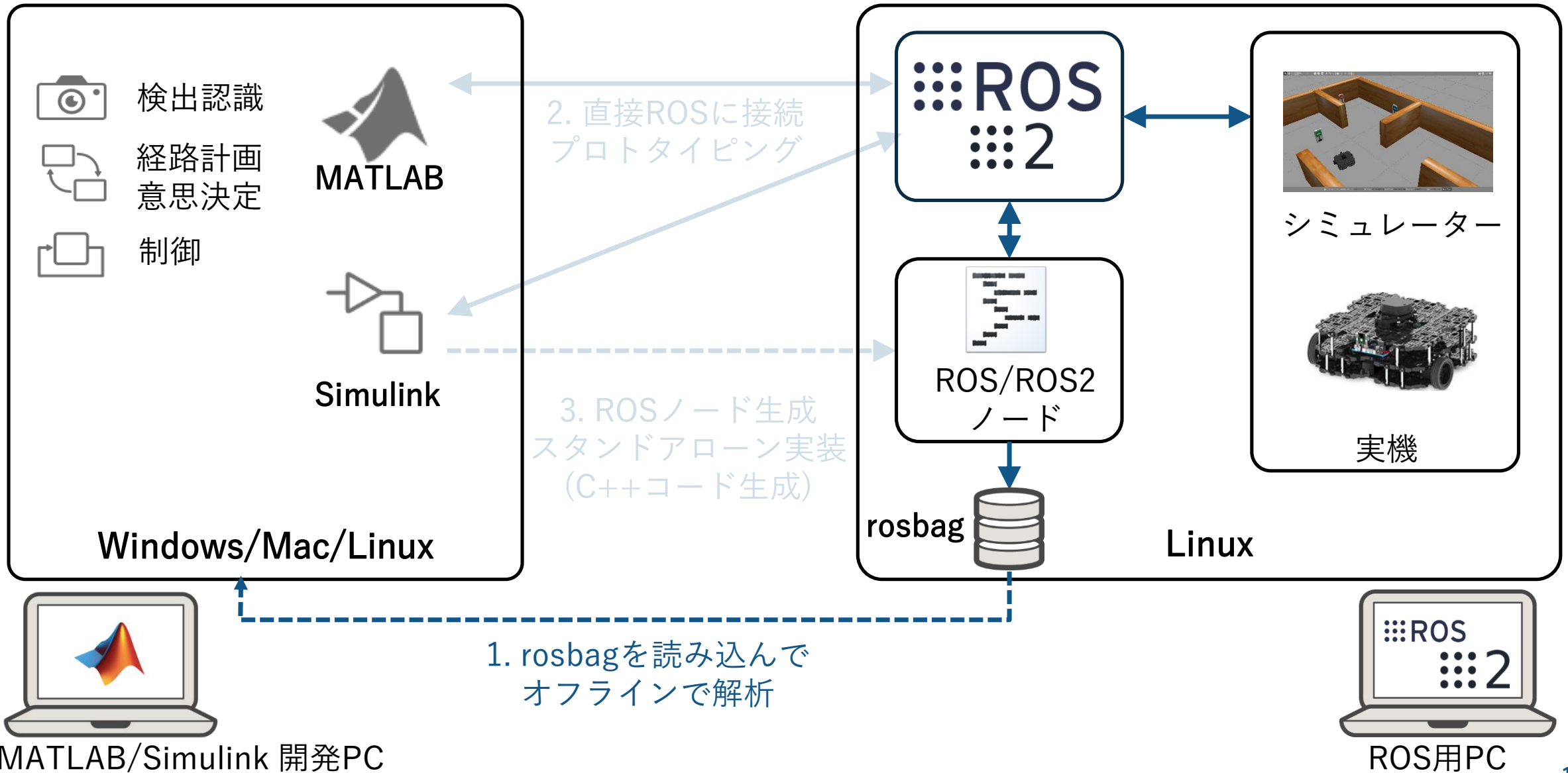
開発の手戻りを最小化



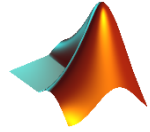
ROS / ROS 2 連携機能



ROS / ROS 2 連携機能



1. rosbagによるオフライン解析



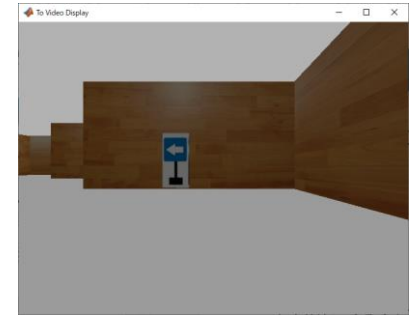
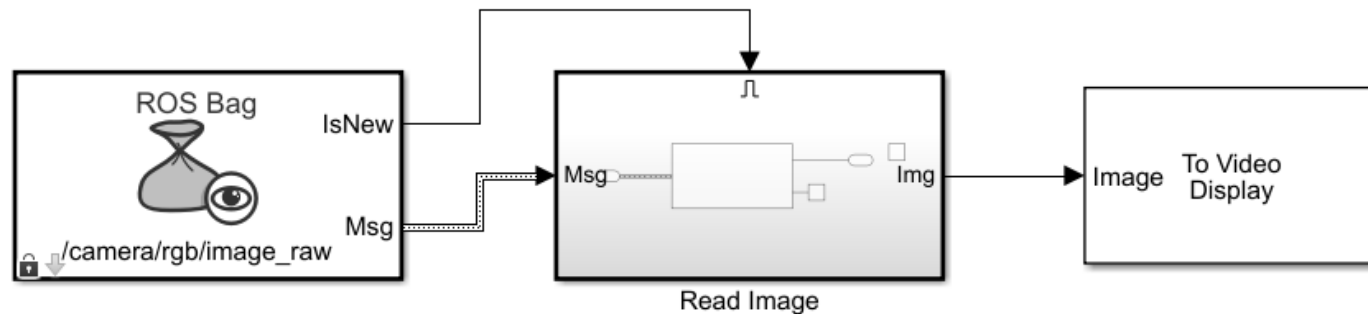
```

bagselect = rosbag('turtlebot3.bag');
bagselectImage = ...
    select(bagselect, 'Topic', '/camera/rgb/image_raw');
msgStructs = ...
    readMessages(bagselectImage, 1, ...
        "DataFormat", "struct"); % 1番目のメッセージ取り出し
    
```



