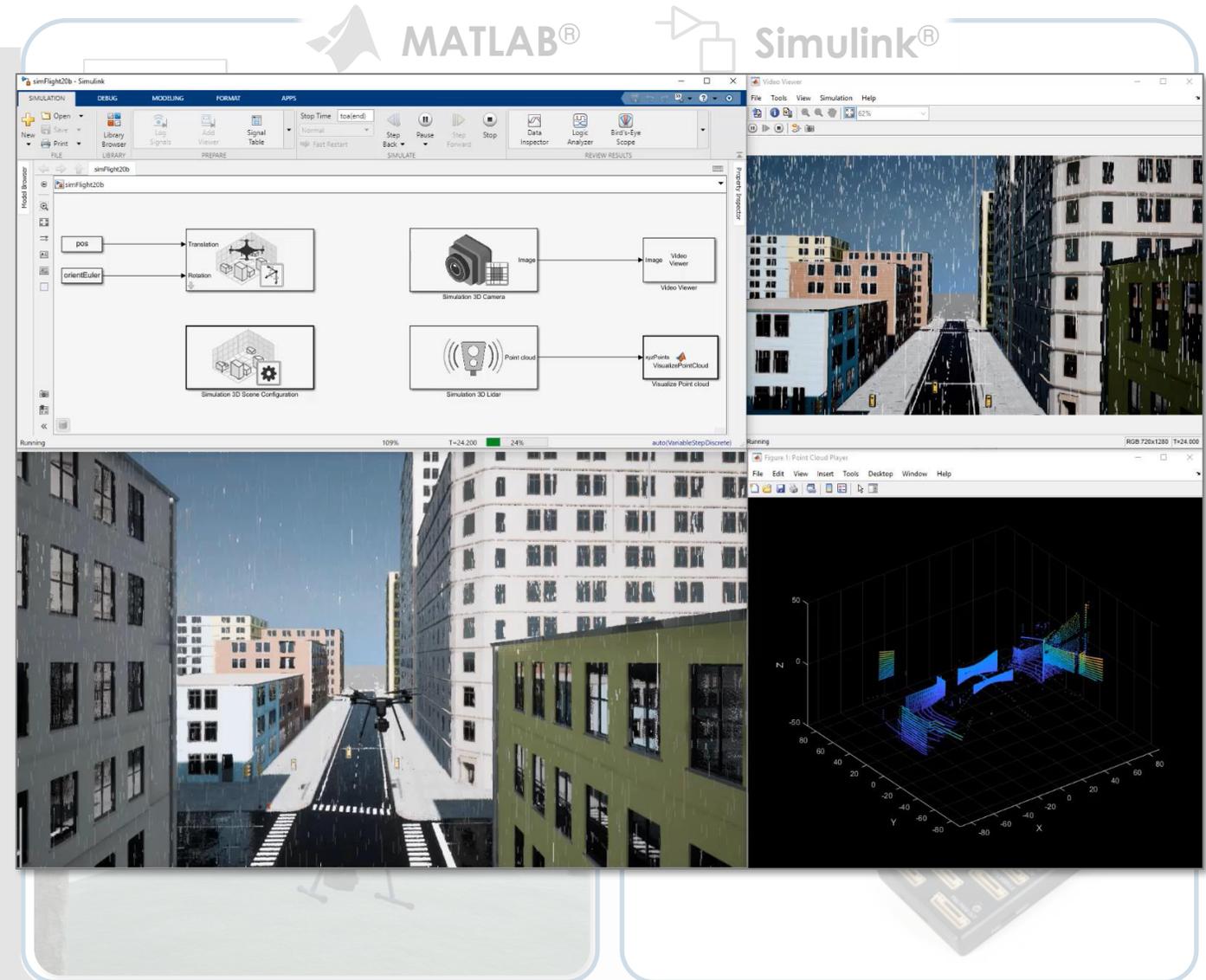


UAV Toolbox : MATLAB および Simulink による 自律無人航空機開発

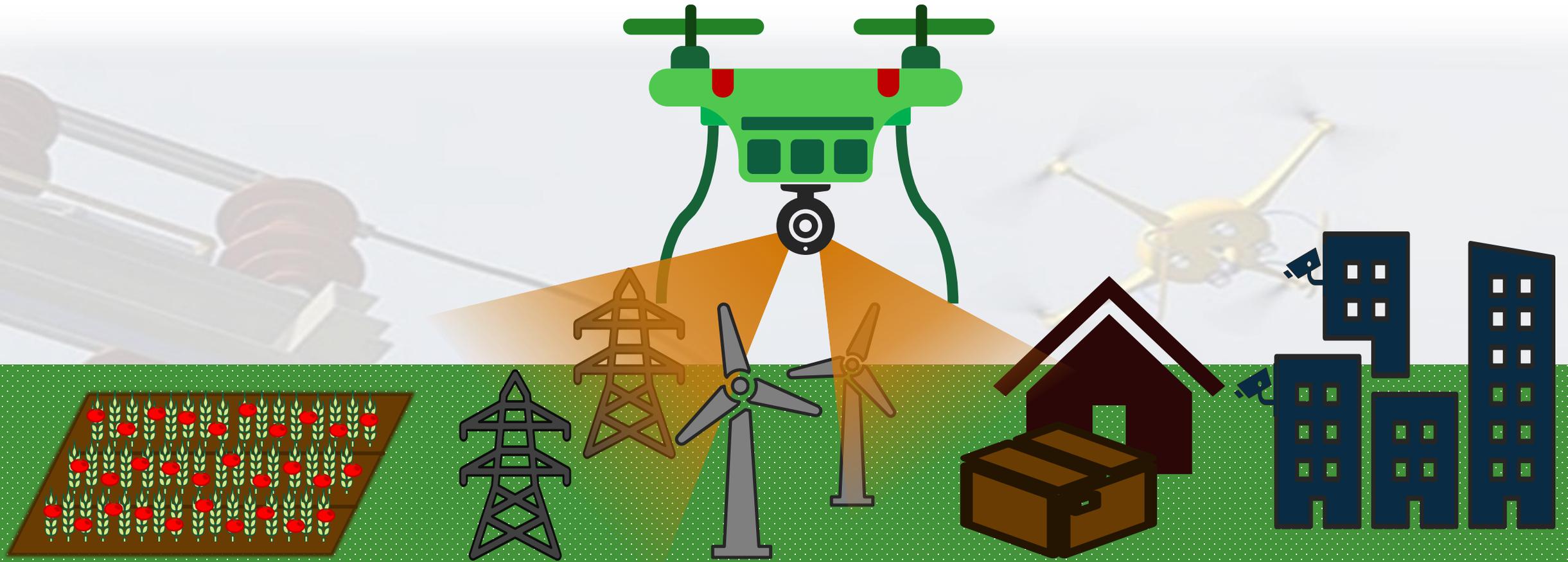
MathWorks Japan

自律無人航空機 (UAV)の開発と評価

- MATLAB および Simulink 統合開発環境
- 無人航空機システムと自律アプリケーションを設計するためのツール
- 無人航空機開発のために適切な手法を選択可能
- センサーモデルを含めた閉ループシミュレーションによりシステム全体を検証



