

MATLAB®/Simulink®による無人航空機の設計・開発

MathWorks Japan

Application Engineering

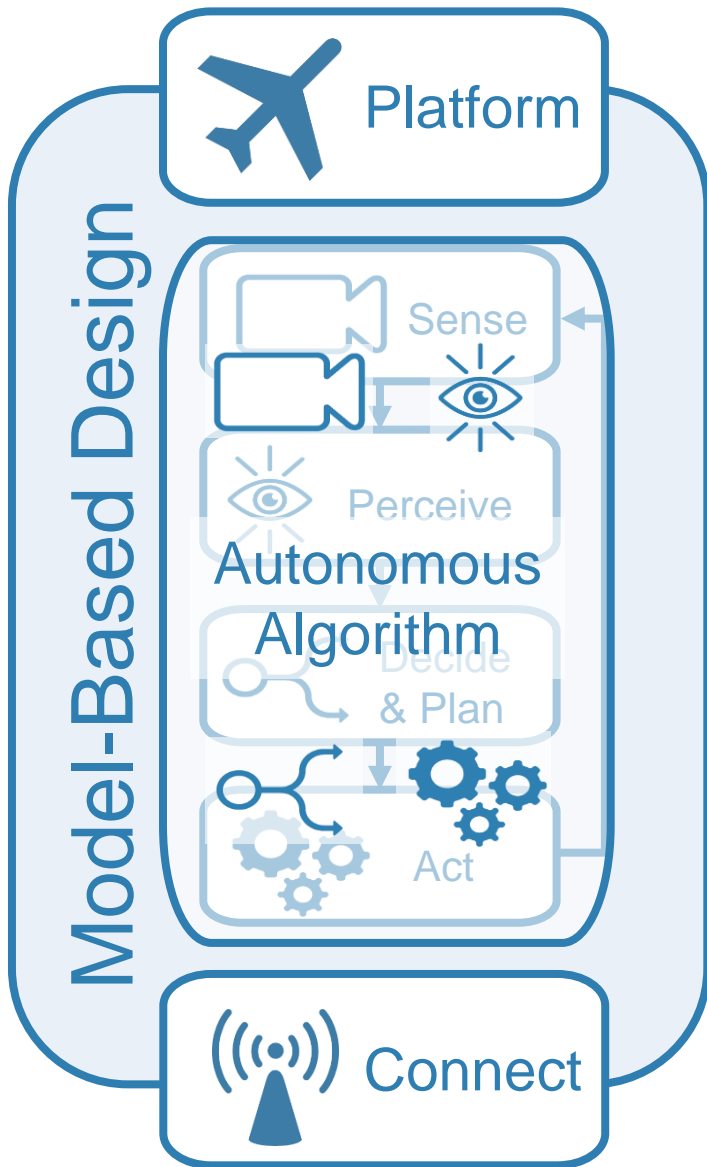
Application Engineer

能戸 フレッド・Fred Noto

無人航空機の進化



はじめに



無人航空機(UAV)はますます普及
UAV自律機能の役割が増加

MATLAB/SimulinkはUAV開発の統一環境



Hobby Drones



Entertainment Coverage



Delivery Drones

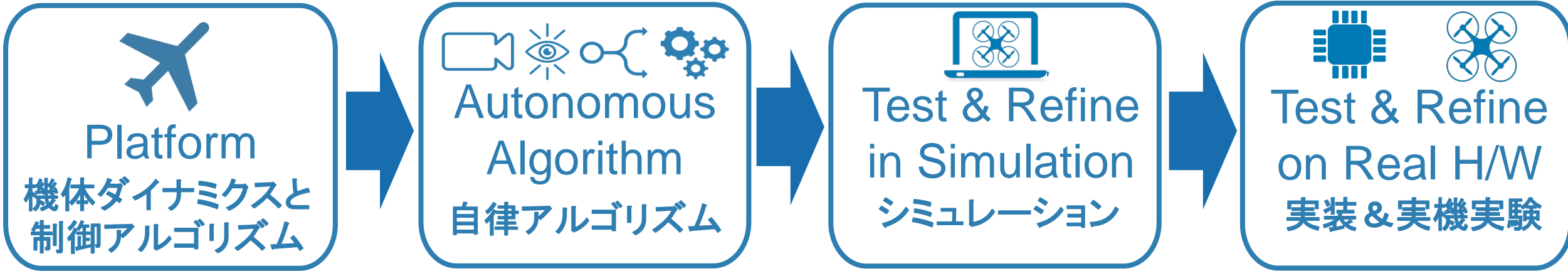


Solar-Powered Drones

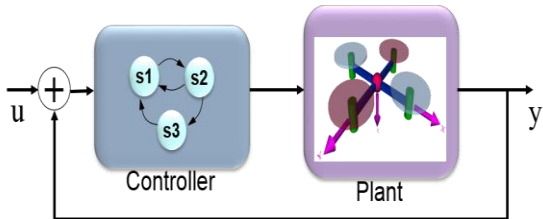


Surveillance and Defense

アジェンダ: UAV開発プロセス



ステップ①:
機体ダイナミクスの
理解と飛行制御
アルゴリズムの設計



ステップ②:
ビジョン、レーダ、知覚
アルゴリズムの設計・検証



ステップ③:
アルゴリズムの
検証と実装

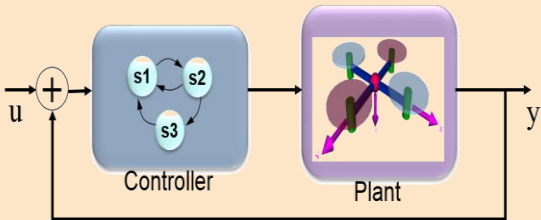



アジェンダ: UAV開発プロセス



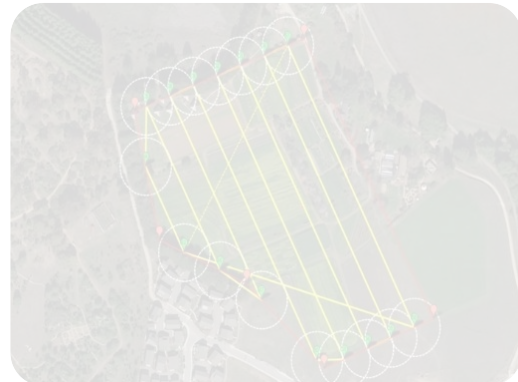
Platform
機体ダイナミクスと
制御アルゴリズム

ステップ①:
機体ダイナミクスの
理解と飛行制御
アルゴリズムの設計

**Autonomous
Algorithm**
自律アルゴリズム

ステップ②:
ビジョン、レーダ、知覚
アルゴリズムの設計・検証




**Test & Refine
in Simulation**
シミュレーション

ステップ③:
アルゴリズムの
検証と実装



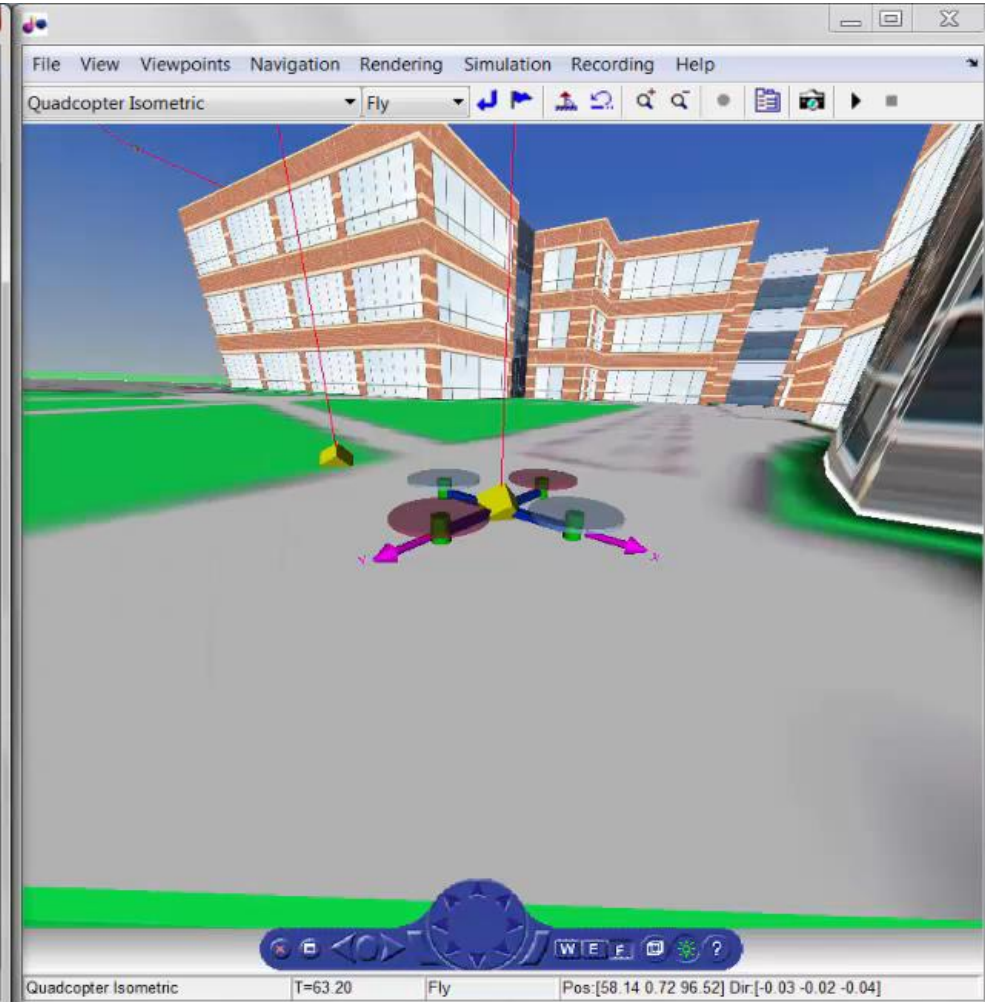
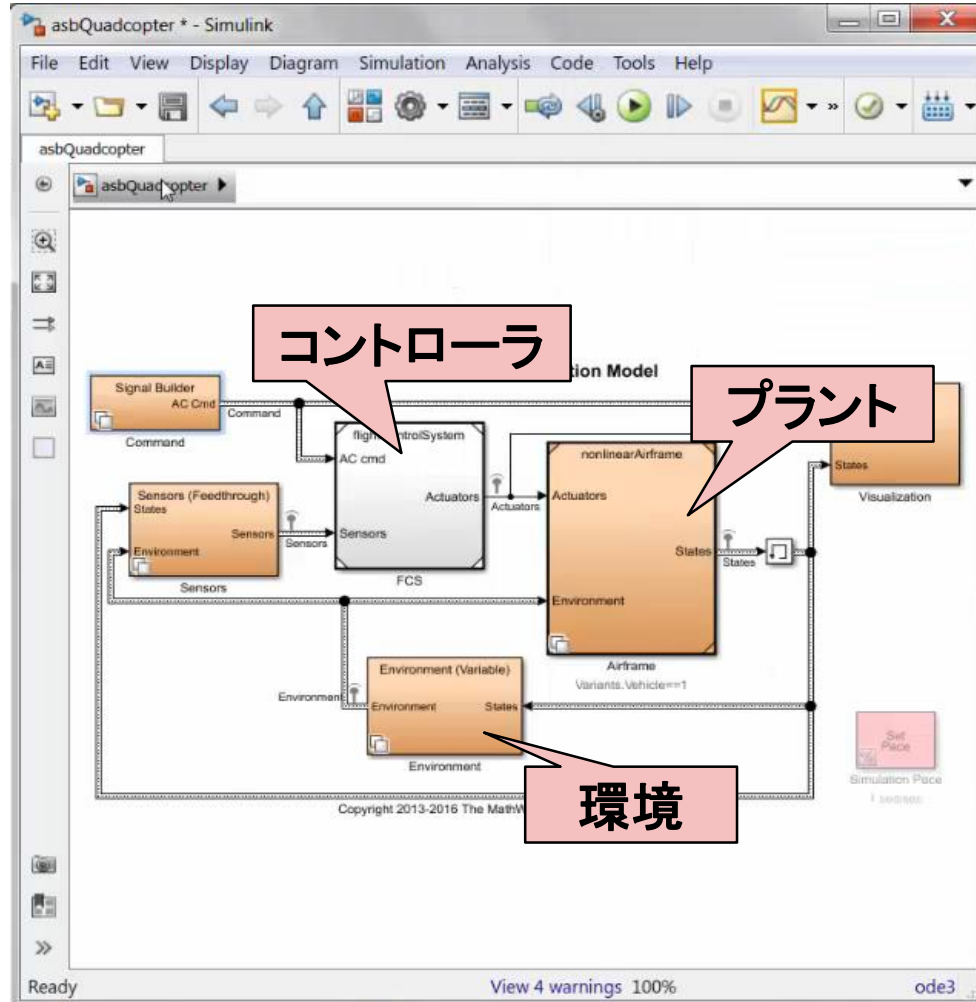

**Test & Refine
on Real H/W**
実装 & 実機実験



MATLAB/Simulinkでのシステムシミュレーション



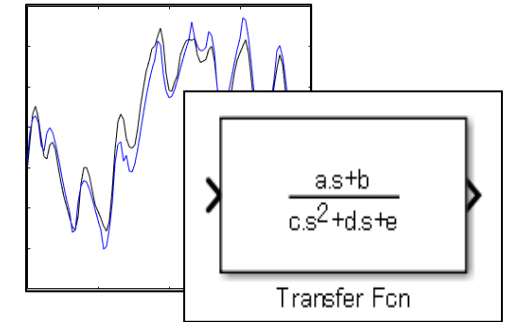