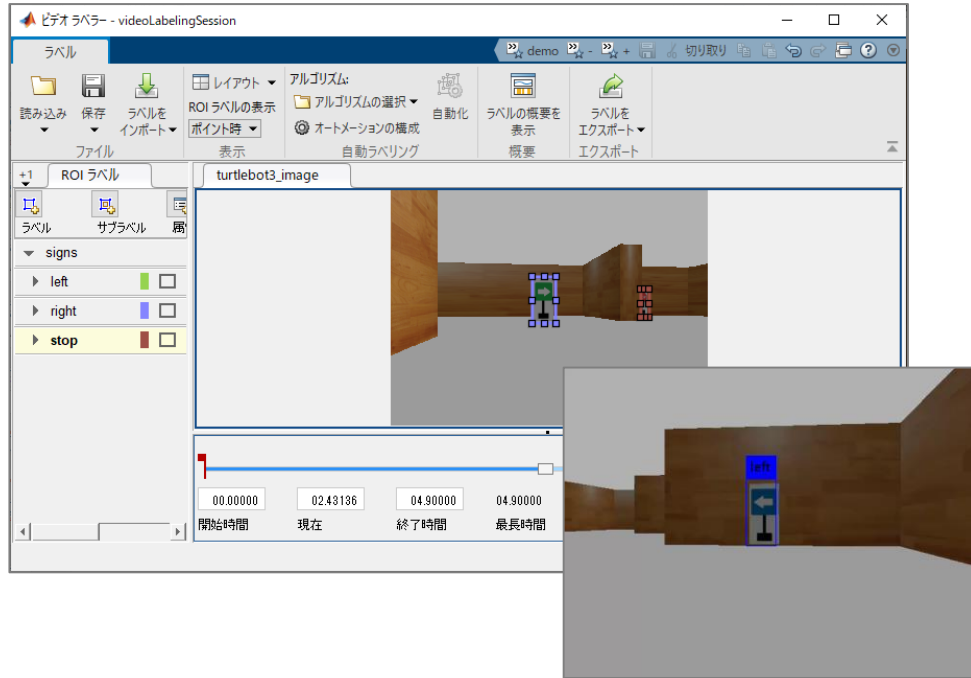
rosbag

ROSが提供しているデータのロギング機能
時系列のデータを保存し、再生することが可能

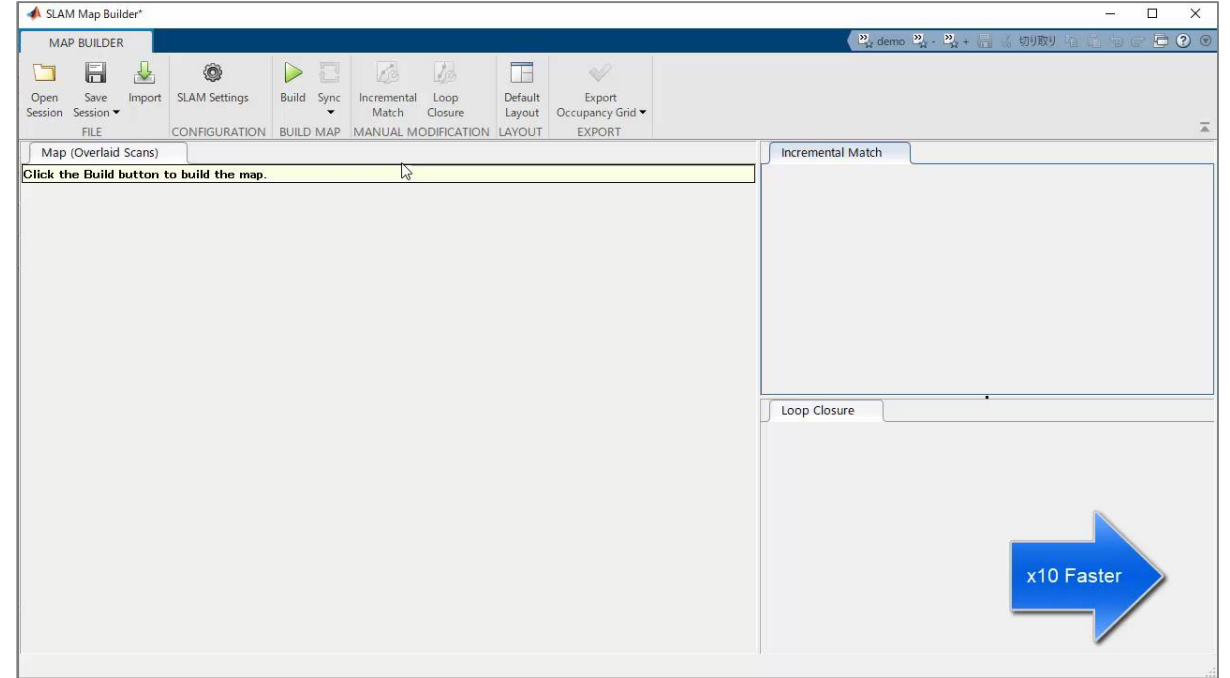


時系列のrosbagから必要なtopicを取り出し、時刻同期や可視化

1. rosbagによるオフライン解析



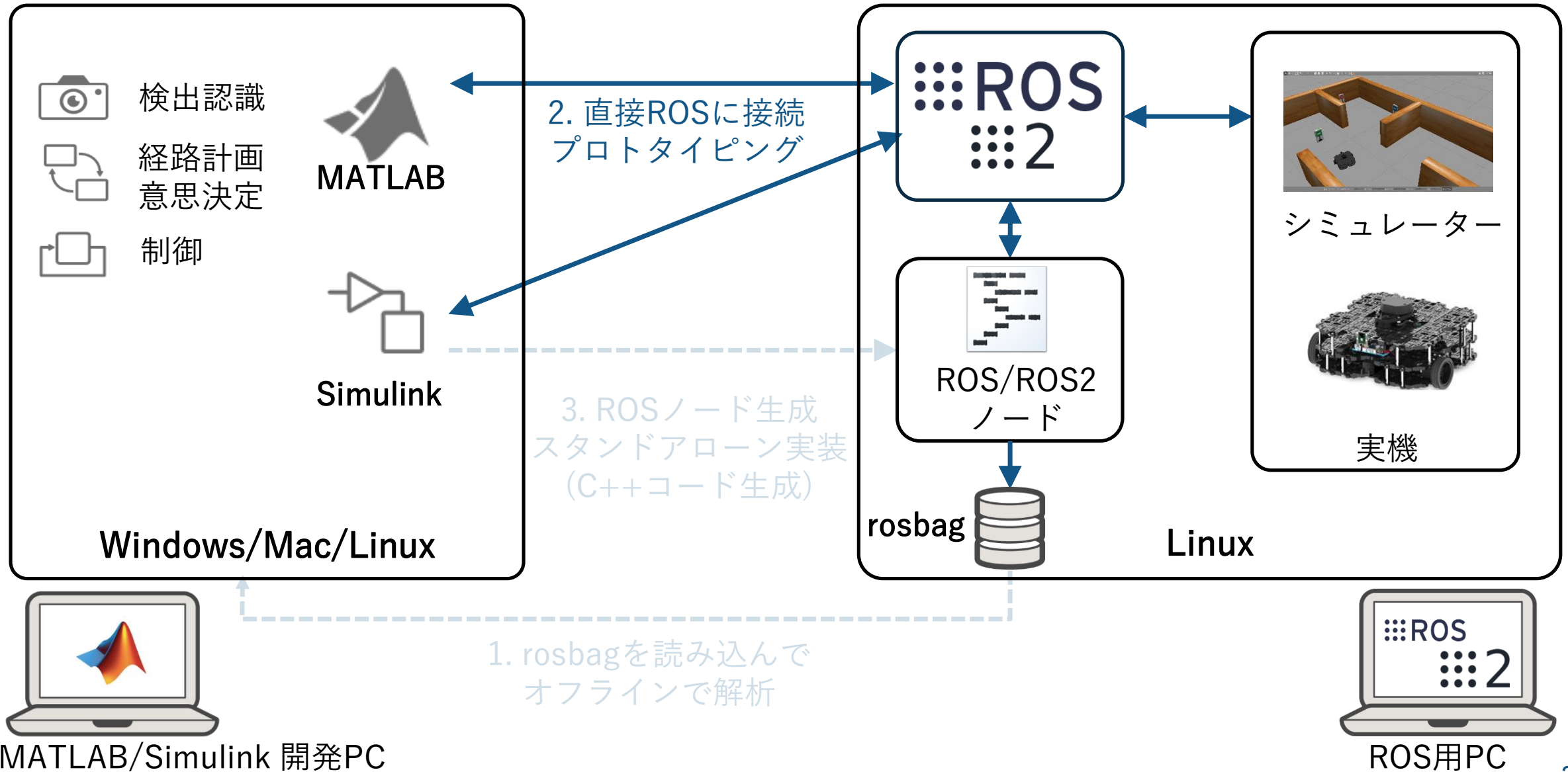
画像topicをラベル付け
YOLO v2/SSDで検出器を学習
Computer Vision Toolbox™, Deep Learning Toolbox™

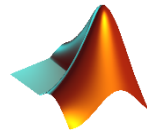


スキャンtopicを使ってSLAMで地図生成
(SLAMマップビルダー)
Navigation Toolbox™

取り出したtopicを使って他のToolboxと連携しアルゴリズム開発

ROS / ROS 2 連携機能





2. 直接 ROS に接続し、プロトタイピング

ROS テンプレート

```

%% ROS Template
rosinit('ipAddress')
odomSub = rossubscriber('/odom');
[velPub,velMsg] = ...
    rospublisher('/mobile_base/commands/velocity');

r = rateControl(10);
while(r.TotalElapsedTime < 20)

    odomMsg = odomSub.LatestMessage;

    % INSERT YOUR ALGORITHM CODE HERE

    velMsg.Angular.Z = ctrlOut;
    send(velPub,velMsg);

    plot(r.TotalElapsedTime,ctrlOut)

    waitfor(r)

end

```

ROS 2 テンプレート

```

%% ROS2 Template
domainId = 1;
node_1 = ros2node('node_1', domainId);
node_2 = ros2node('node_2', domainId);

odomSub = ros2subscriber(node_1, '/odom');
[velPub,velMsg] = ...
    ros2publisher(node_2, '/mobile_base/commands/velocity');

r = rateControl(10);
while(r.TotalElapsedTime < 20)

    odomMsg = odomSub.LatestMessage;

    % INSERT YOUR ALGORITHM CODE HERE

    velMsg.Angular.Z = ctrlOut;
    send(velPub,velMsg);

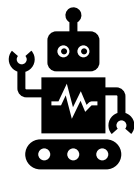
    plot(r.TotalElapsedTime,ctrlOut)

    waitfor(r)

end