自律無人航空機の利用シーンの増加



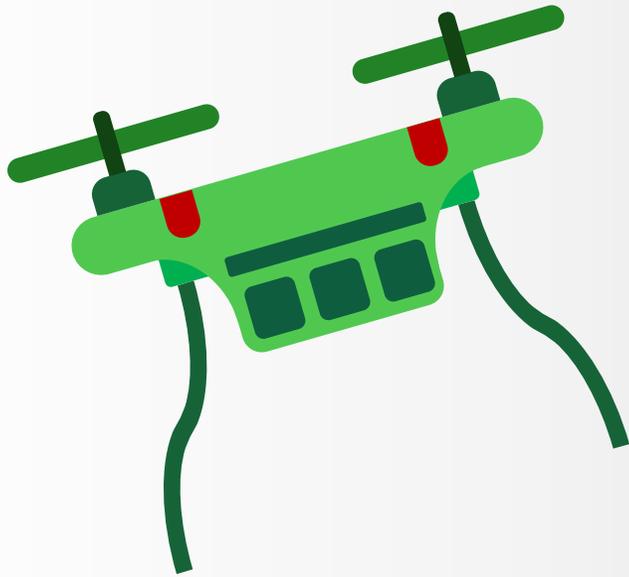
地図作成 & 調査

検査 & モニタリング

配達 & 輸送

セキュリティ

自律無人航空機システムとアプリケーション開発における課題



複雑かつ先進的な
自律アルゴリズムの開発

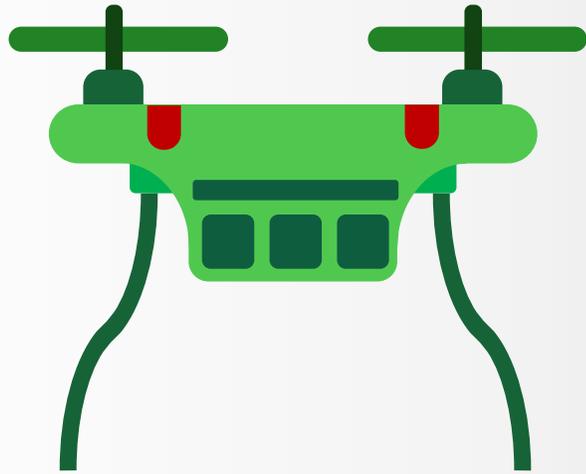


エンド・ツー・エンドの
ワークフロー



システムの性能保証と
飛行リスクの低減

自律無人航空機システムとアプリケーション開発のソリューション



無人航空機システムと
自律アルゴリズムを
設計検証するための強力な機能

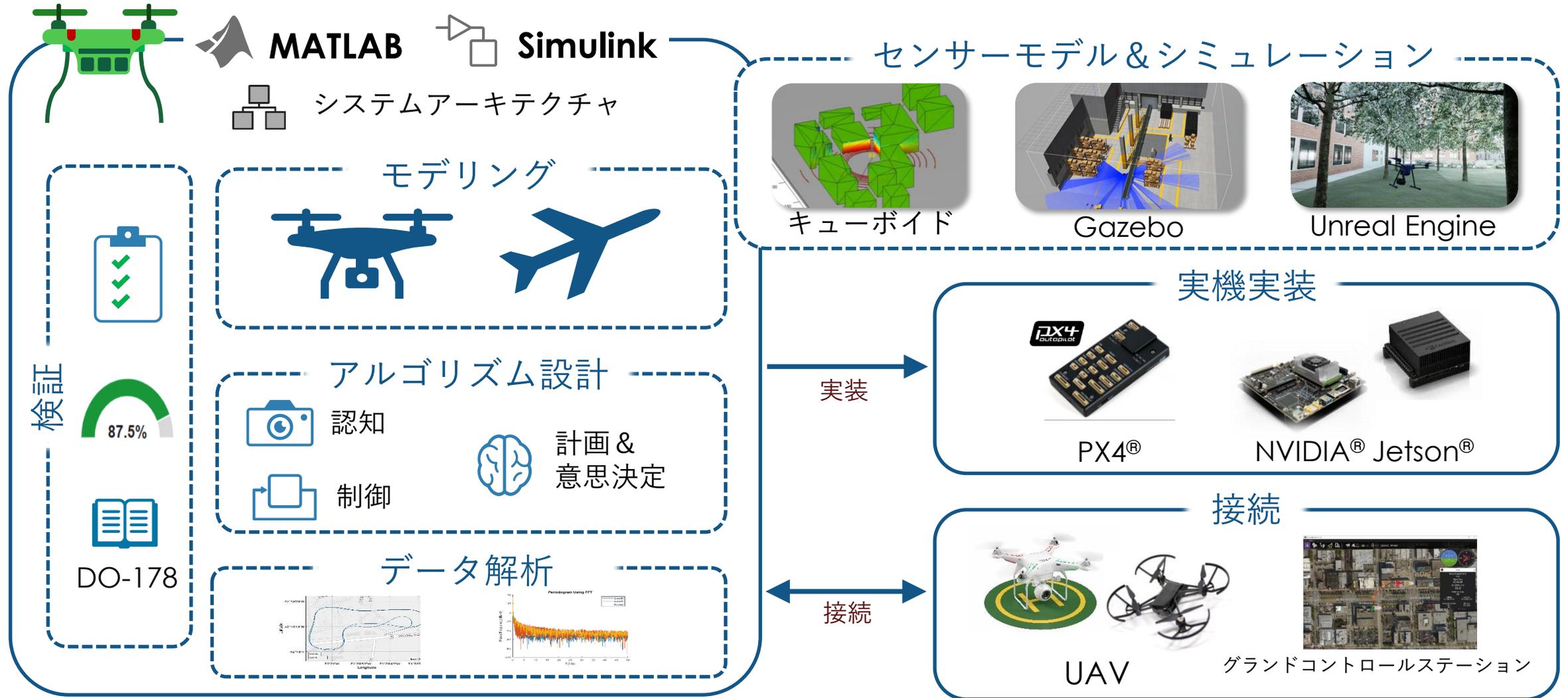


アイデアから製品開発までを
カバーする統合開発環境



仮想飛行テストを通じて
テスト・検証ツールへの拡張が可能

無人航空機アプリケーションを開発するための統合ワークフロー



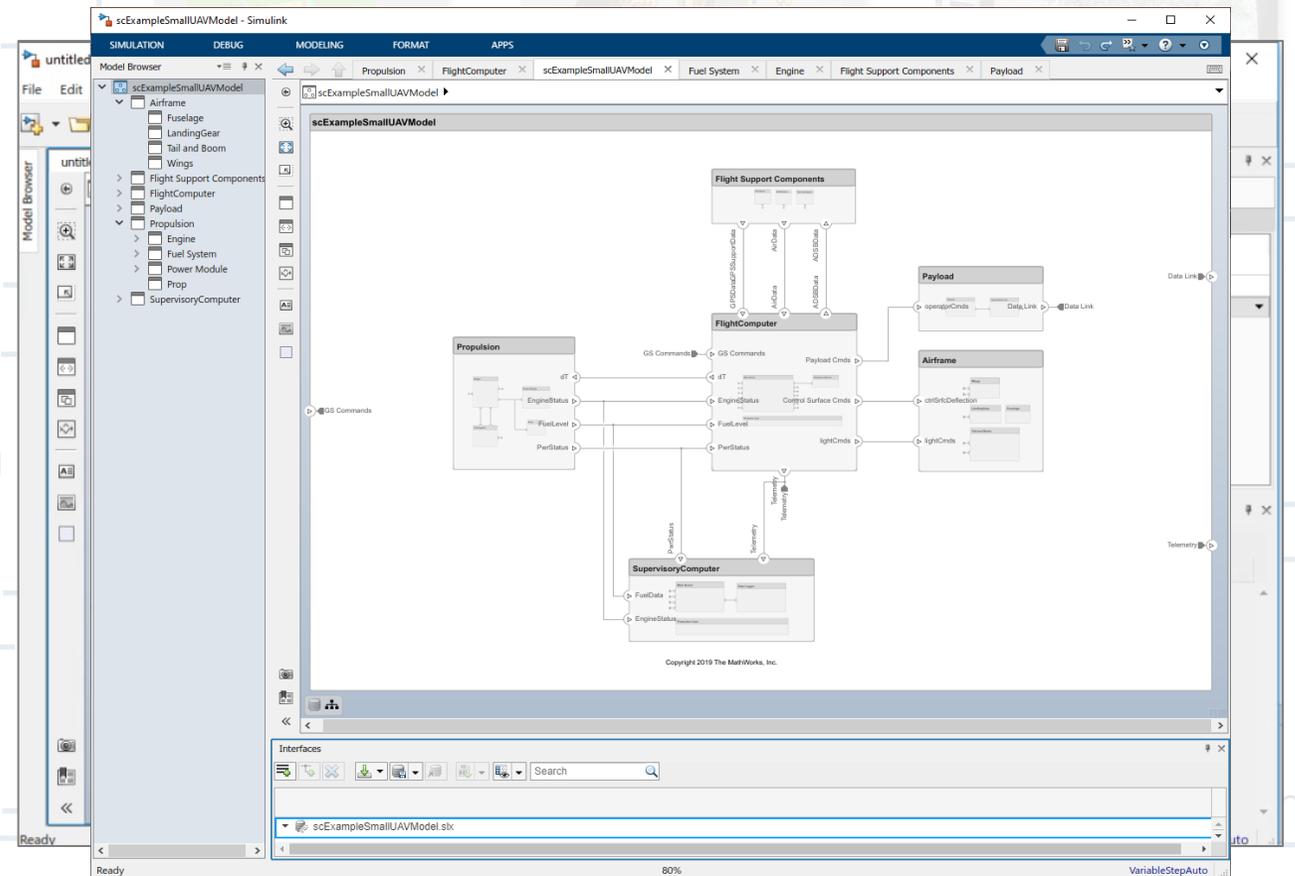
無人航空機アプリケーションを開発するための統合ワークフロー

システムアーキテクチャ

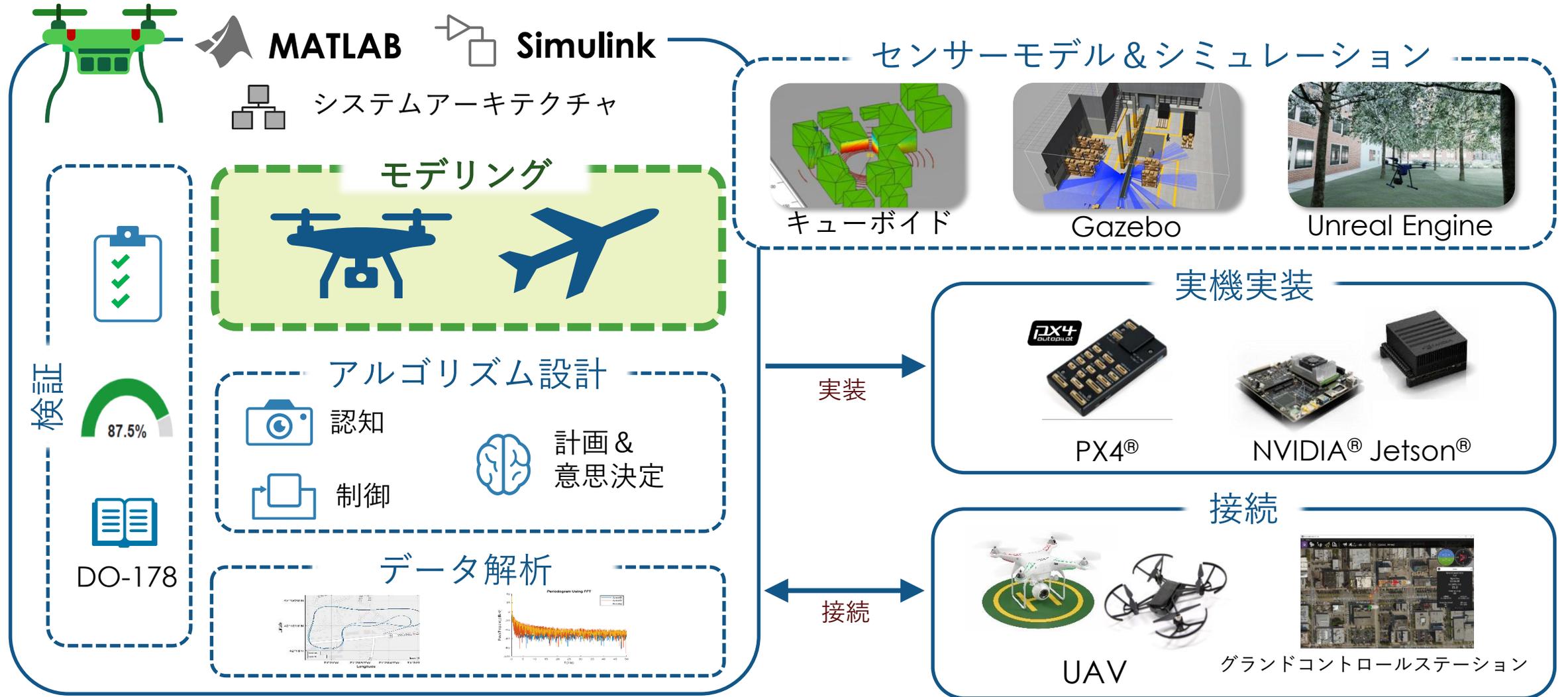
- システムとソフトウェアアーキテクチャを設計解析するための System Composer™

- トレーサビリティのために Simulink と要求仕様をリンク

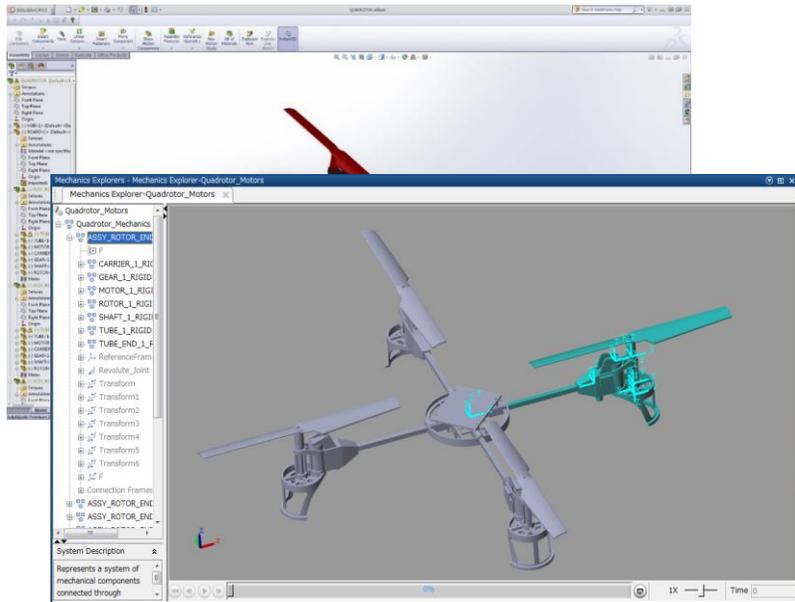
[Link](#)



無人航空機アプリケーションを開発するための統合ワークフロー



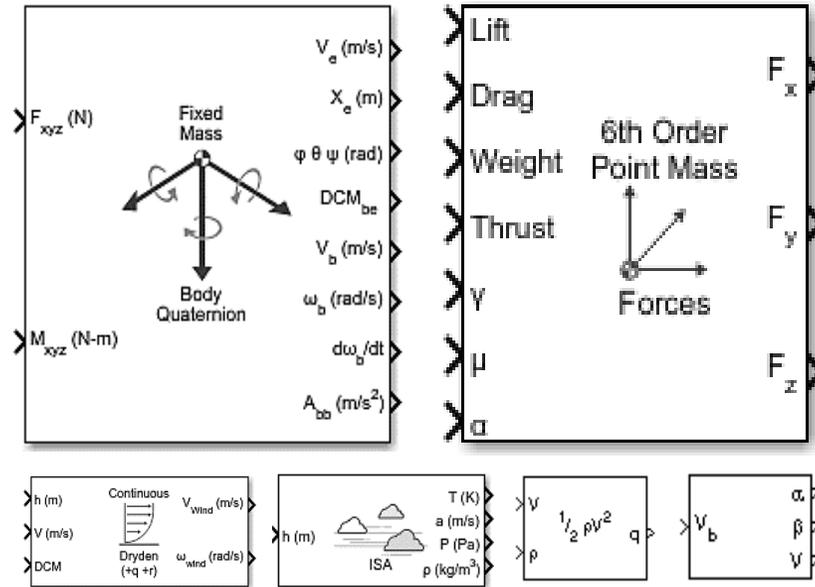
無人航空機のプラントモデリング:適切な詳細度を選択



Physical Modeling

[Link](#)

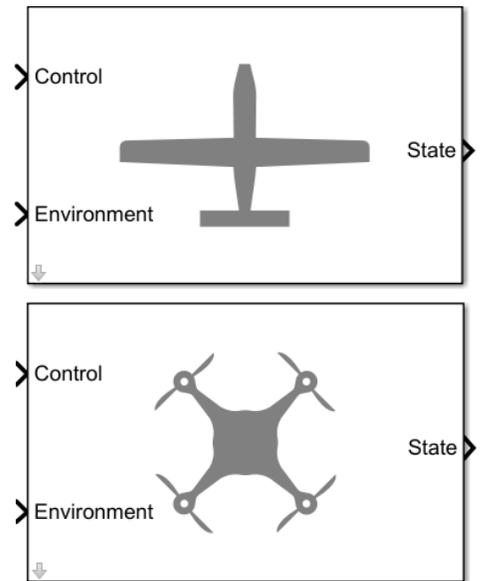
Model construction techniques and best practices, domain-specific modeling, physical units



Vehicle Dynamics

[Link](#)

Model aerodynamics, propulsion, and motion of aircraft and spacecraft



Guidance Model [Link](#)

Reduced-order model for UAV

近似モデルから高詳細度モデルまでの遷移

Live Editor - C:\Users\Ron\Documents\MATLAB\Examples\R2021a\shared_uav_aeroblocks\UAVFidelityExample\UAVFidelityExample.mlx

UAVFidelityExample.mlx

Transition From Low to High Fidelity UAV Models in Three Stages

Simulation models for UAVs often need different levels of fidelity during different development stages. A system designer may get incrementally better access to UAV characteristics as the design progresses.