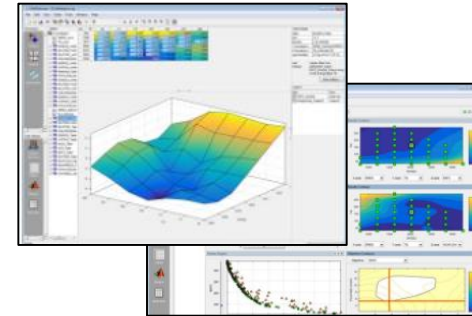
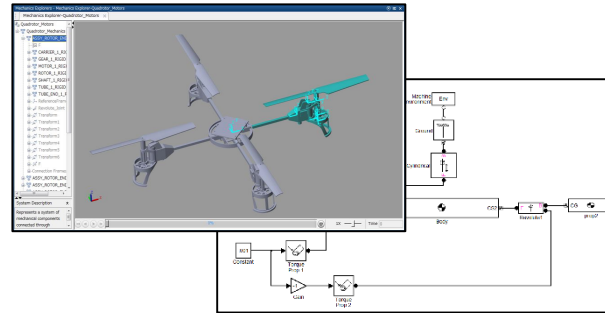
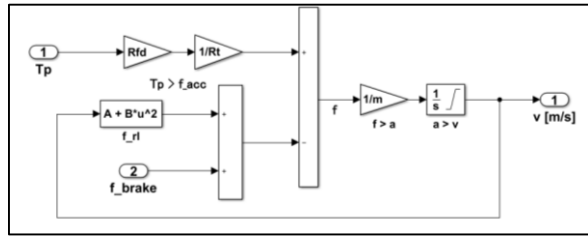
システムレベルシミュレーションを目的としたモデル設計



MATLAB/Simulinkによるプラントモデリング



Simulink環境で様々な作成方法を柔軟に対応



物理法則に基づく方法
数式モデリング

実験データに基づく方法
データドリブンモデリング



プログラミング
MATLAB

ブロック線図
Simulink

数式処理
Symbolic Math Toolbox™

物理モデリング
Simscape™製品群

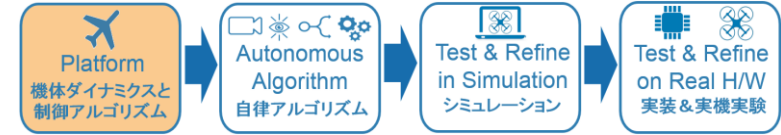
パラメータチューニング
Simulink Design Optimization™

統計的手法
Model Based Calibration Toolbox™

システム同定
System Identification Toolbox™

ニューラルネットワーク
Neural Network Toolbox™

Simscapeによる制御対象の物理モデリング



数式不要で直感的にコンポーネントの追加・削除が容易

強電 パワエレ	弱电 デジアナ	油圧・ 熱流体	機械 3D	動力伝達 1D
Power Systems	Electronics	Fluids	Multibody	Driveline

Simscape

MATLAB & Simulink

基本部品ライブラリ

Simscape

機械	油圧	電気	磁気
熱	熱流体	二相流体	空気圧

“Simscape Language”
によるカスタム部品/ドメイン開発

初期段階ではトレード
オフスタディーで活用

CADモデル
インポート可能

Aerospace Blockset™でのダイナミクスモデリング



航空機コンポーネントのモデリングを簡略化・効率化

ライブラリ: aerolibv1 - Simulink

ファイル(F) 編集(E) ツール表示(V) 情報表示(D) ブロック図(R) 解析(A) ヘルプ(H)

aerolibv1

- Equations of Motion
- Propulsion
- Actuators
- Flight Instruments
- Pilot Models