```

セットアップ



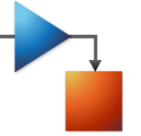
認知

処理

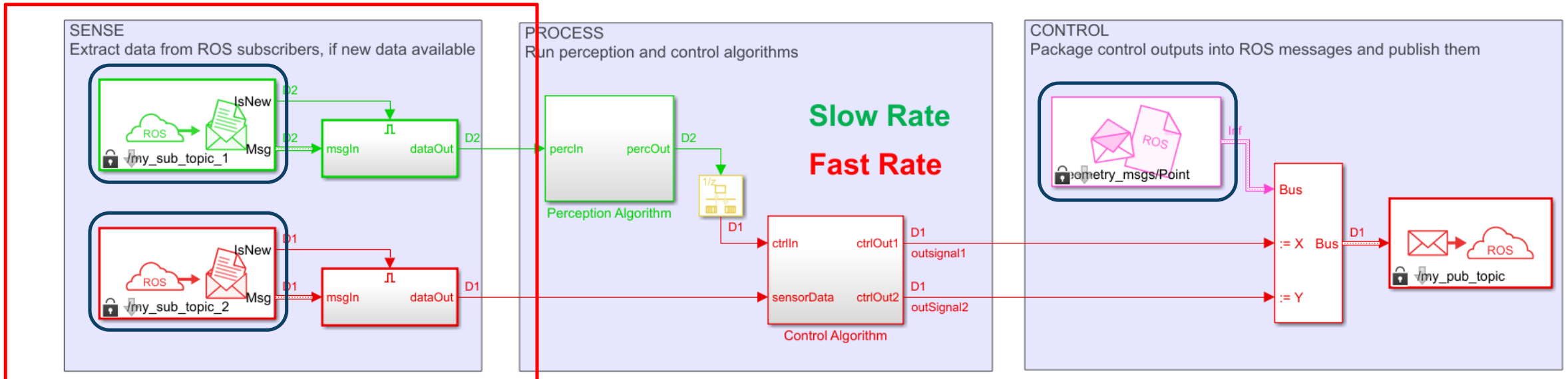
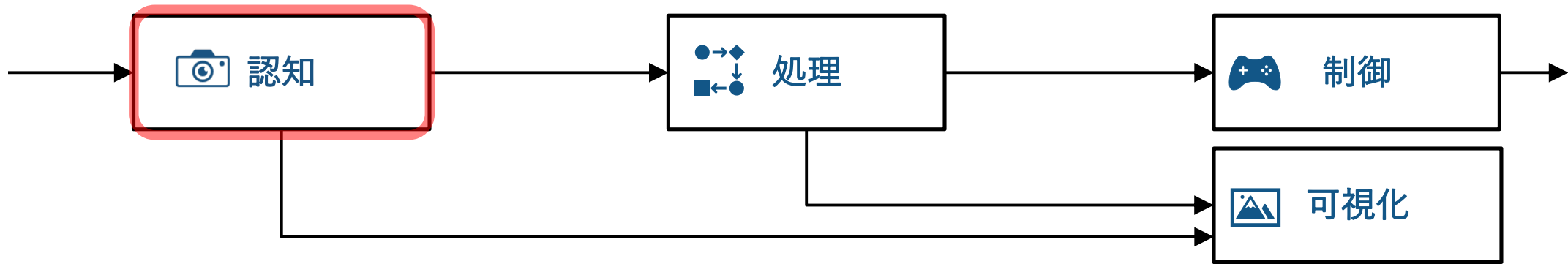
制御