In the [Approximate High Fidelity UAV Model with UAV Guidance Model Block](#) example, you tune a guidance block to match the characteristics of a high-fidelity fixed-wing aircraft. However, some more advanced modeling parameters may not be available at that point when the design is in progress. This can include sensor models, complete aerodynamics modelling and actuator dynamics.

This example shows how to design a medium-fidelity model using *aerodynamic coefficients*, *thrust curves*, and *response time specifications*. Assuming the high-fidelity model is unavailable until the end of the design process, this medium-fidelity model enables you to test your path planner and design a mid-level controller without needing the complexity of the high-fidelity model.

When the high-fidelity model is made available, you can then model the additional effects and study the changed system response. Compare the medium-fidelity model with a complex high fidelity system over a desired set of waypoints. This example demonstrates that the medium-fidelity model provided an accurate estimate of the UAV trajectory and step response.

Table of Contents

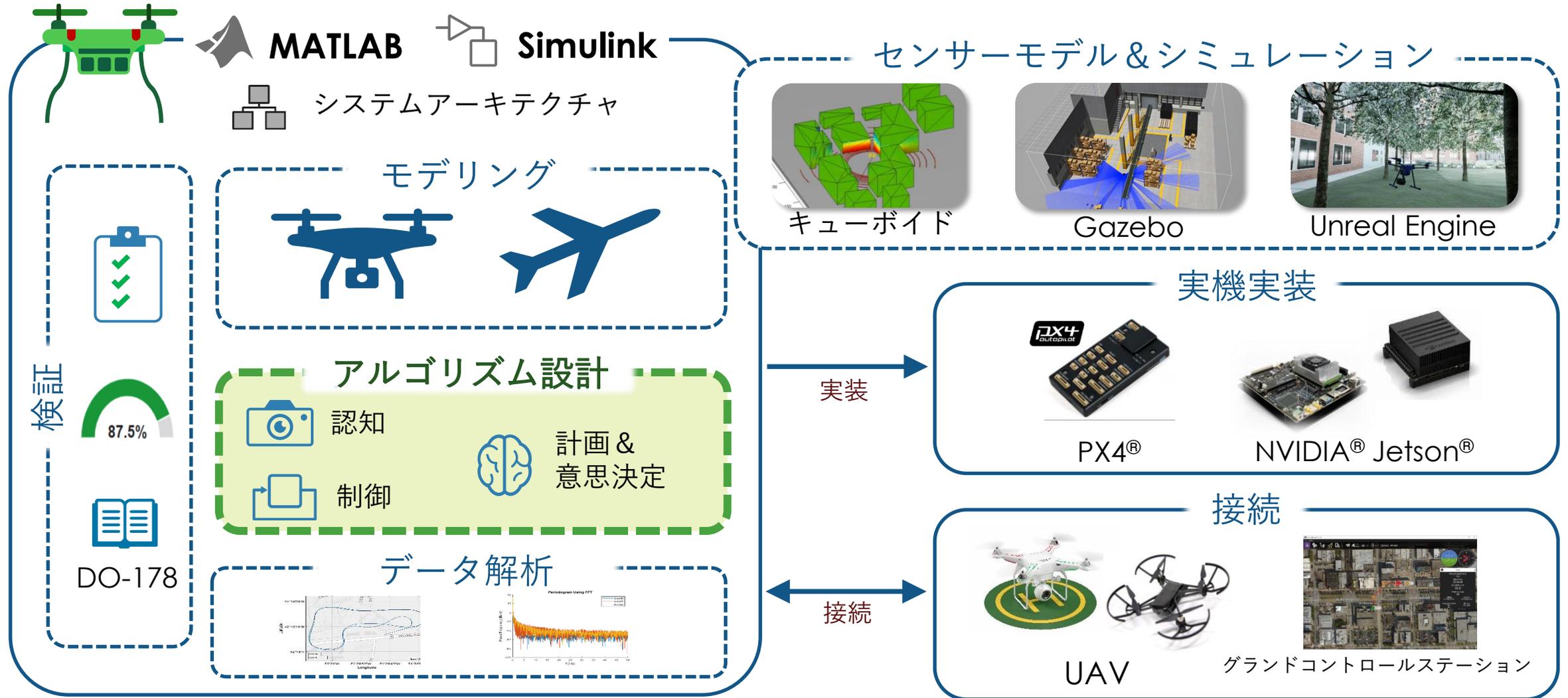
- [Open Example and Project Files](#)
- [Low Fidelity Model](#)
- [Medium Fidelity Model](#)
- [Medium Fidelity Step Response](#)
- [Simulate Path Following Algorithm](#)
- [High Fidelity Step Response](#)
- [Simulate Path Following Algorithm for High-Fidelity](#)
- [Conclusion](#)

Open Example and Project Files

To access the example files, click **Open Live Script** or use the `openExample` function.

UTF-8 LF script

無人航空機アプリケーションを開発するための統合ワークフロー



自律無人航空機のアルゴリズム設計

アルゴリズム設計



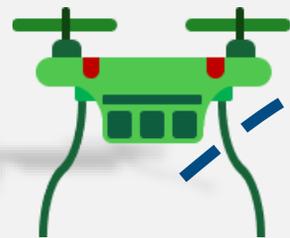
認知



制御



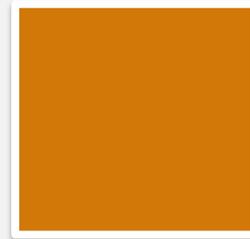
計画 &
意思決定



開始地点



地図データ



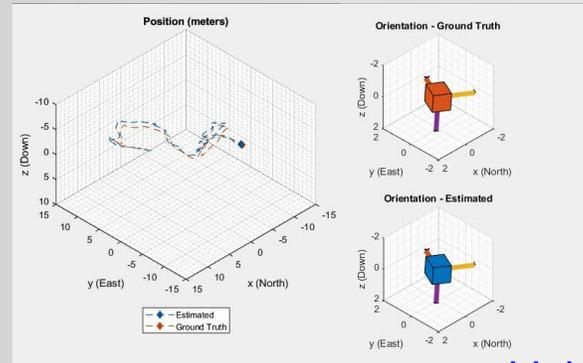
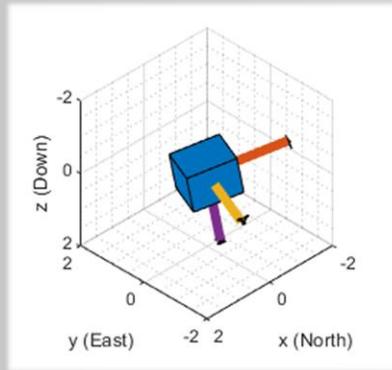
計画された経路

終了地点

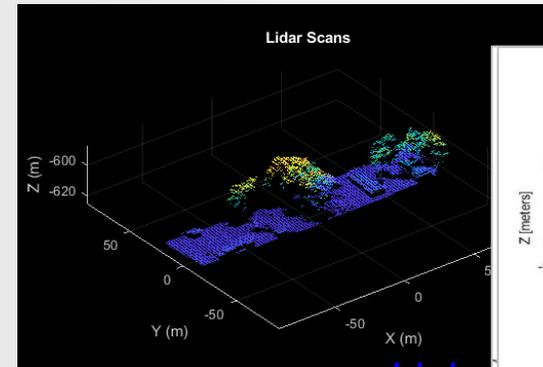
強力な専用ライブラリを活用し自律アルゴリズム設計



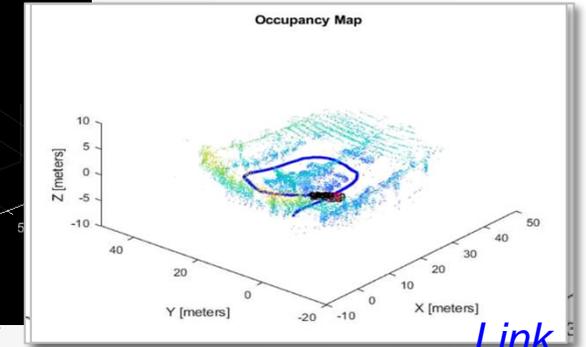
自己認知



[Link](#)

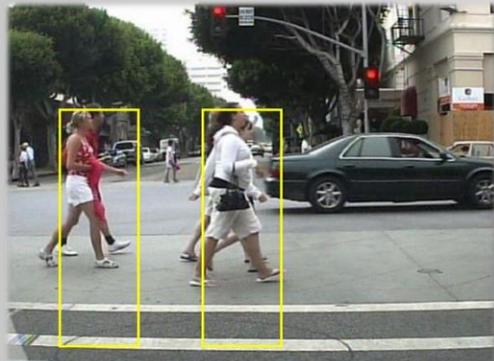


[Link](#)

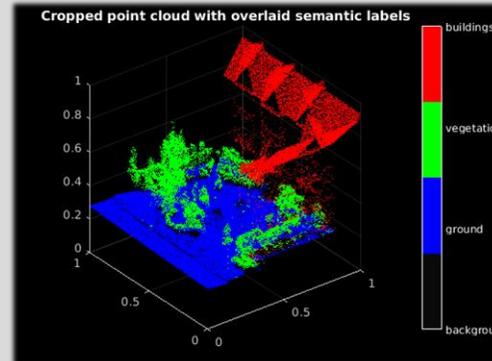


[Link](#)

周辺認知

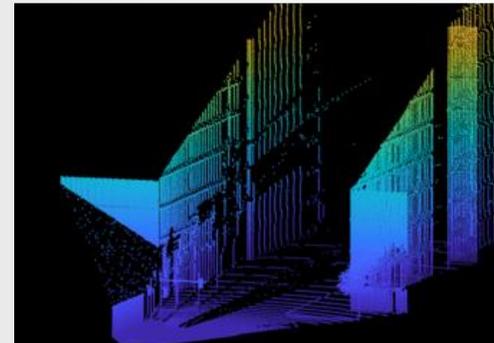


[Link](#)

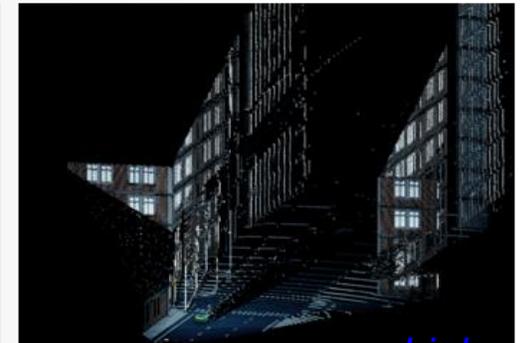


[Link](#)

Lidar point cloud data



Fused lidar and camera data

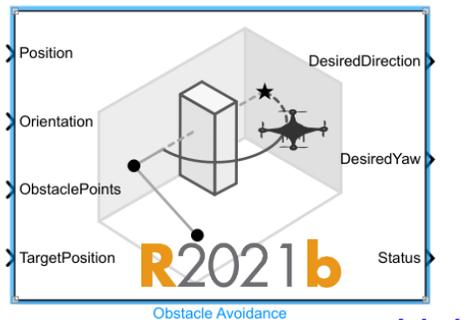
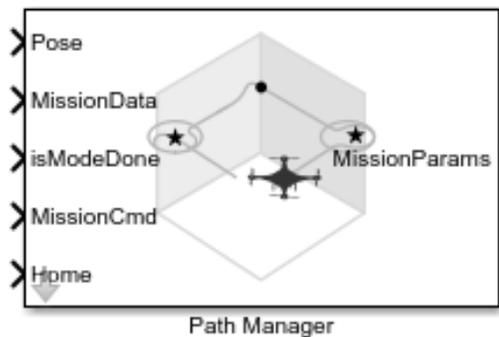
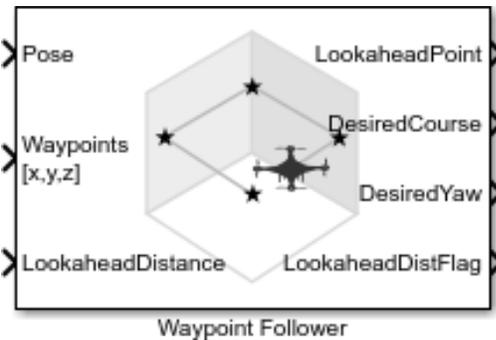
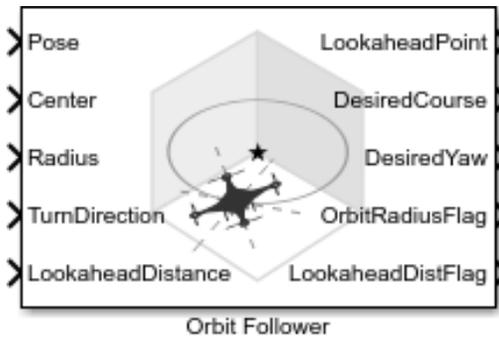