ライブラリ: aerolib6dof2 - Simulink

ファイル(F) 編集(E) ツール表示(V) 情報表示(D) ブロック図(R) 解析(A) ヘルプ(H)

aerolib6dof2

6DOF (Euler Angles)

6DOF (Quaternion)

6DOF Wind (Wind Angles)

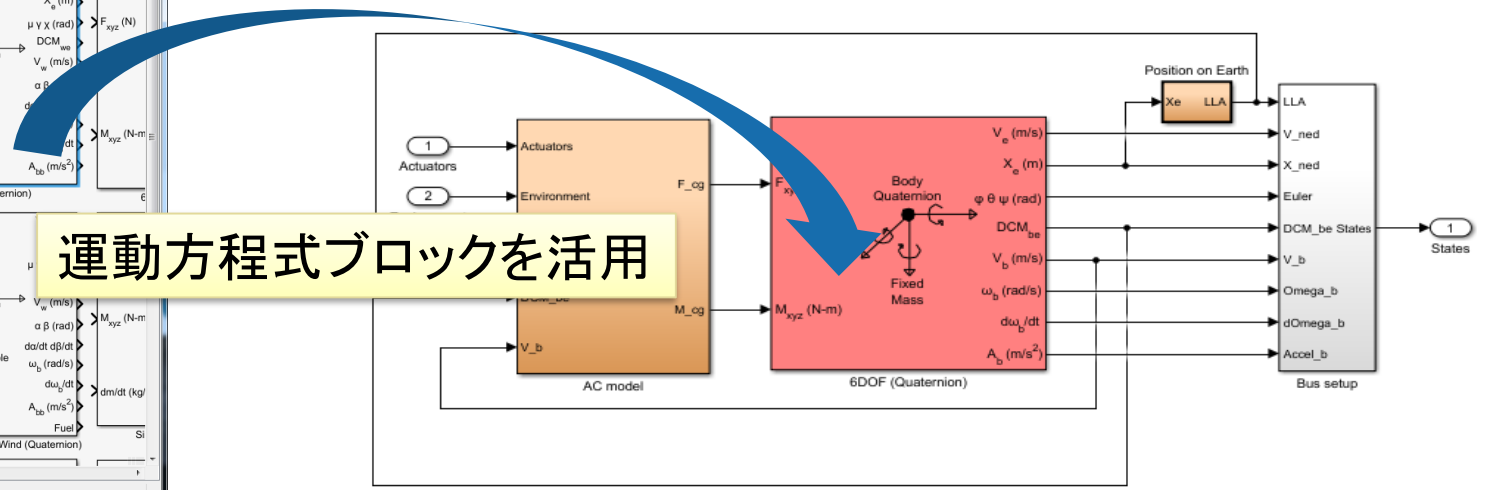
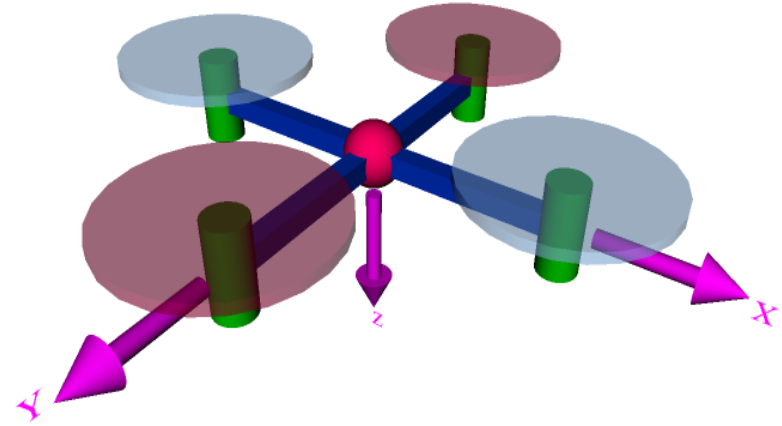
6DOF Wind (Quaternion)

Simple Variable Mass 6DOF (Euler Angles)

Simple Variable Mass 6DOF (Quaternion)

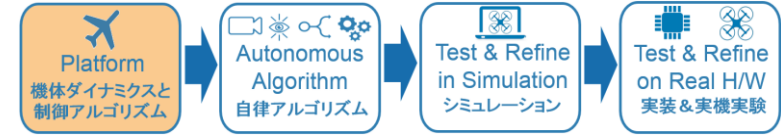
Simple Variable Mass 6DOF Wind (Wind Angles)

Simple Variable Mass 6DOF Wind (Quaternion)

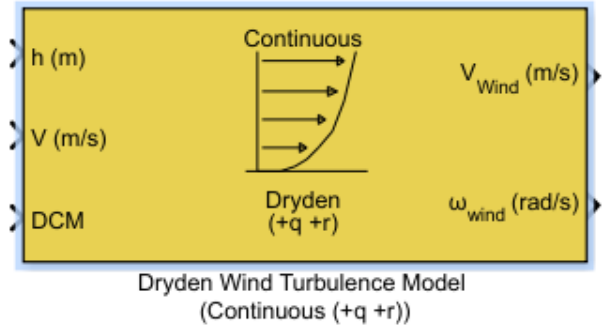
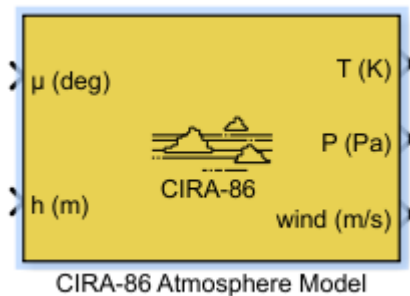
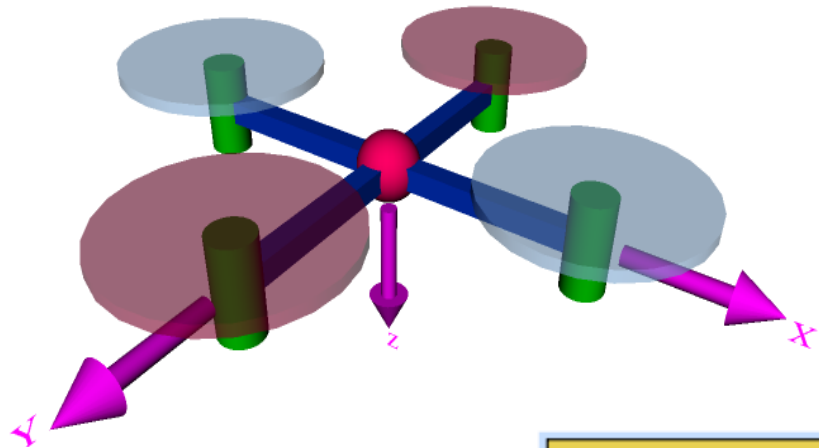
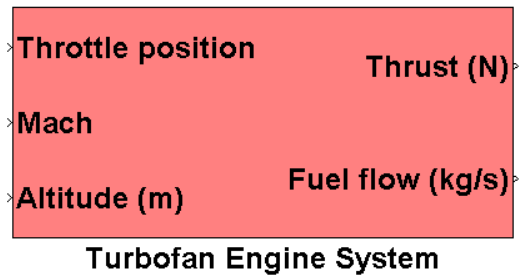
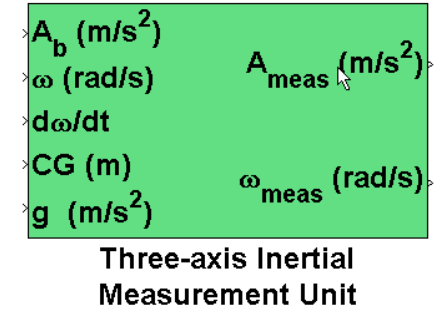
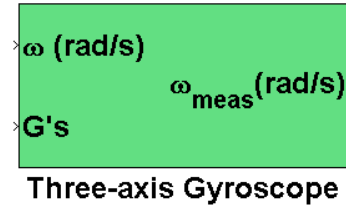
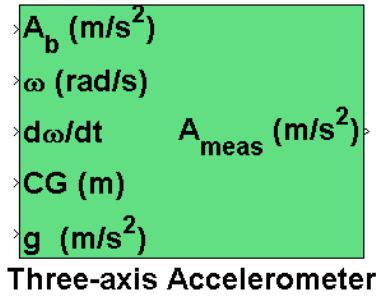