可視化

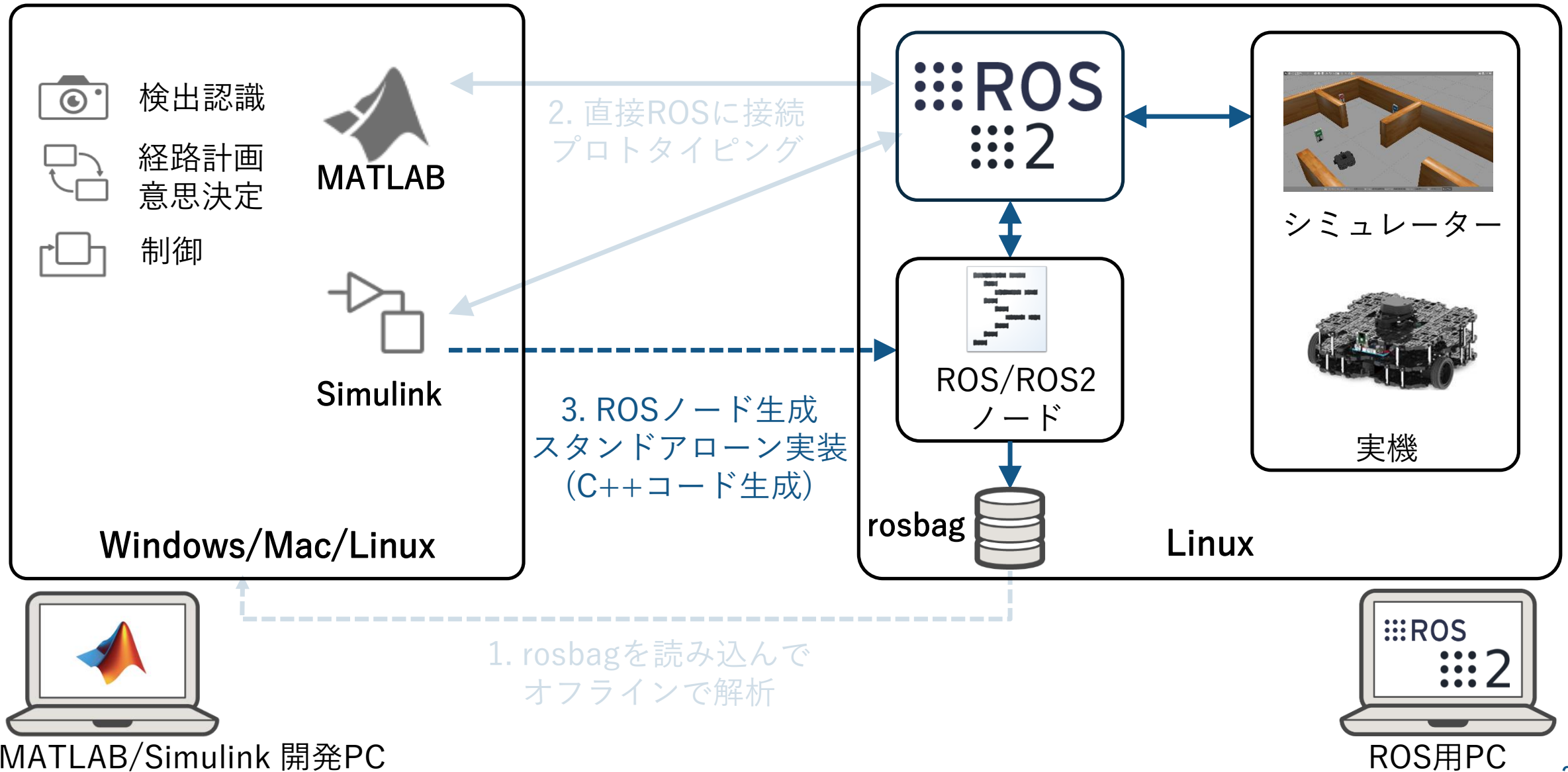
オプション



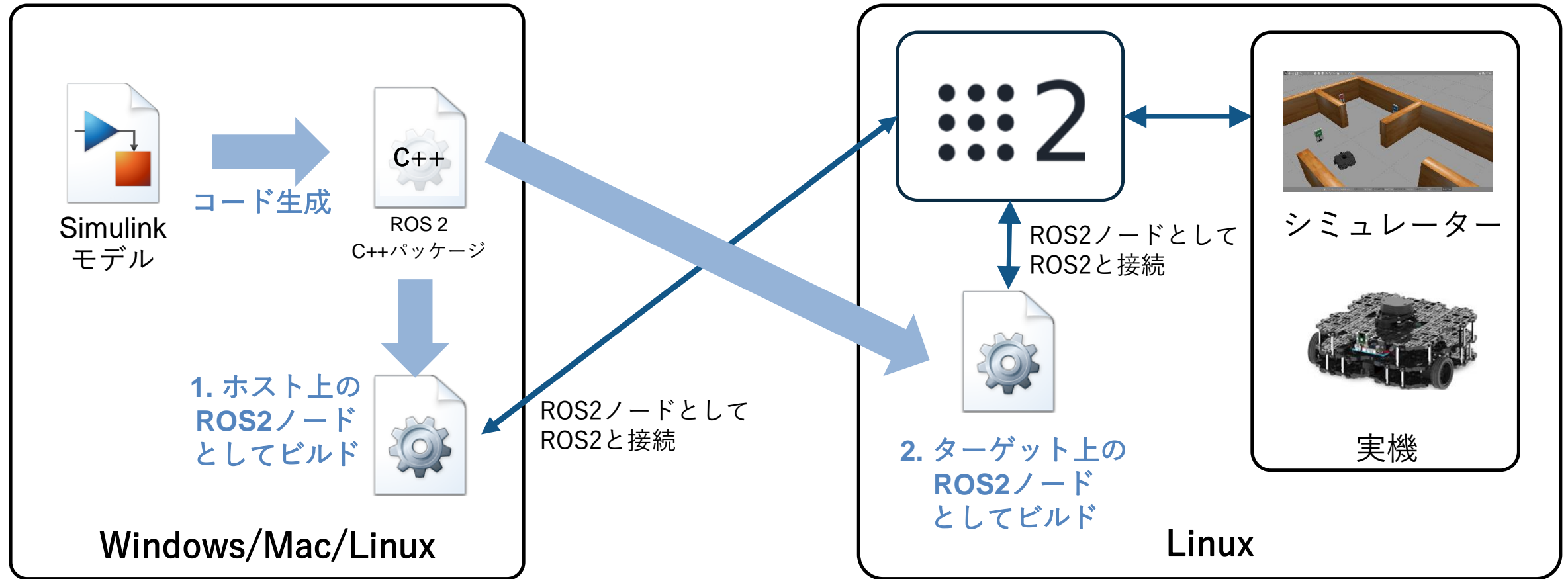
2. 直接 ROS に接続し、プロトタイピング



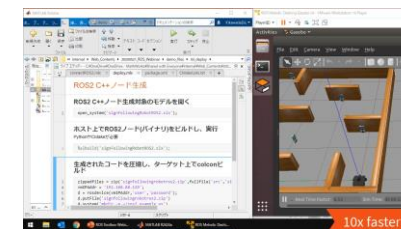
ROS / ROS 2 連携機能



3. ROSノード生成 (ROS2の場合)



MATLAB/Simulink 開発PC



ROS用PC

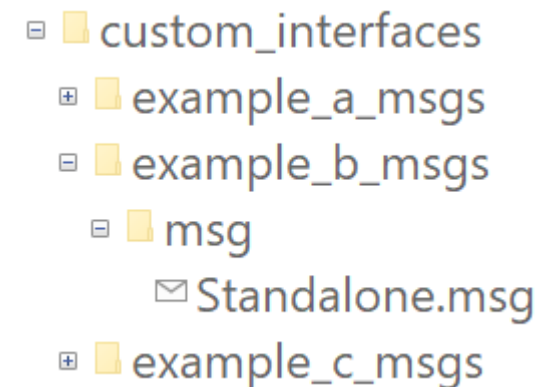
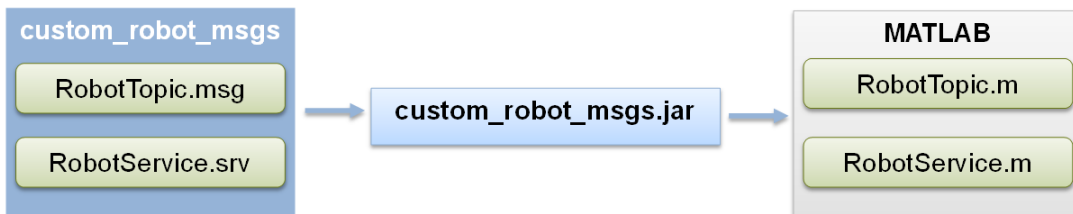
ROS / ROS 2 カスタムメッセージの取り込み

■ ROS1 カスタムメッセージ

- **rosgenmsg** 関数で生成
- .msg/.srvからMATLABで使用する
JARファイルとWrapperのMATLABコード生成