[Link](#)

強力な専用ライブラリを活用し自律アルゴリズム設計

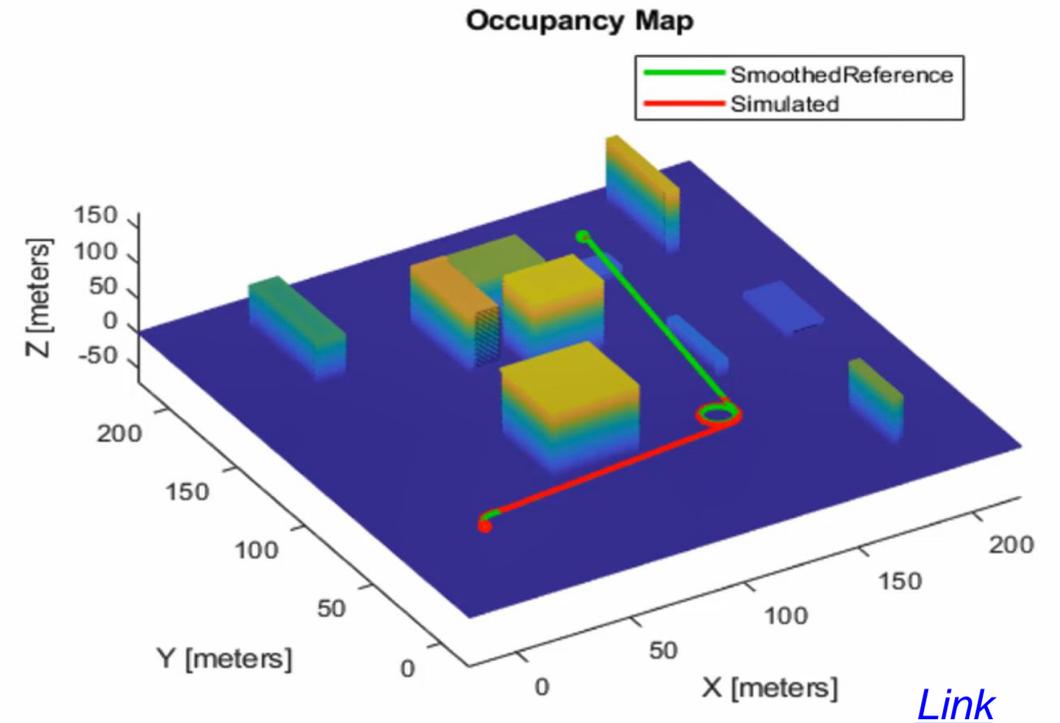


計画 & 意思決定



[Link](#)

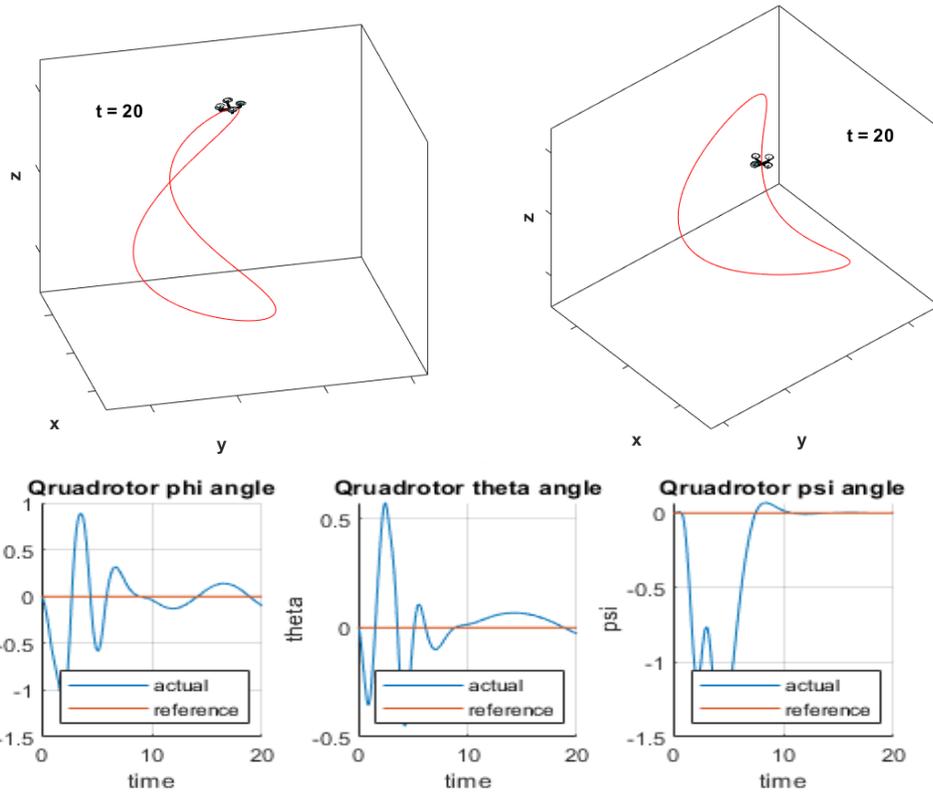
ウェイポイントによるミッションの定義と軌道追従アルゴリズム



[Link](#)

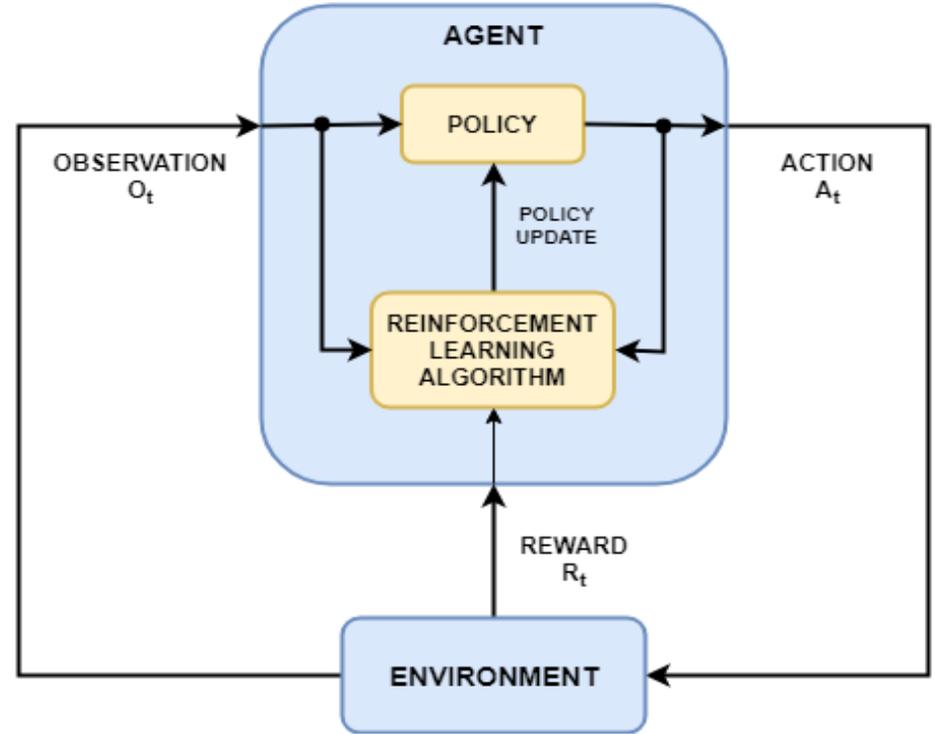
アドバンストなプランナーを活用したモーションプランニング

強力な専用ライブラリを活用し自律アルゴリズム設計



[Link](#)

非線形モデル予測制御(MPC)による軌道追従コントローラ



[Link](#)

強化学習アルゴリズムによる軌道生成のためのポリシーの獲得

例: RRTによる固定翼無人機のモーションプランニング

The screenshot displays the MATLAB R2021a Live Editor environment. The central workspace shows a 3D plot titled "Occupancy Map" with axes X [meters], Y [meters], and Z [meters]. The plot compares two paths: a "SmoothedReference" path (green line) and a "Simulated" path (red line). The smoothed path is significantly shorter and more direct than the simulated path, which exhibits a large loop. A legend in the top right of the plot area identifies the two paths.

Below the plot, the following text is displayed:

The smoothed path is much shorter and shows improved tracking overall.

The Live Editor window shows the following code snippet:

```

[xENU,yENU,zENU] = exampleHelperSimulateUAV(waypoints,ss.AirSpeed,timeToReachGoal);
hSimulated = plot3(xENU,yENU,zENU,"LineWidth",2,"Color","r");

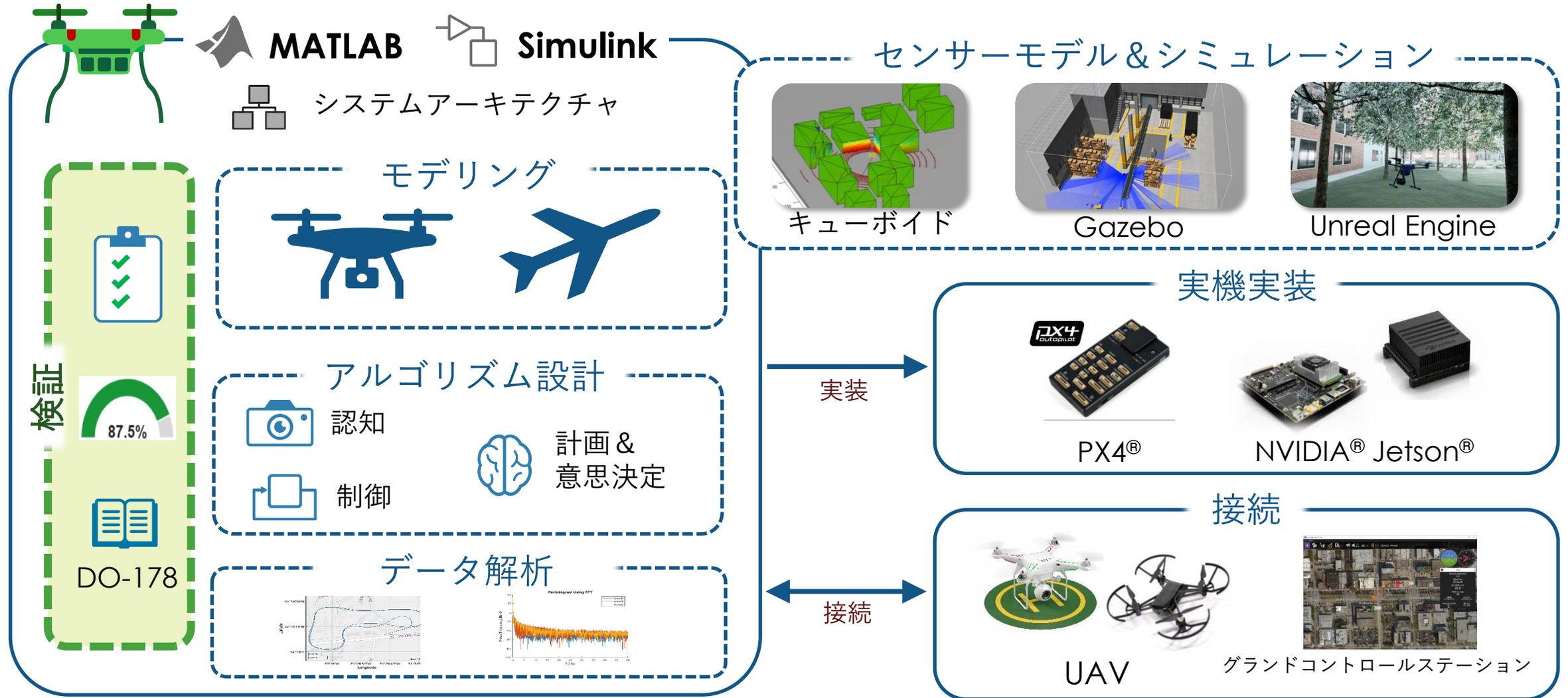
legend([hReference,hSimulated],"smoothedReference","Simulated","Location","best")
hold off
view([-31 63]);
end

```

The Workspace window on the right lists the following variables and their values:

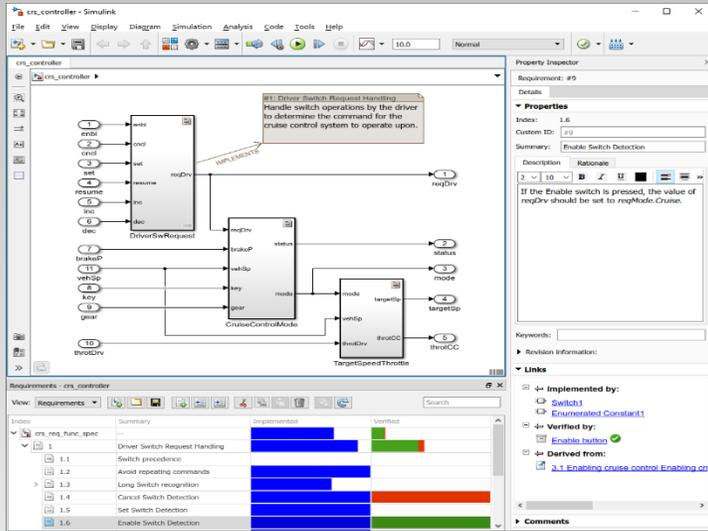
Name	Value
goalPose	[150,180,35,1.5708]
hMap	1x1 Axes
hReference	1x1 Line
hSimulated	1x1 Line
interpolatedPathObj	1x1 navPath
interpolatedSmoothW...	1x1 navPath
mapData	1x1 struct
omap	1x1 occupancyMap3D
planner	1x1 plannerRRT
pthObj	1x1 navPath
smoothWaypointsObj	1x1 navPath
solnInfo	1x1 struct
ss	1x1 ExampleHelperU...
startPose	[12,22,25,1.5708]
sv	1x1 validatorOccup...
threshold	[149.5000,150.5000;17...
timeToReachGoal	54.2273
waypoints	1000x4 double
xENU	1x936 double
yENU	1x936 double
zENU	1x936 double

無人航空機アプリケーションを開発するための統合ワークフロー

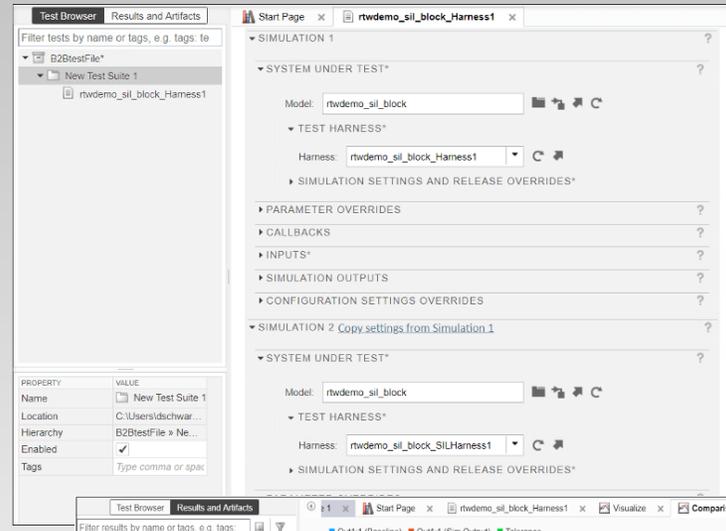


検証の自動化とトラッキング

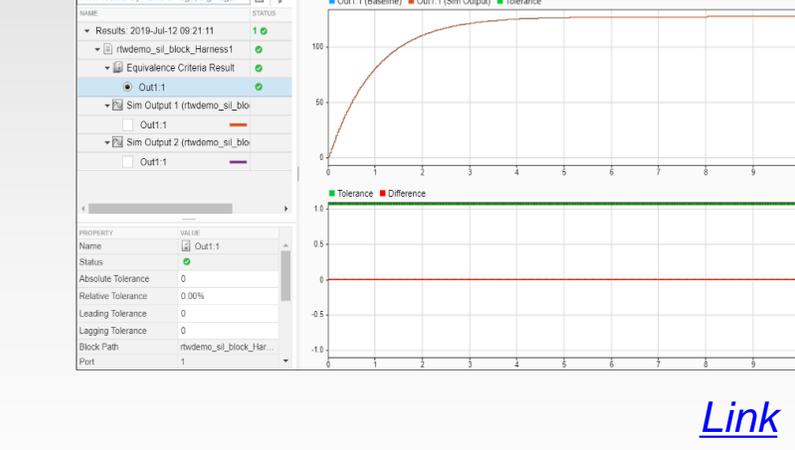
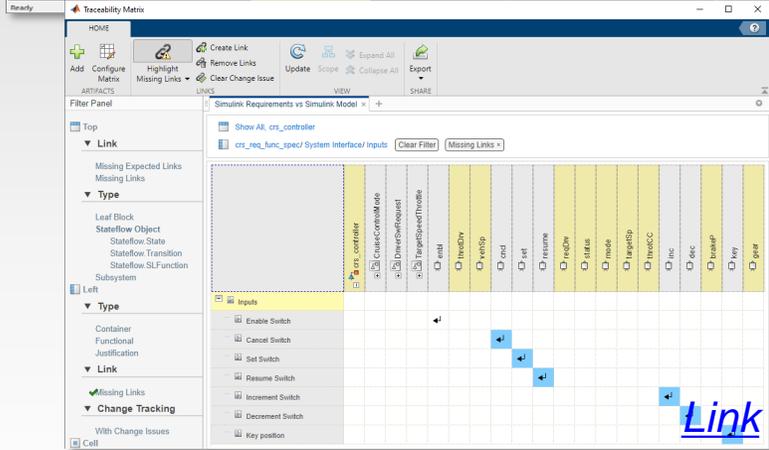
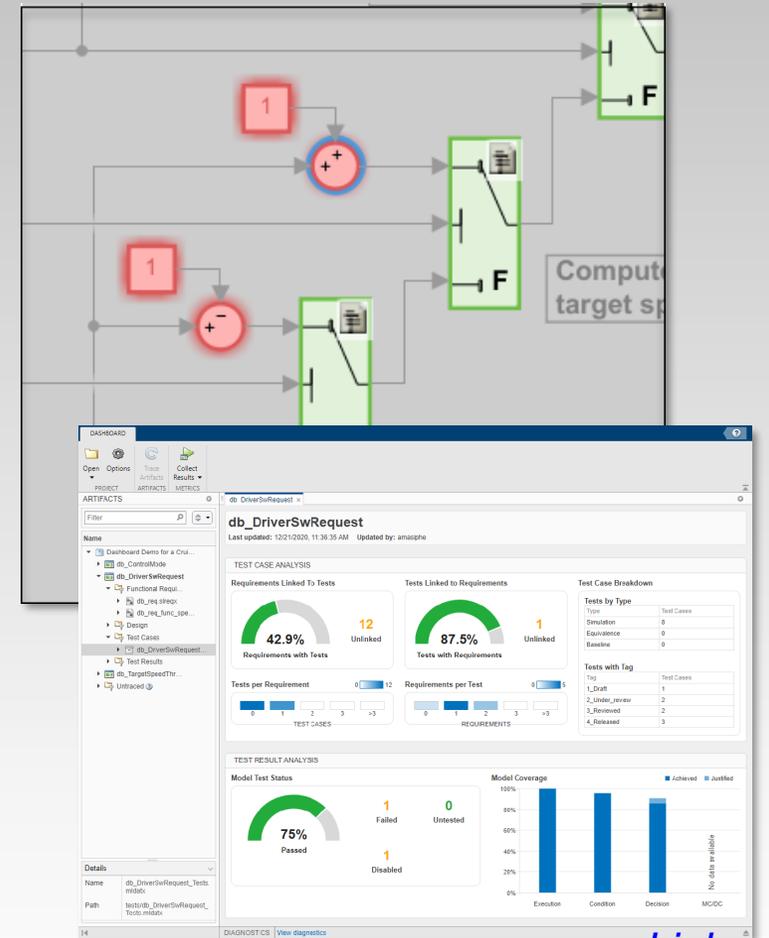
要求のトレーサビリティ



テストケースの管理と自動化



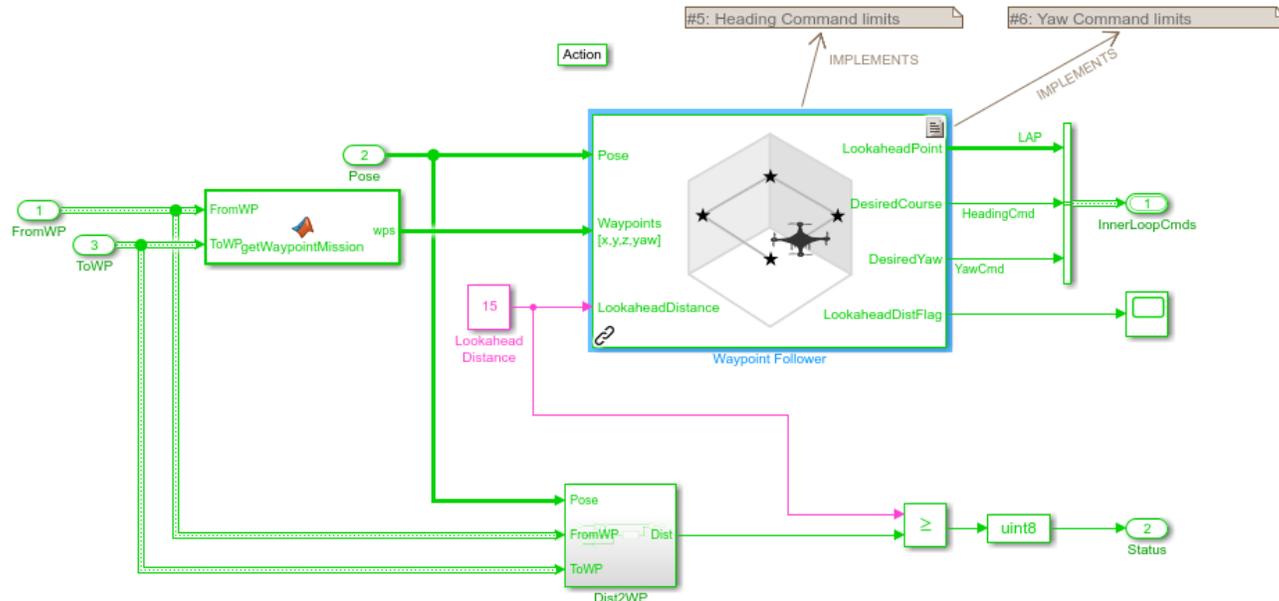
検証カバレッジ



[Link](#)

[Link](#)

例: 要求のリンクとアルゴリズムのテスト自動化



トレーサビリティのための要求リンク

NAME	STATUS
Results: 2020-Dec-21 15:19:38	3 ✓
MulticopterModelTest	3 ✓
GuidanceLogicSuite	1 ✓
Test Guidance State Transiti	✓
ControlSystemSuite	2 ✓
Test Control System Cmd Lin	✓
Test Control System Perform	✓

テストの自動実行と結果の評価

Index	Summary
1	Flight Controller
1.1	Guidance
1.1.1	Heading Command limits
1.1.2	Yaw Command limits
1.1.3	Not start until flight is started
1.1.4	Transition to Descend
1.1.5	Finishing Flight
1.1.6	Transition out of Takeoff
1.2	Control
1.2.1	Control Commands Limits

Requirement: #19

Type: Functional

Index: 1.2.1.4

Custom ID: #19

Summary: Pitch Command Limits

Description: Commanded Pitch angle shall not exceed 25 degrees in either direction

Index	Summary	Implemented	Verified
1	Flight Controller	✓	✓
1.1	Guidance	✓	✓
1.1.1	Heading Command limits	✓	✓
1.1.2	Yaw Command limits	✓	✓
1.1.3	Not start until flight is started	✓	✓
1.1.4	Transition to Descend	✓	✓
1.1.5	Finishing Flight	✓	✓
1.1.6	Transition out of Takeoff	✓	✓
1.2	Control	✓	✓
1.2.1	Control Commands Limits	✓	✓
1.2.1.1	Velocity Command Limits	✓	✓
1.2.1.2	Lateral Acceleration Command Limits	✓	✓
1.2.1.3	Vertical Acceleration Command Limits	✓	✓
1.2.1.4	Pitch Command Limits	✓	✓