運動方程式ブロックを活用

Aerospace Blocksetでのダイナミクスモデリング



センサー、推進力、アクチュエータ、環境ブロックを提供



制御アルゴリズムの設計

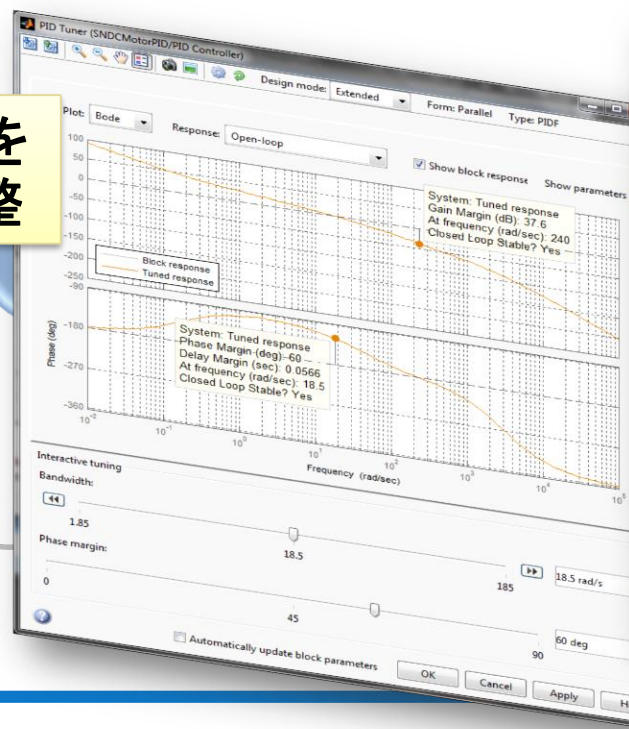


プラントとコントローラを同一環境で組み合わせたシステムレベルの動作検証

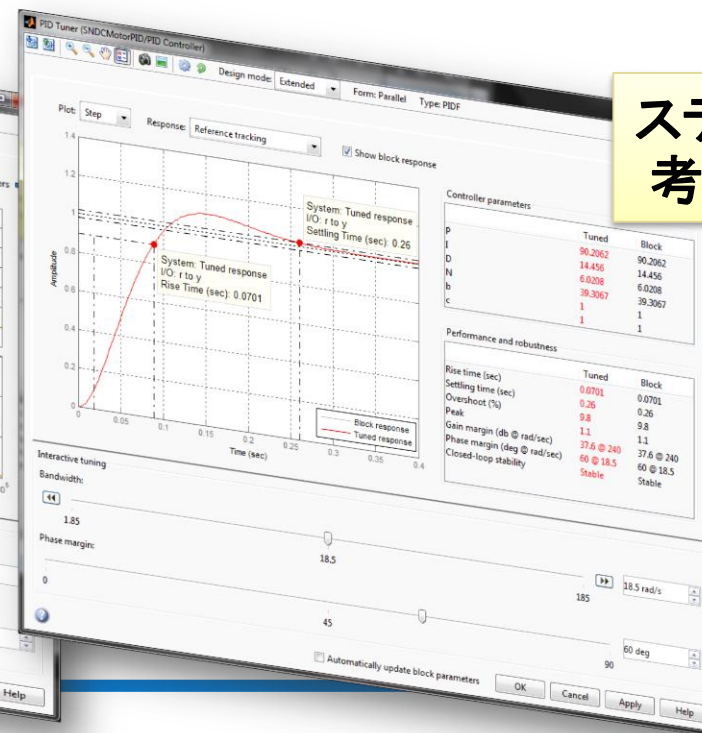
Simulink Control Design™ と Control System Toolbox™

プラントモデルの線形化、PIDコントローラの自動チューニング

周波数応答を
考慮した調整



ステップ応答を
考慮した調整



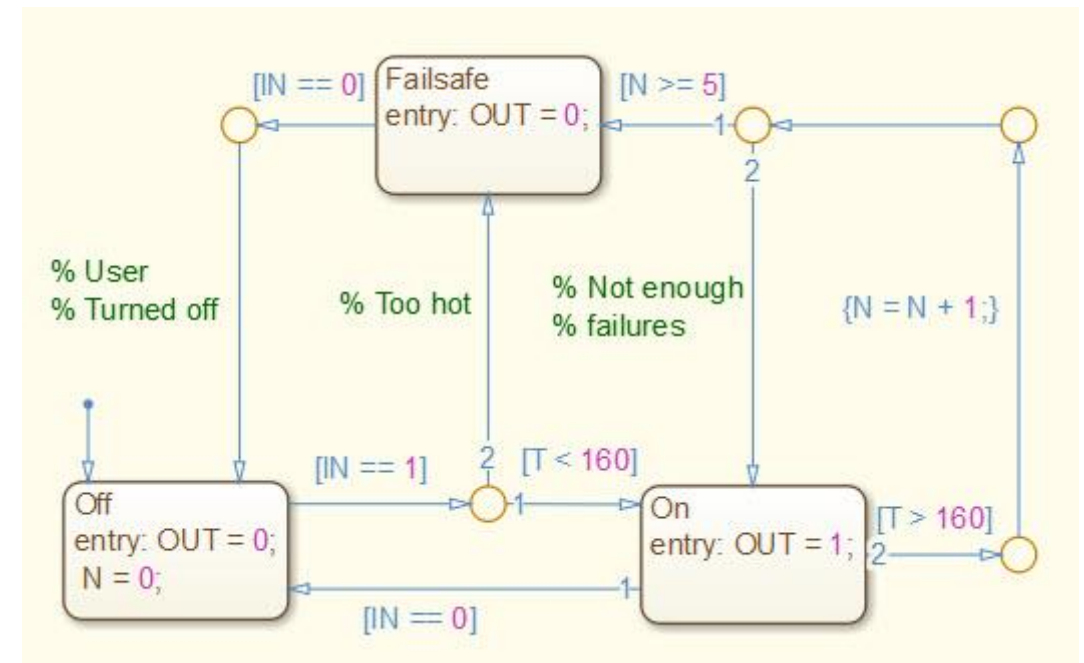
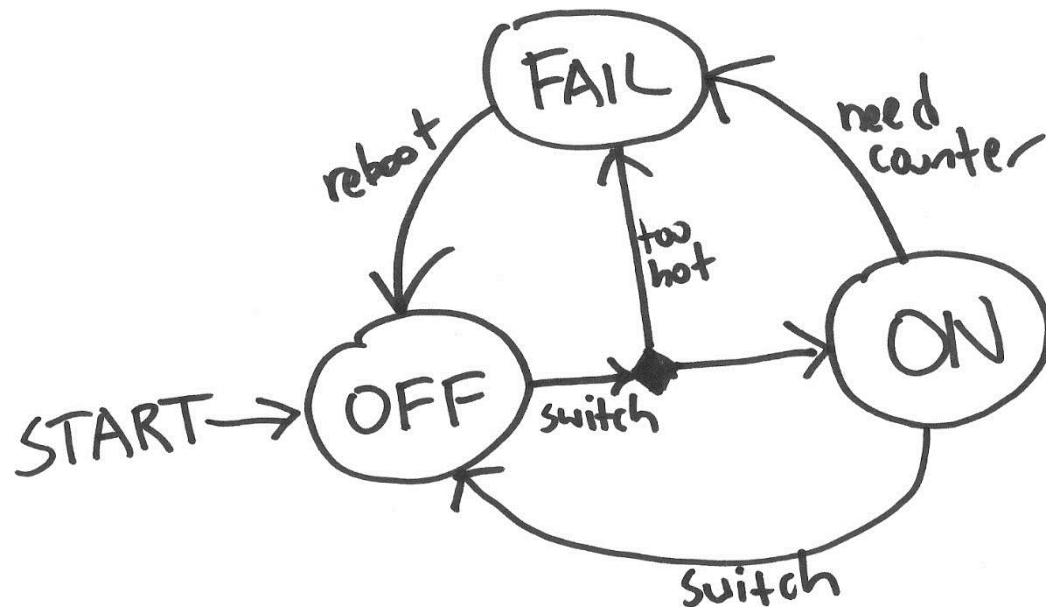
制御アルゴリズムの設計