- 下記のサポートパッケージが必要
 - <https://www.mathworks.com/help/ros/ug/ros-custom-message-support.html>

■ ROS2 カスタムメッセージ

- **ros2genmsg** 関数で生成
 - <https://www.mathworks.com/help/R2019b/ros/ug/ros-2-custom-message-support.html>
- 下記の2つが必要
 - Python 3.7+
 - CMake 3.10+



アジェンダ

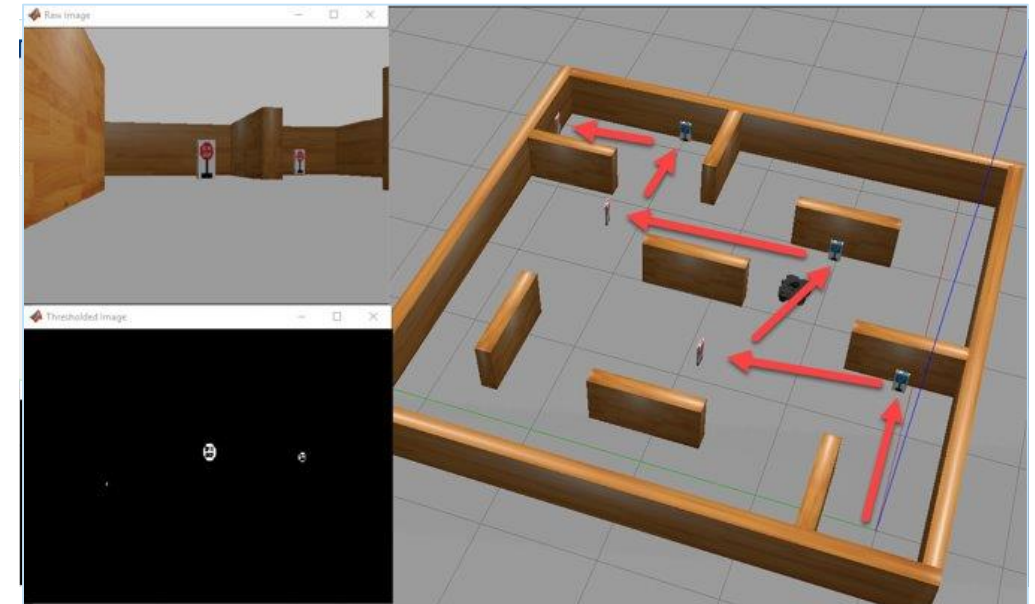
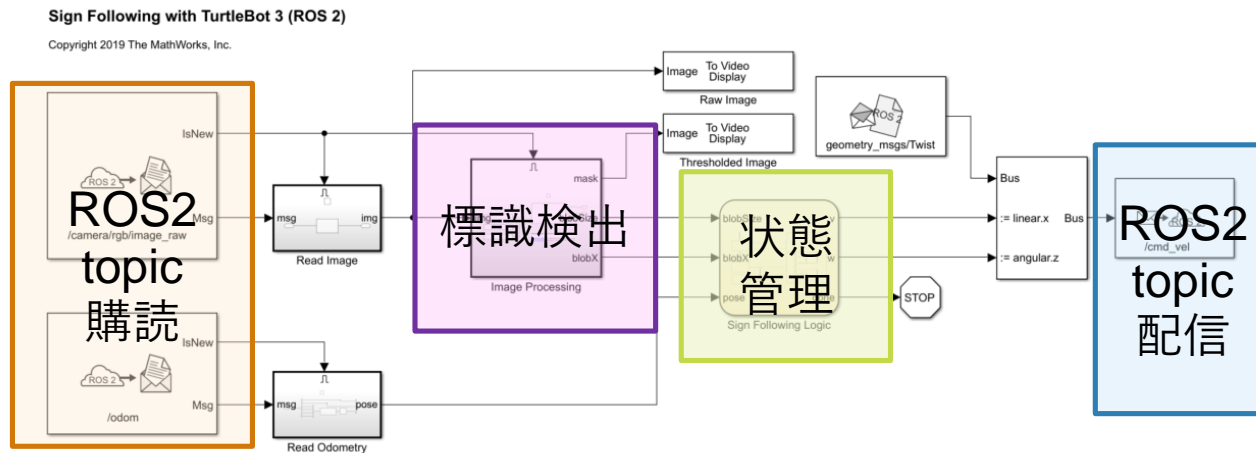
- ROS Toolboxの概要
- ユーザー事例
- ROS / ROS2とは？
- ROS / ROS2 連携のワークフロー
- ROS / ROS2 連携アプリケーションの例
- まとめ