Requirement: #25

Type: Functional

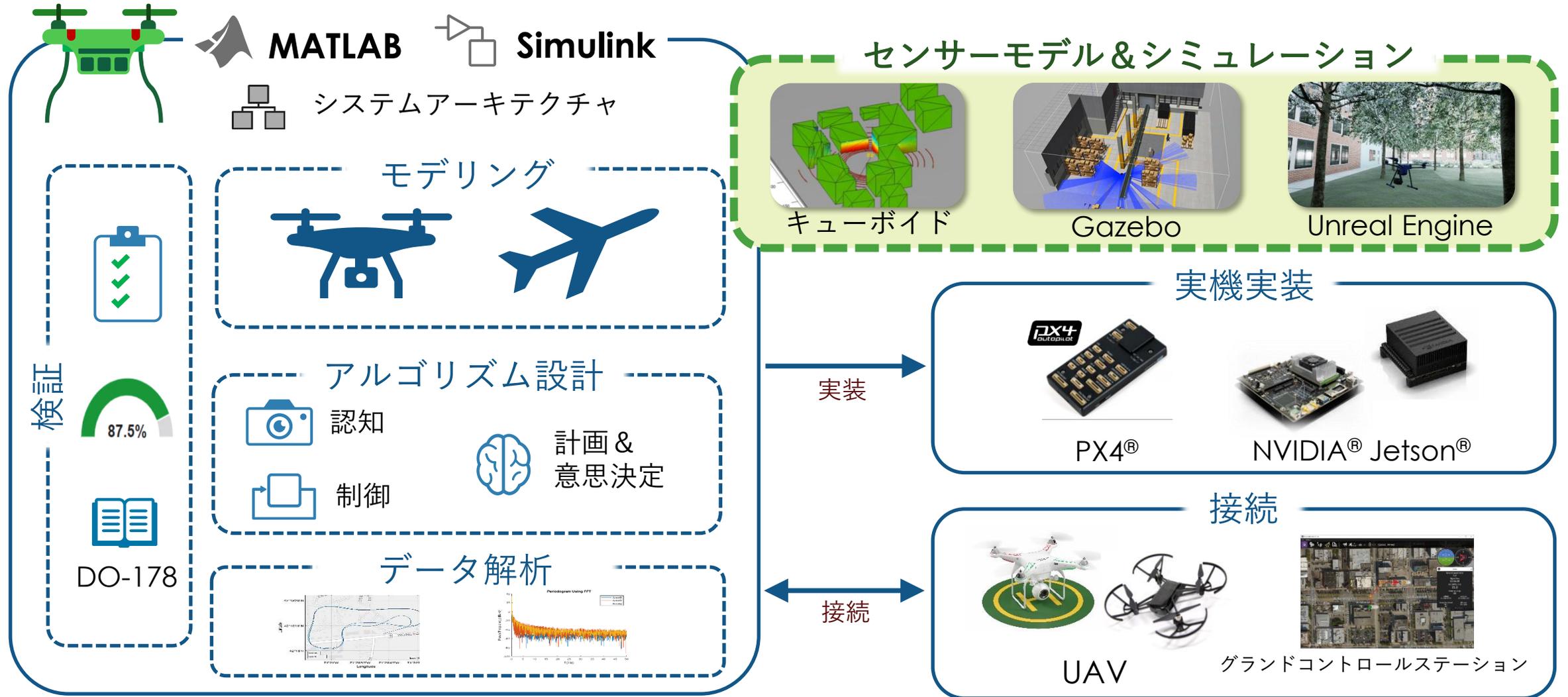
Index: 1.2.2.1

Custom ID: #25

Summary: Lateral Position Steady State

Description: Vehicle shall reach a steady state within 50 cm of its lateral (XY) commanded position

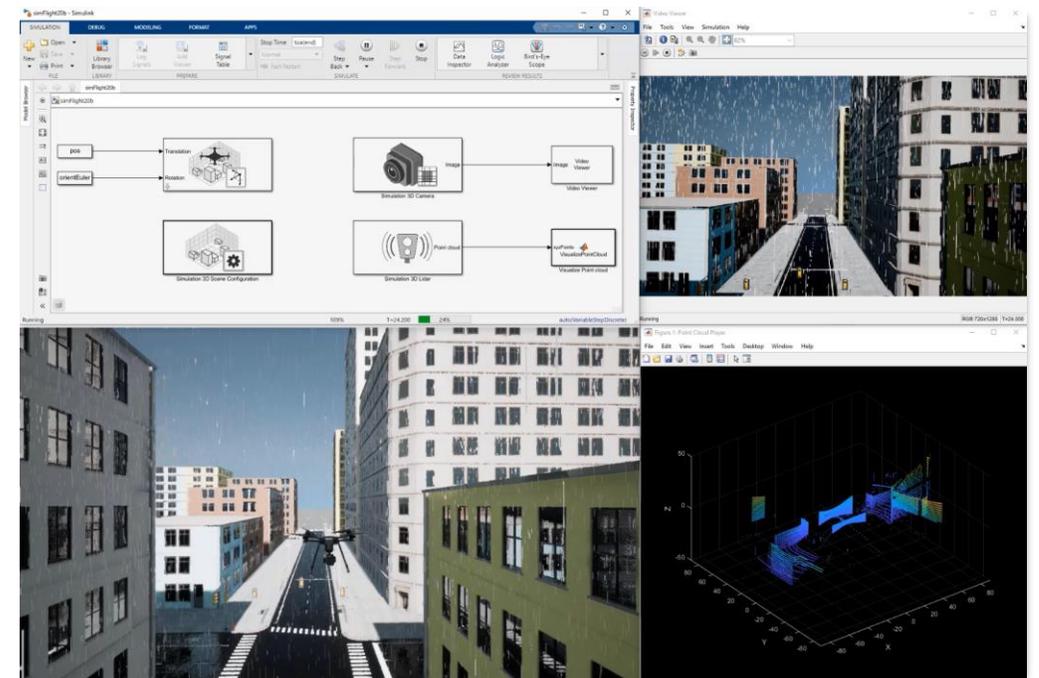
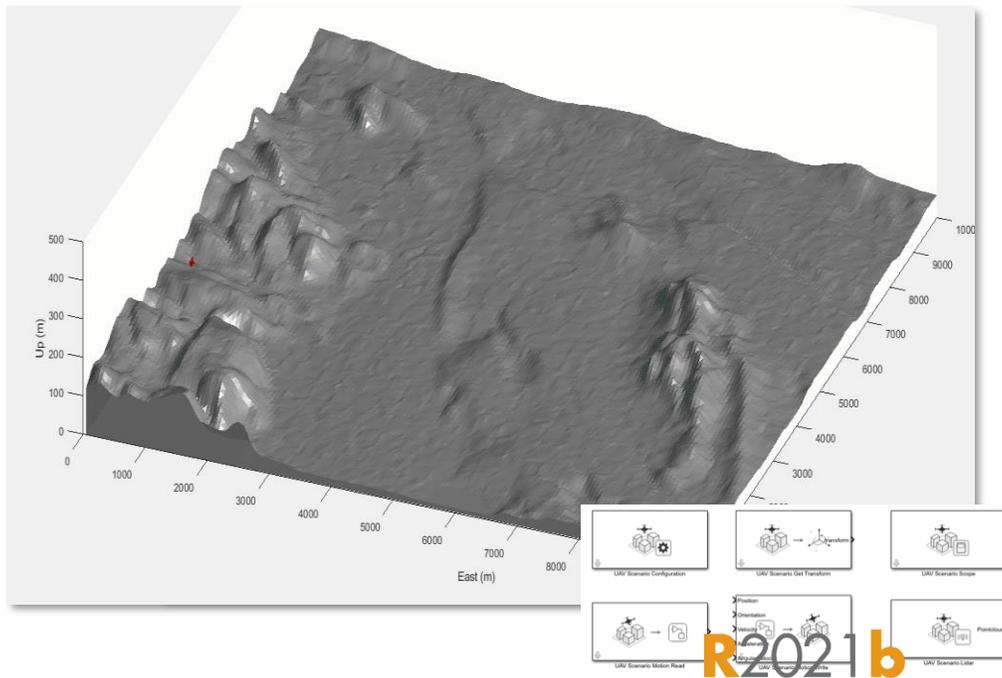
無人航空機アプリケーションを開発するための統合ワークフロー