Stateflow®を活用したモードロジック等の瞬時的変化の設計を容易化

Simulinkは動的システムの連続的な変化の設計を得意とします

システムは連続的と瞬時的な変化に対応する必要があります→モードロジック設計

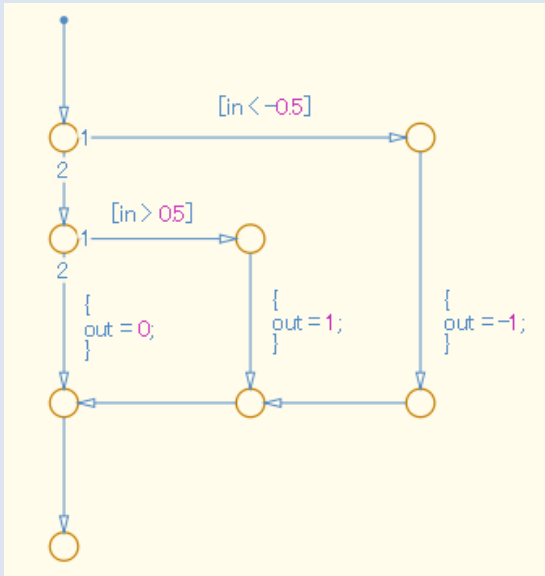




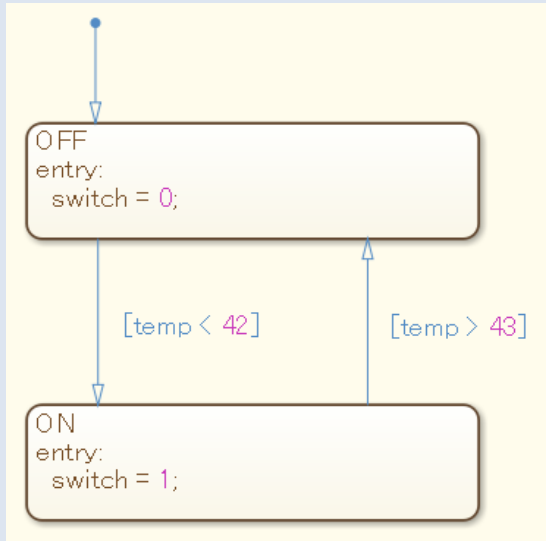
Stateflowによる色々なロジック表現

設計したいアルゴリズムに応じて様々なデザインスタイルを活用できます

フローチャート



状態遷移図



状態遷移表

Block: untitled/State Transition Table*

ステート	遷移 (条件/アクション/遷移先ステート)	
	if	else-if(1)
state1 en: out = 1;	[in > 1]	[in > 2]
state1_1 en, du: cnt = 0;	[flag]	[x > 0]
state1_2 en: cnt = cnt + 1;	\$NEXT	[x = x + 1;]
state2 en: out = 2;	[in < -1]	
state3 en: out = 3;	[after(5, sec)]	
	state1	\$PREV
	state1	\$NEXT
	\$SELF	state1
	state3	state3
	% IGNORE %	

真理値表

条件テーブル

	説明	条件	D1	D2
1	Hot	t > T_thresh	T	T
2	Dry	h < H_thresh	T	-
	アクション: アクション テーブルから行を指定		CoolOn, HumidOn	CoolOn

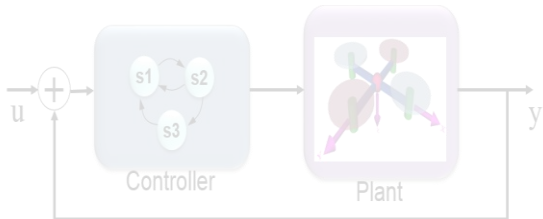
アクション テーブル

#	説明	アクション
1	Turn On Cooling (This implicitly reduces humidity)	CoolOn: cooler = 1; heater = 0; humidifier = 0;
2	Turn On Heater (This implicitly reduces humidity)	HeatOn: heater = 1; cooler = 0; humidifier = 0;
3	Turn On Humidifier	HumidOn: humidifier = 1;