アプリケーション例

移動ロボットによる標識追従

- 標識を識別し速度指令でTurtlebot3を制御
- Simulink と ROS 2 を使ってプロトタイピング
- GazeboからROS 2経由で画像やオドメトリを取得
- 速度指令をROS 2経由で配信
- C++ ROS2ノード生成可能(Gazeboや実機でテスト可能)

[Sign Following Robot with ROS in MATLAB](#)
[Sign Following Robot with ROS in Simulink](#)
[Sign Following Robot with ROS 2 in MATLAB](#)
[Sign Following Robot with ROS 2 in Simulink](#)

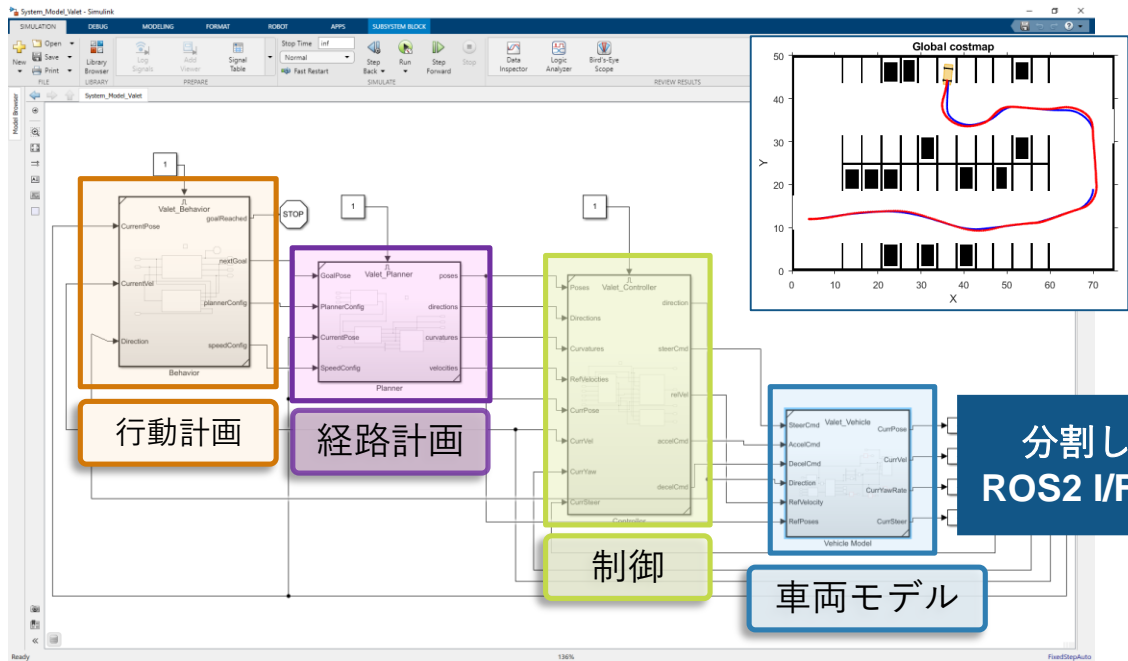


アプリケーション例

自動駐車

- 自動駐車をROS / ROS2ノードとして実装する例
- ROS 2 ネットワーク上でモデルをテスト
- C++ ROS 2ノードを生成し、高速実行が可能

- [Automated Parking Valet with ROS in MATLAB](#)
- [Automated Parking Valet with ROS in Simulink](#)
- [Automated Parking Valet with ROS 2 in MATLAB](#)
- [Automated Parking Valet with ROS 2 in Simulink](#)



各アルゴリズムがシリアル実行

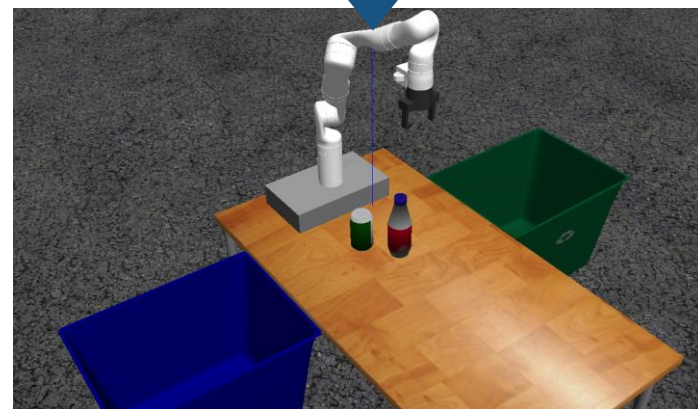
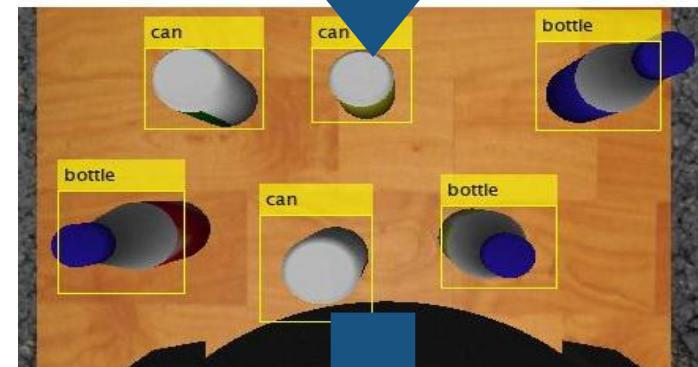