センサーモデルと統合されたシミュレーション

キューボイド
高速

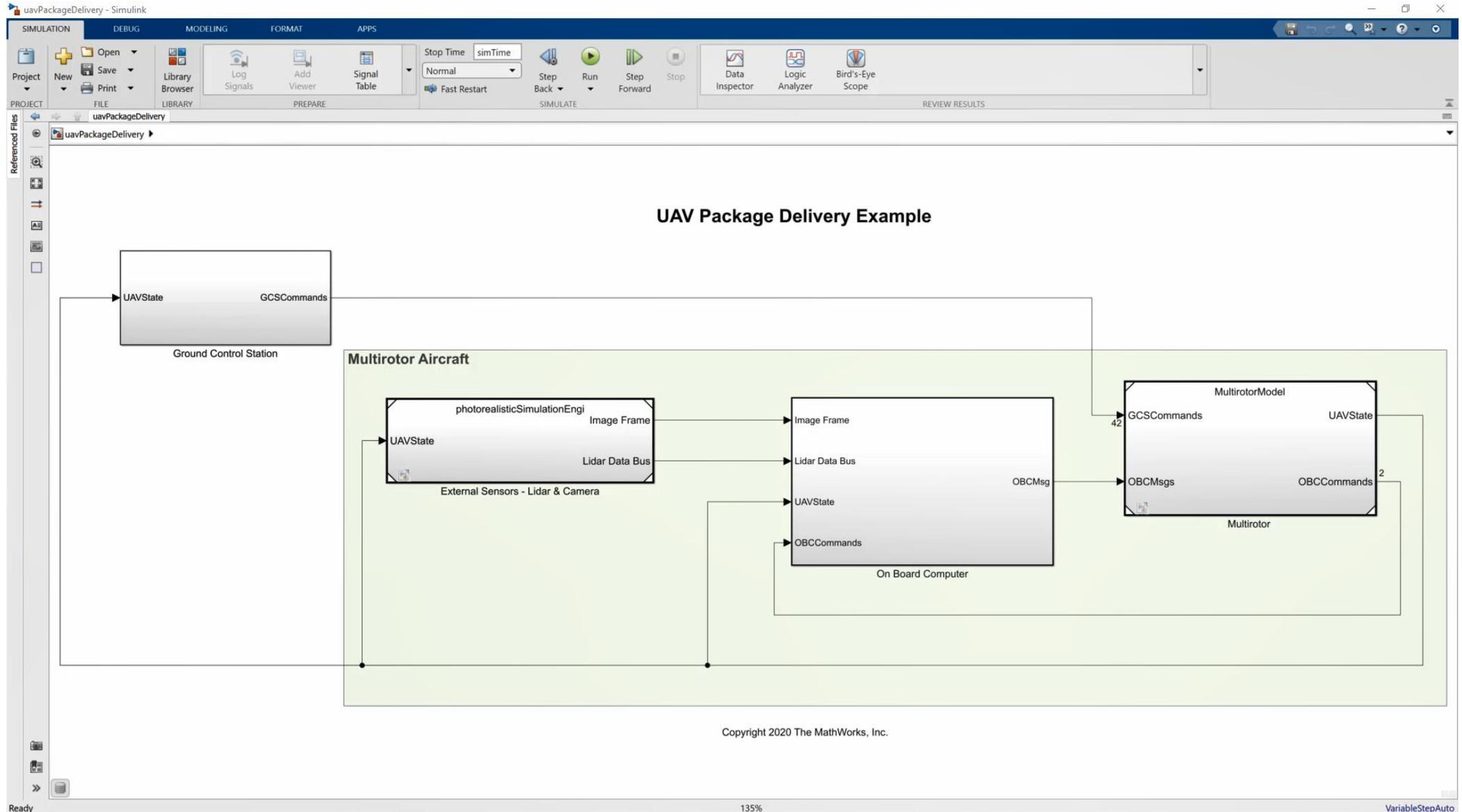
Unreal Engine®
フォトリアリスティック



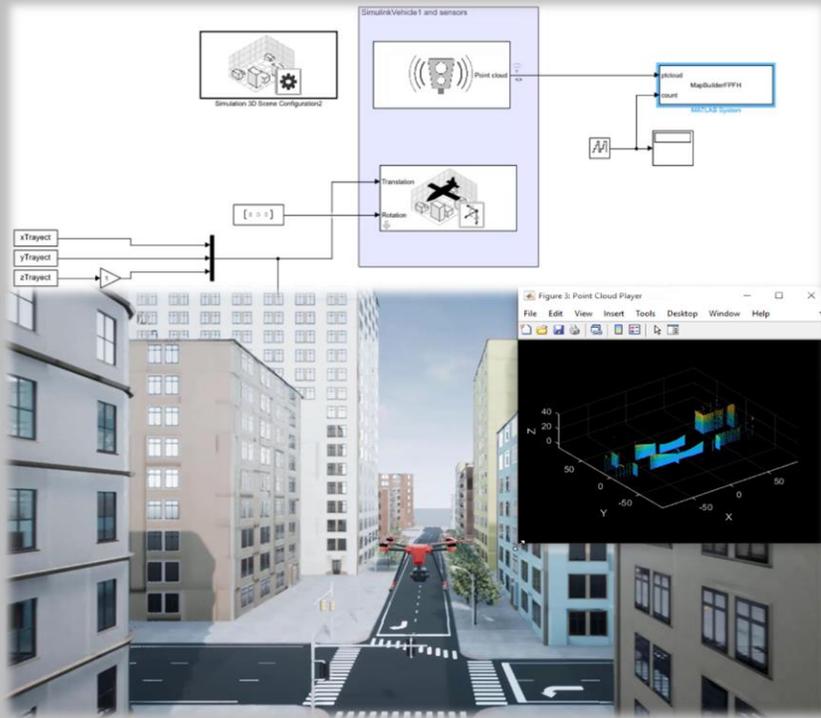
素早くシナリオを作成し、
センサーデータを生成

閉ループシミュレーションで
テストするためのリアルな環境

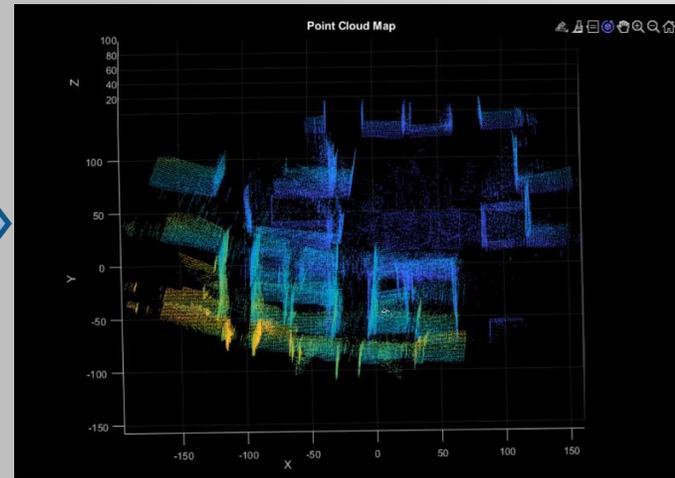
センサーモデルも含めた統合シミュレーション



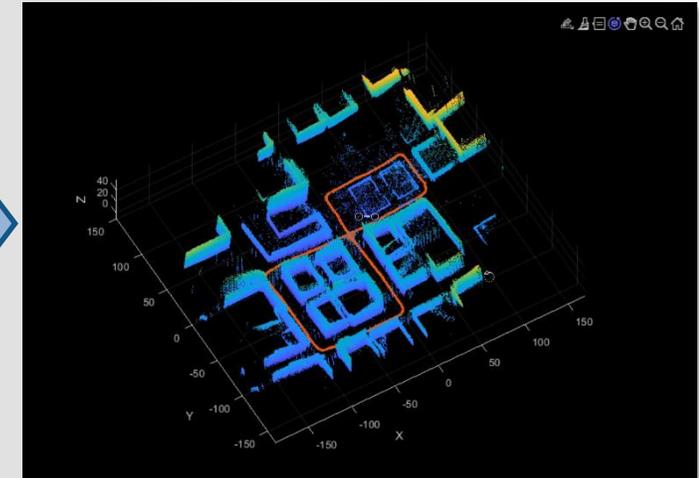
例: Lidarをシミュレーションし、3Dの地図を構築



シミュレーションによる
センサーデータ取得

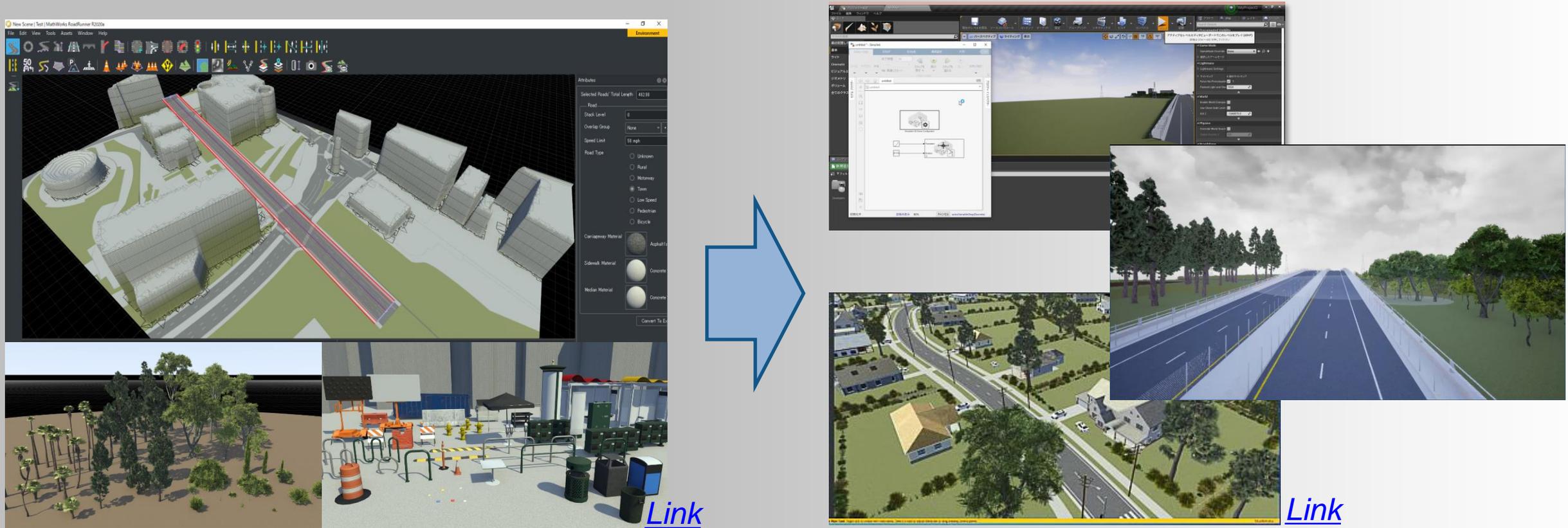


点群特徴の抽出とマッチング



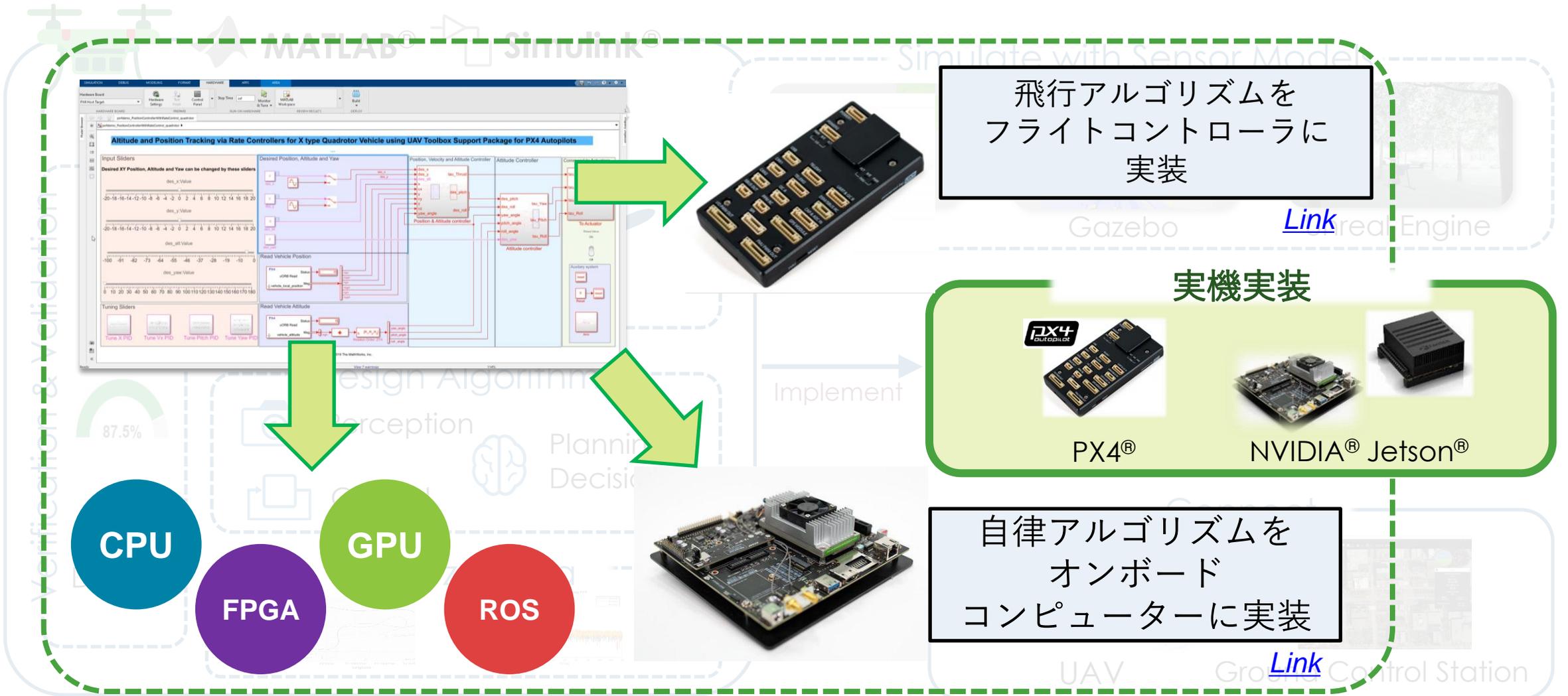
閉ループ検出
姿勢グラフの作成
姿勢最適化

無人航空機シミュレーションのための3Dシーン構築



自律アルゴリズムのテストやシミュレーションのための3Dシーン構築

自動コード生成と実機実装



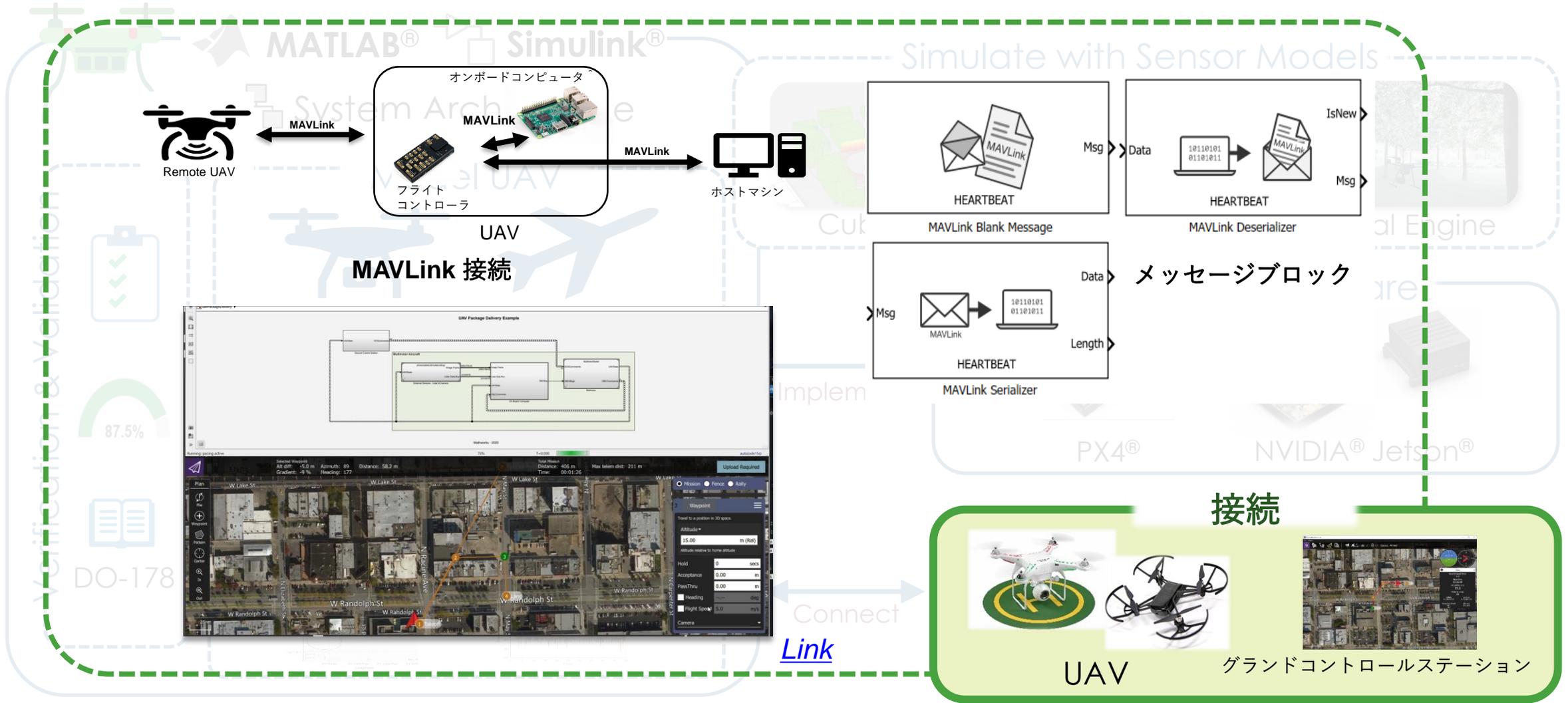
飛行制御アルゴリズムをPX4ターゲットに実装

The image shows a MATLAB/Simulink workspace with two main windows:

- Simulink Model:** Titled "px4demo_PCtrl_RateCtrl_quad". It contains a Simulink diagram for "Altitude and Position Tracking via Rate Controllers for X type Quadrotor Vehicle using UAV Toolbox Support Package for PX4 Autopilots". The model includes:
 - Input Sliders:** Controls for desired XY position, altitude, and yaw.
 - Desired Position, Altitude and Yaw:** A block that outputs desired position (des_x, des_y, des_z) and yaw (des_yaw).
 - Position, Velocity and Altitude Controller:** A block that takes desired position and yaw as input and outputs desired velocity (des_vx, des_vy, des_vz) and desired attitude (des_roll, des_pitch, des_yaw).
 - Altitude Controller:** A block that takes desired altitude and current altitude as input and outputs a desired roll/pitch angle (des_roll_pitch).
 - Command to Actuators:** A block that outputs throttle (des_thrust) and roll/pitch/yaw angles (des_roll_pitch_yaw) to the actuators.
- jMAVSim Window:** A 3D visualization of a quadrotor drone flying in a virtual environment. The drone is shown in flight against a blue sky and green trees. A compass is visible in the bottom left corner of the jMAVSim window.