アジェンダ: UAV開発プロセス

Platform
機体ダイナミクスと
制御アルゴリズム

ステップ①:
機体ダイナミクスの
理解と飛行制御
アルゴリズムの設計



**Autonomous
Algorithm**
自律アルゴリズム

ステップ②:
ビジョン、レーダ、知覚
アルゴリズムの設計



**Test & Refine
in Simulation**
シミュレーション

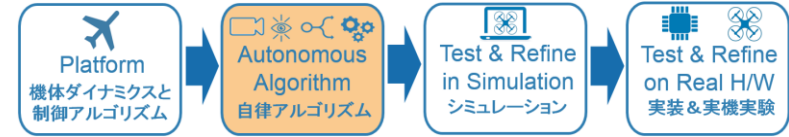
ステップ③:
アルゴリズムの
検証と実装



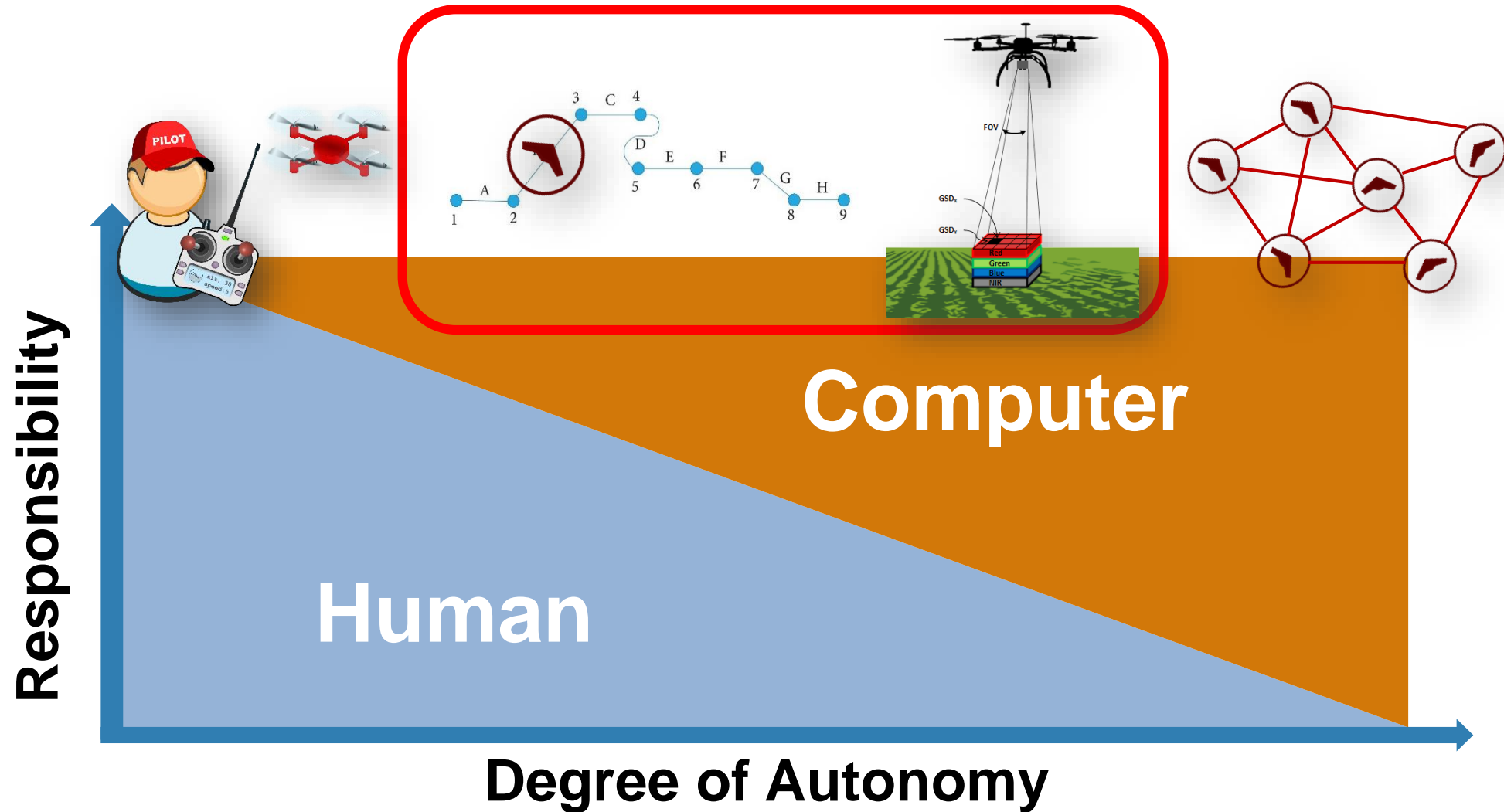
**Test & Refine
on Real H/W**
実装 & 実機実験



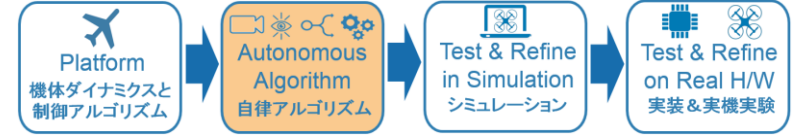
自律テクノロジーのトレンド



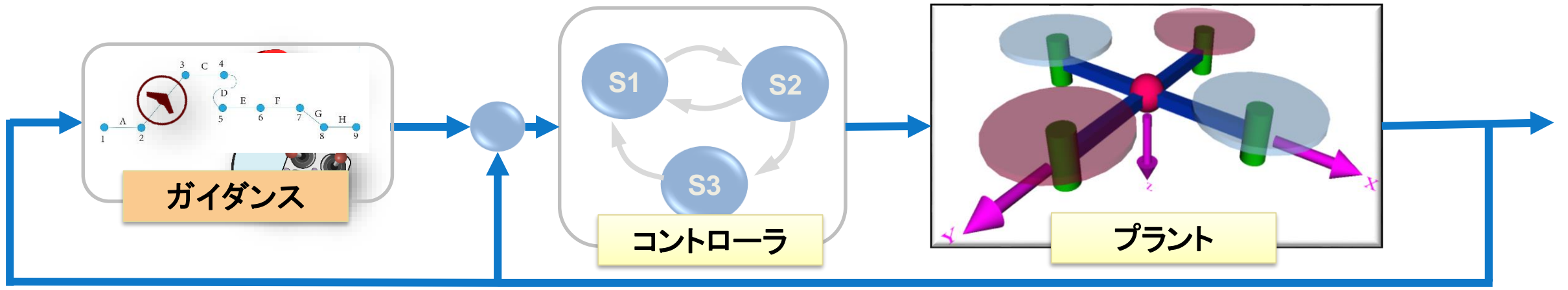
設計したいアルゴリズムに応じて様々なデザインスタイルを活用できます



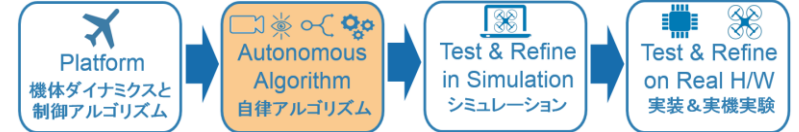
パスプランニングの追加



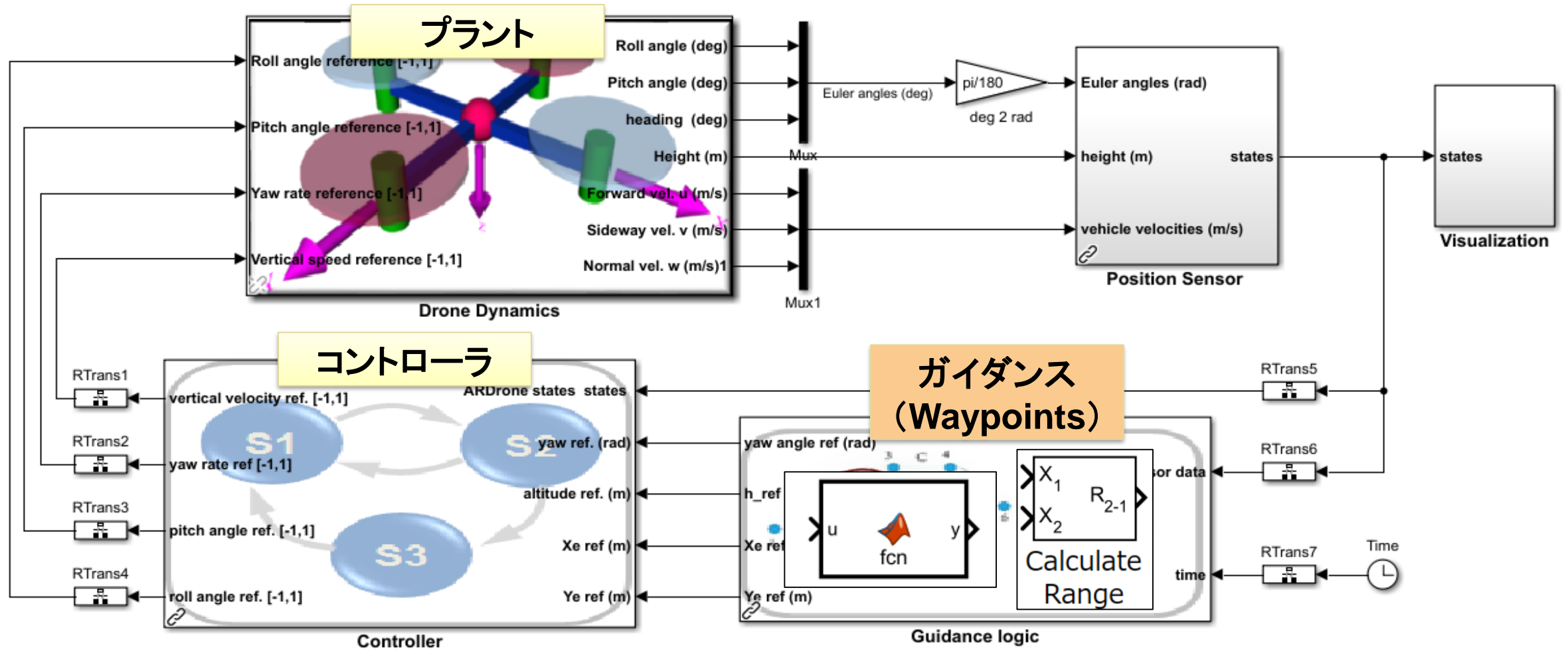
ガイダンスアルゴリズムを組み合わせたシステムシミュレーション



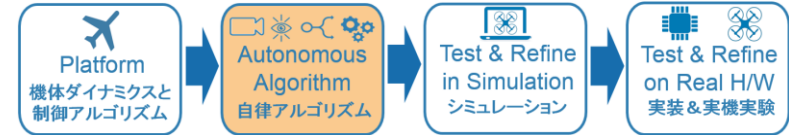
パスプランニングの追加



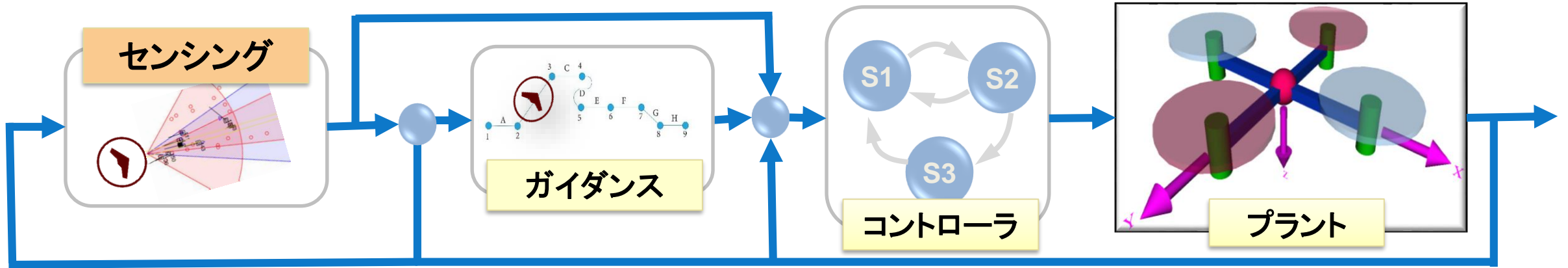
ガイダンスアルゴリズムを組み合わせたシステムシミュレーション



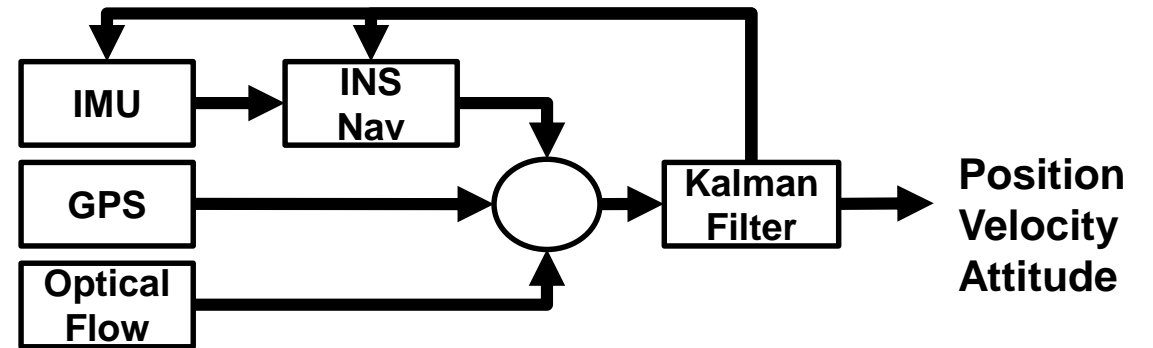
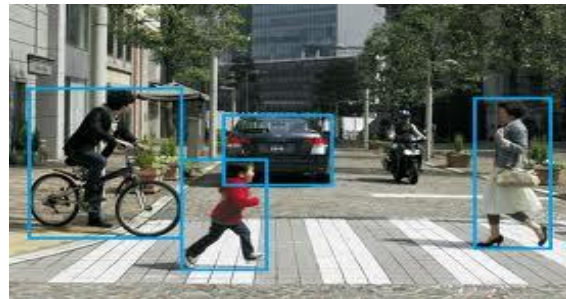
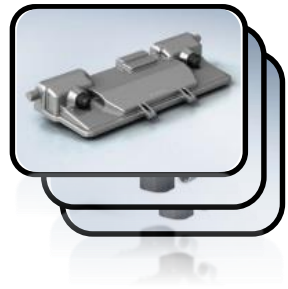
センシングアルゴリズムの追加