ROS 2 pub/subを追加し、個別ノードとして分散実行

アプリケーション例

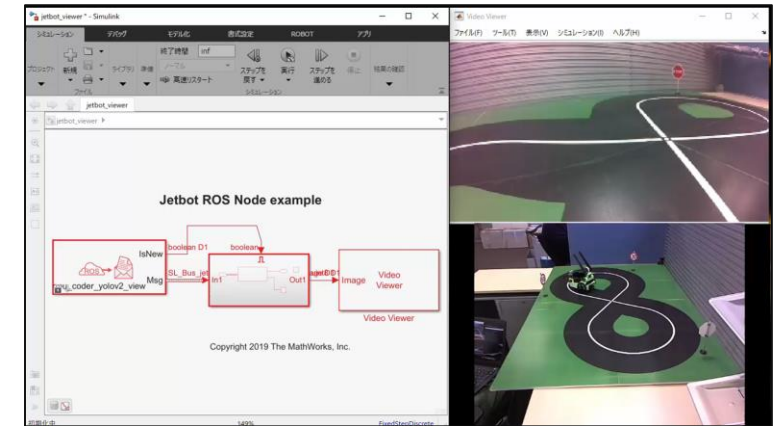
ロボットアームによるピック & プレイス

- ディープラーニングを使って
ペットボトルと缶を検出
- 青と緑のリサイクルボックスに分別する
- 物体検出
 - ディープラーニング(YOLO v2)を活用し、
シミュレーションのカメラ画像から物体検出
- タスクスケジューリング
 - Stateflow を使って
一連のロボットのタスクを実行
- パスプランニング
 - 補間ベースの軌道生成機能を活用
(`transformtraj` / `trapveltraj`)

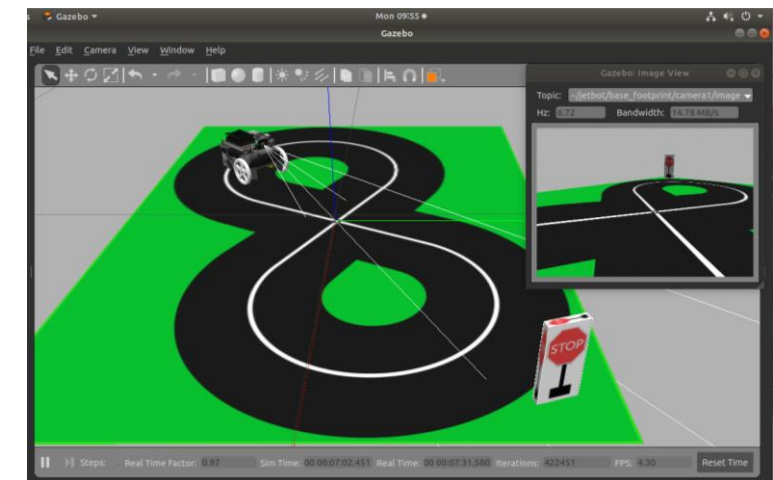


アプリケーション例

移動ロボットによるライントレースと標識検出



実機

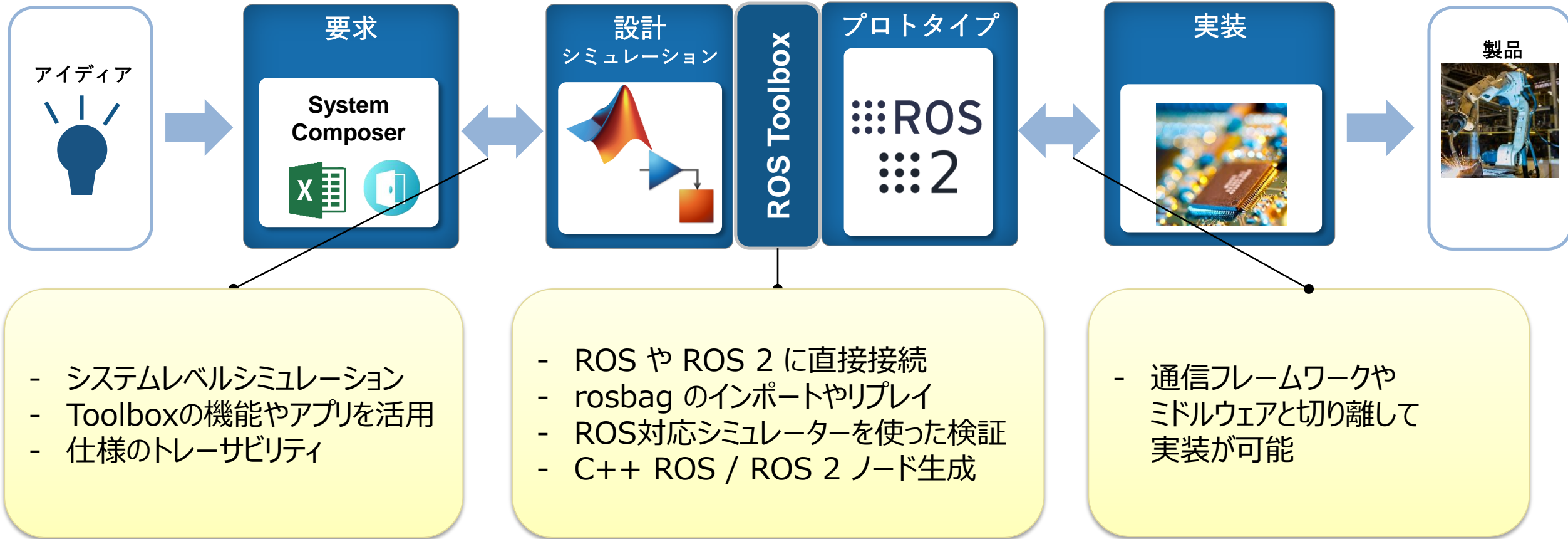


シミュレーション

アジェンダ

- ROS Toolboxの概要
- ユーザー事例
- ROS / ROS2とは？
- ROS / ROS2 連携のワークフロー
- ROS / ROS2 連携アプリケーションの例
- まとめ

まとめ: ROS Toolboxによるロボット開発ワークフロー



ROS / ROS 2 と連携し、効率的に自律システムを開発



© 2020 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.