At the bottom of the Simulink window, the status bar indicates: "Running the model on 'PX4 Host Target'...", "View diagnostics", "54% T=2.480", and "auto(FixedStepDiscrete)".

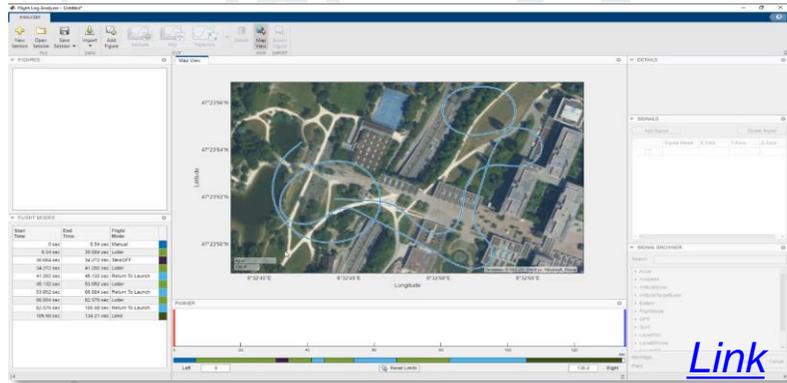
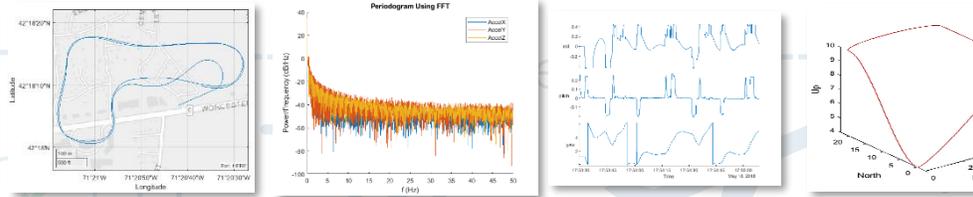
MAVLinkプロトコルを通じて無人航空機ハードウェアに接続



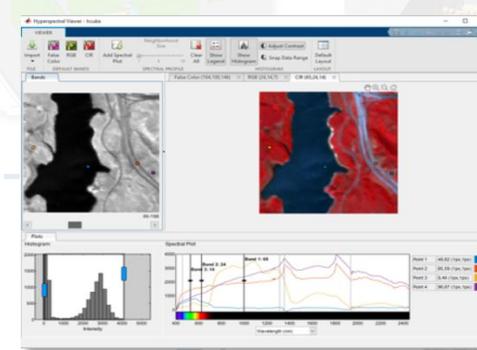
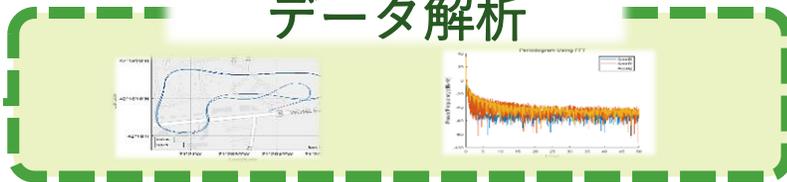
飛行データ解析

フライトログ解析

センサーデータ解析



データ解析



[Link](#)

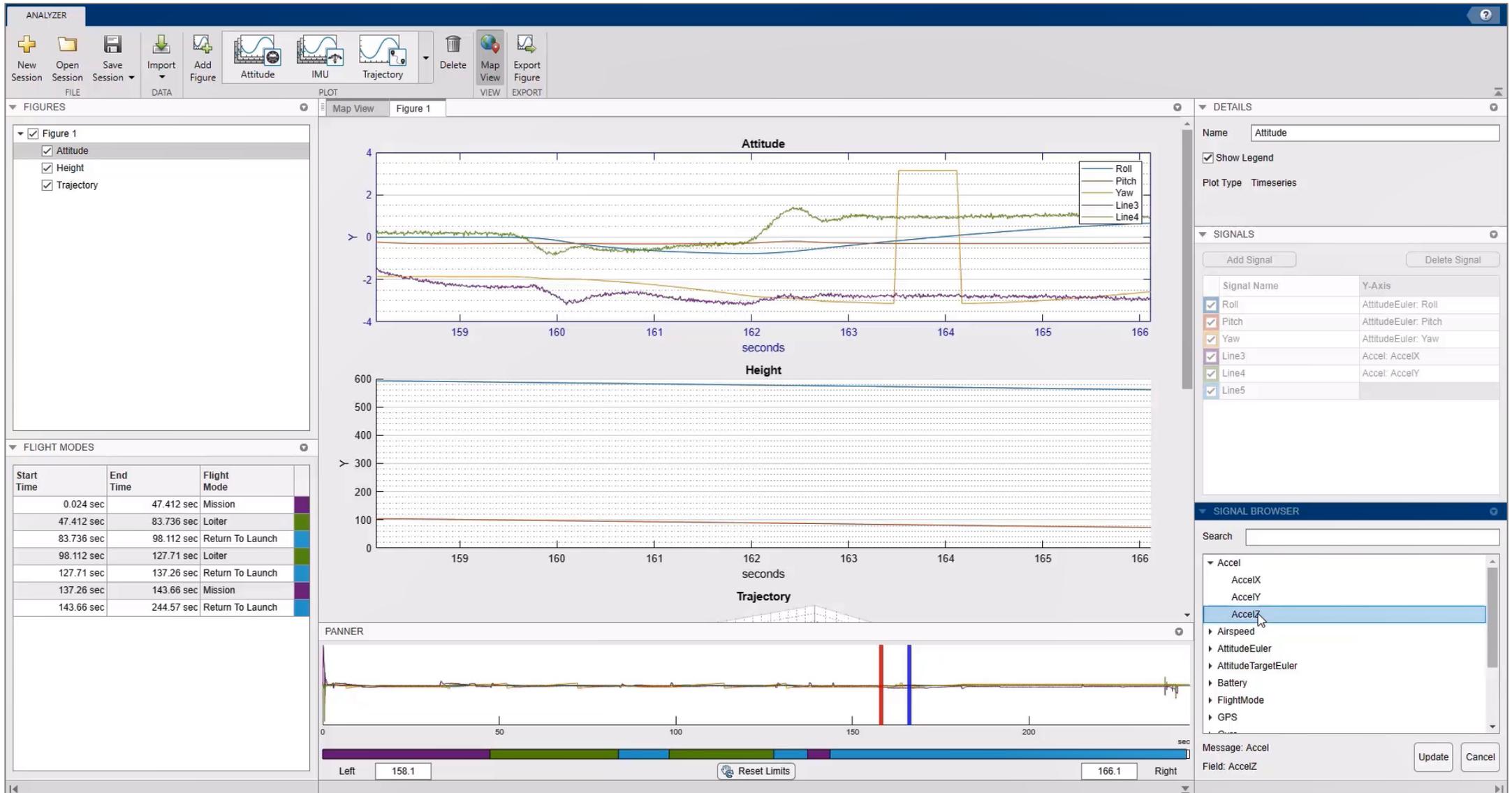
[Link](#)

Connect

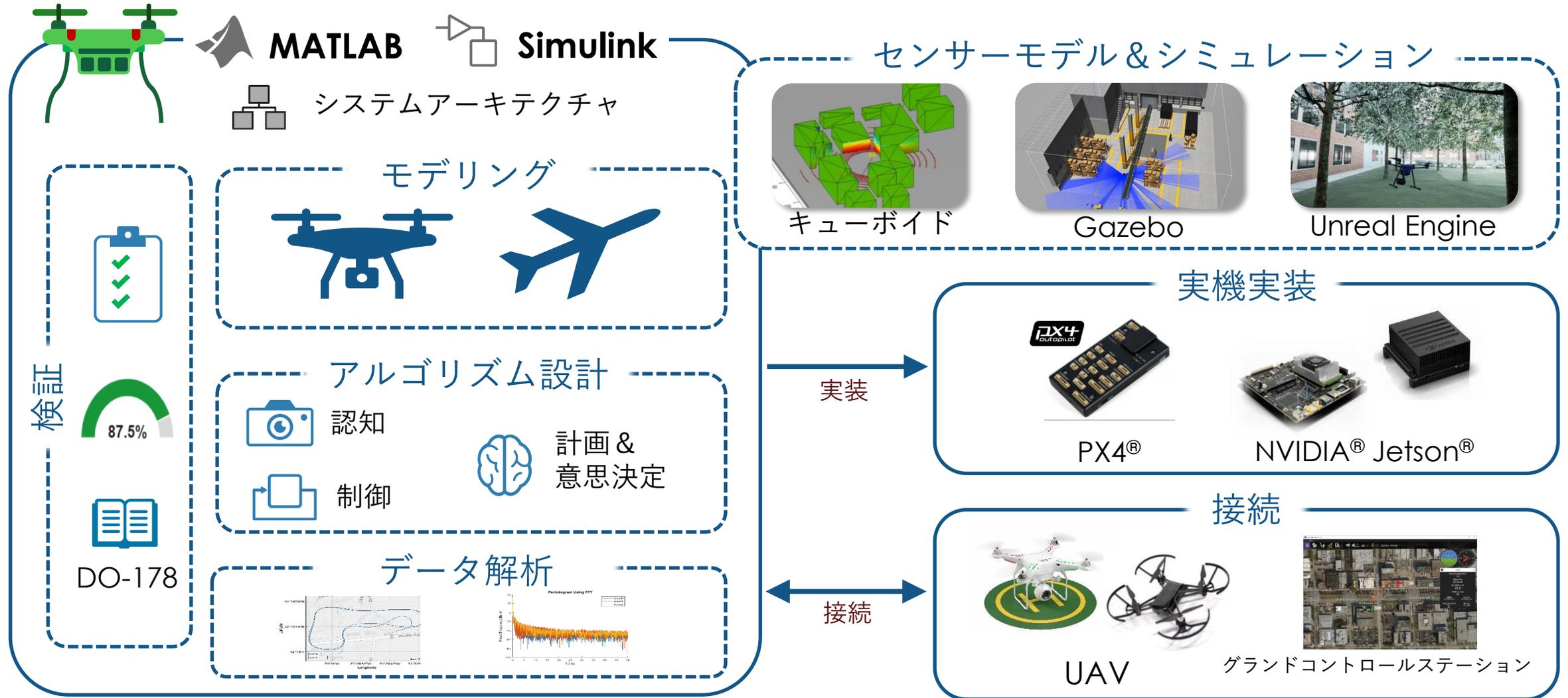
UAV

Ground Control Station

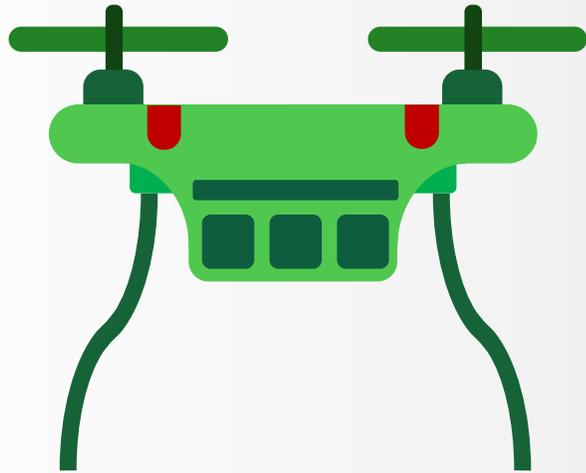
例: フライトログ解析



無人航空機アプリケーションを開発するための統合ワークフロー



まとめ：MATLAB および Simulink による自律無人航空機開発



無人航空機システムと
自律アルゴリズムを
設計検証するための強力な機能



アイデアから製品開発までを
カバーする統合開発環境



仮想飛行テストを通じて
テスト・検証ツールへの拡張が可能



© 2021 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.