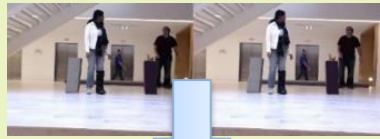
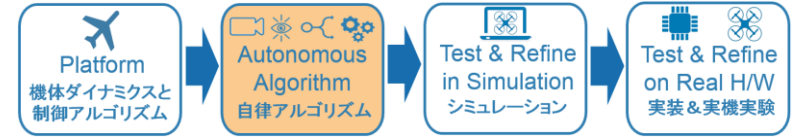
視覚アルゴリズムによる自律機能の増加



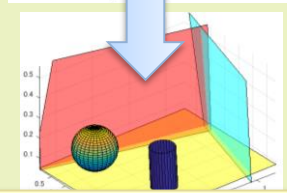
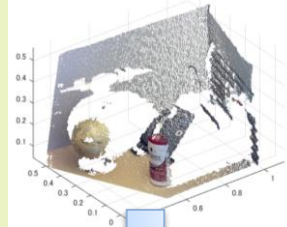
カメラ,
レーダー,
LiDAR,
GPS,
IMU,
, ...



MATLABによる認識ソリューション



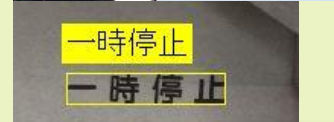
ステレオビジョン



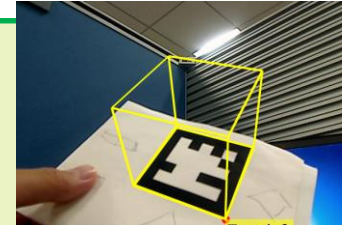
3次元点群処理



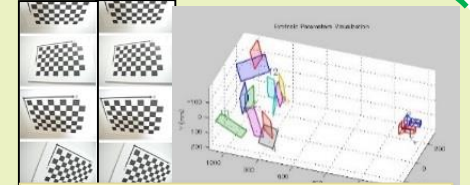
物体の検出



文字認識(OCR)



AR(拡張現実)



カメラキャリブレーション

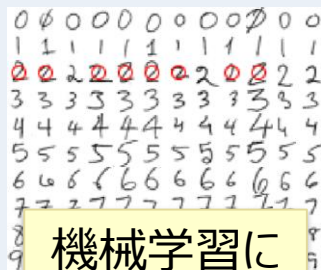
Image Processing Toolbox™
Computer Vision System Toolbox™



顔、人物認識



画像検索、
分類(BoF)



機械学習に
よる分類



ディープラーニング
(CNN/Faster R-CNN)

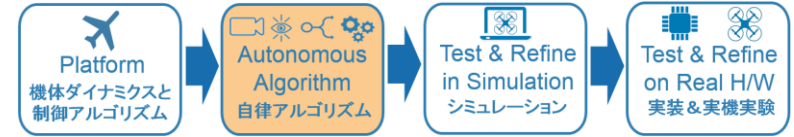
Neural Network Toolbox™
Statistics and Machine Learning Toolbox™

トラッキング・センサーフュージョン



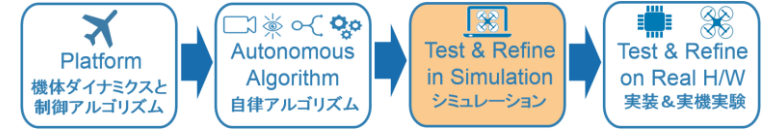
Automated Driving System Toolbox™

センサーデータの取得



The image displays a workflow for sensor data acquisition and processing. On the left, a 'Video Player' window shows a first-person perspective of a modern hallway with wooden floors and a reception desk. In the center, a 'Figure 1' window shows a 3D point cloud of the same hallway, with a red robot-like object positioned on the floor. On the right, a script editor window shows a portion of a MATLAB script with the text 'script' and 'Ln 26 Col 16'. The background includes a physical sensor device and a close-up of a sensor's lens.

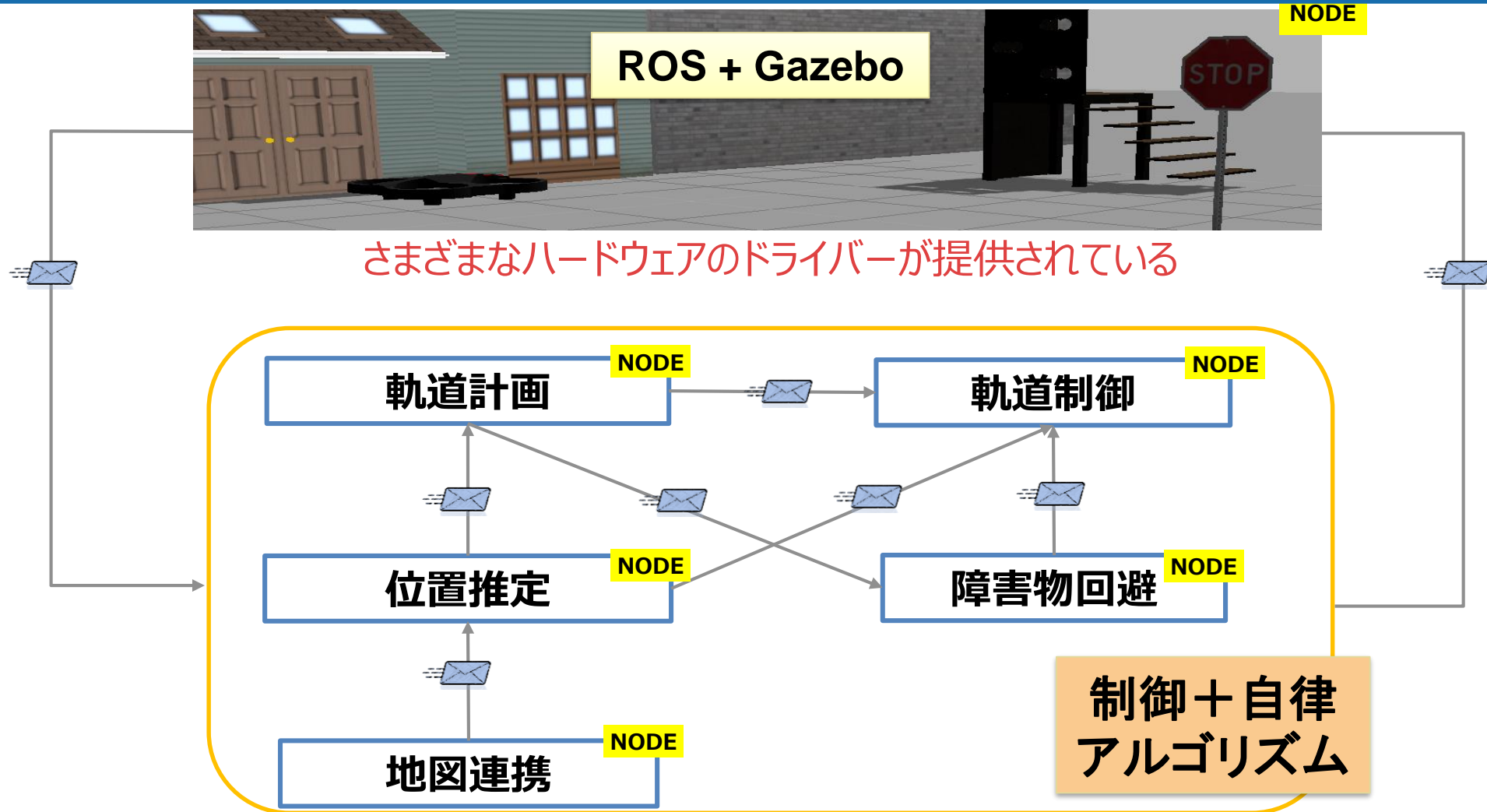
Robot Operating System (ROS)の登場



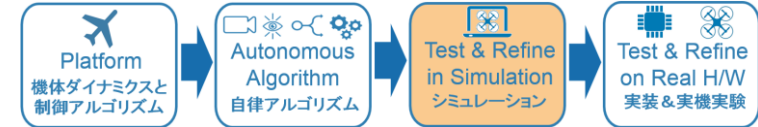
仮想センサーからデータ取得してアルゴリズムを評価



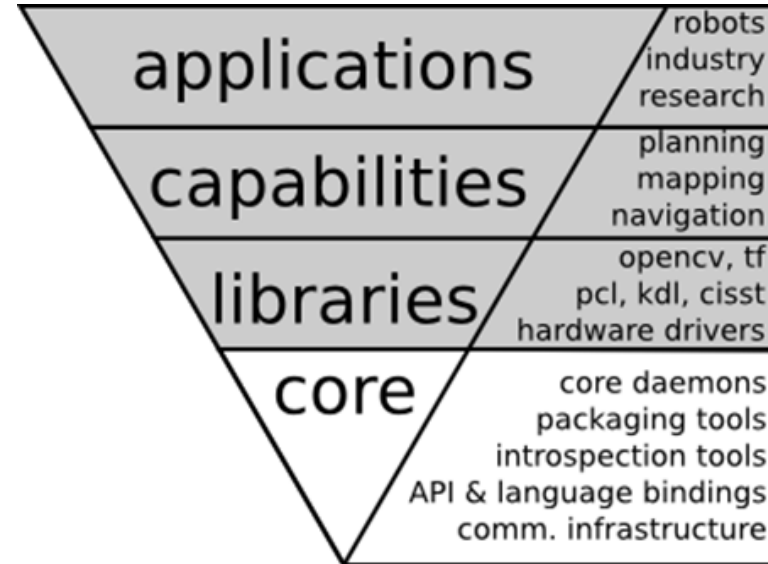
さまざまなハードウェアのドライバーが提供されている



ROS(Robot Operating System)とは?



- ロボットの内部プロセス間通信のための標準的なプロトコルを提供
- 複数の開発言語とのインタフェース (C++, Python, Lua, Java, etc.)
- ランタイムやデータ解析用ツール
- 良く使用されるアルゴリズムやドライバのパッケージ
- オープンソース

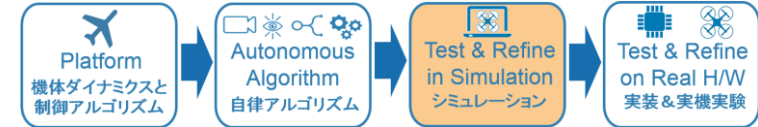


研究者の皆様が、標準的なツールやインタフェースを使用して既存のコンポーネントを一から作り直さずに、新しいロボティクスシステムを手早く開発することが出来るようにすること、を意図しています。



Jonathan Bohren

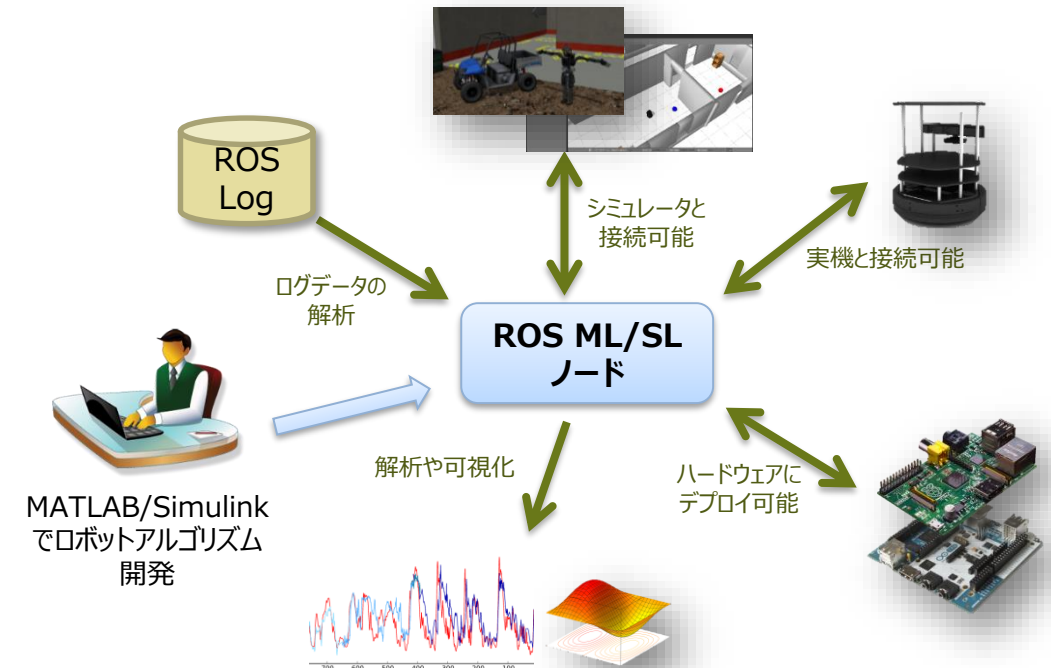
MATLABとROSによる自律制御システム開発



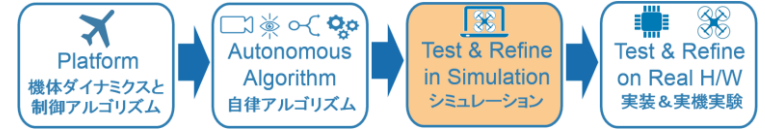
MATLAB/Simulinkの柔軟な開発環境とROSを連携しロボティクス開発を加速化

Robotics System Toolbox™

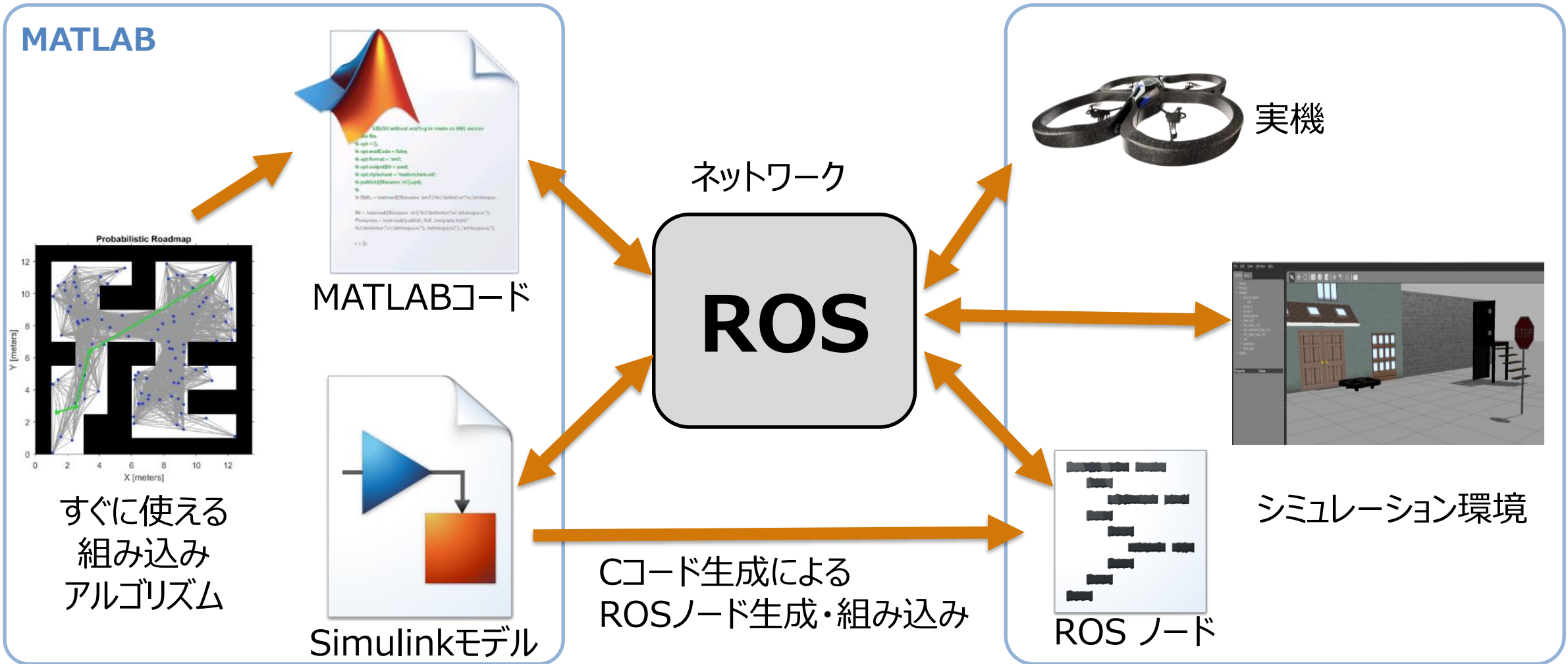
- MATLAB/SimulinkをROSネットワーク上のノードとして登録可能
 - ROSマスターとして起動することも可能
- MATLAB上で開発したアルゴリズムを、直接ROSネットワークに接続して検証可能
- Simulinkモデルからのコード生成機能を使い、C++ ROSノードを生成可能



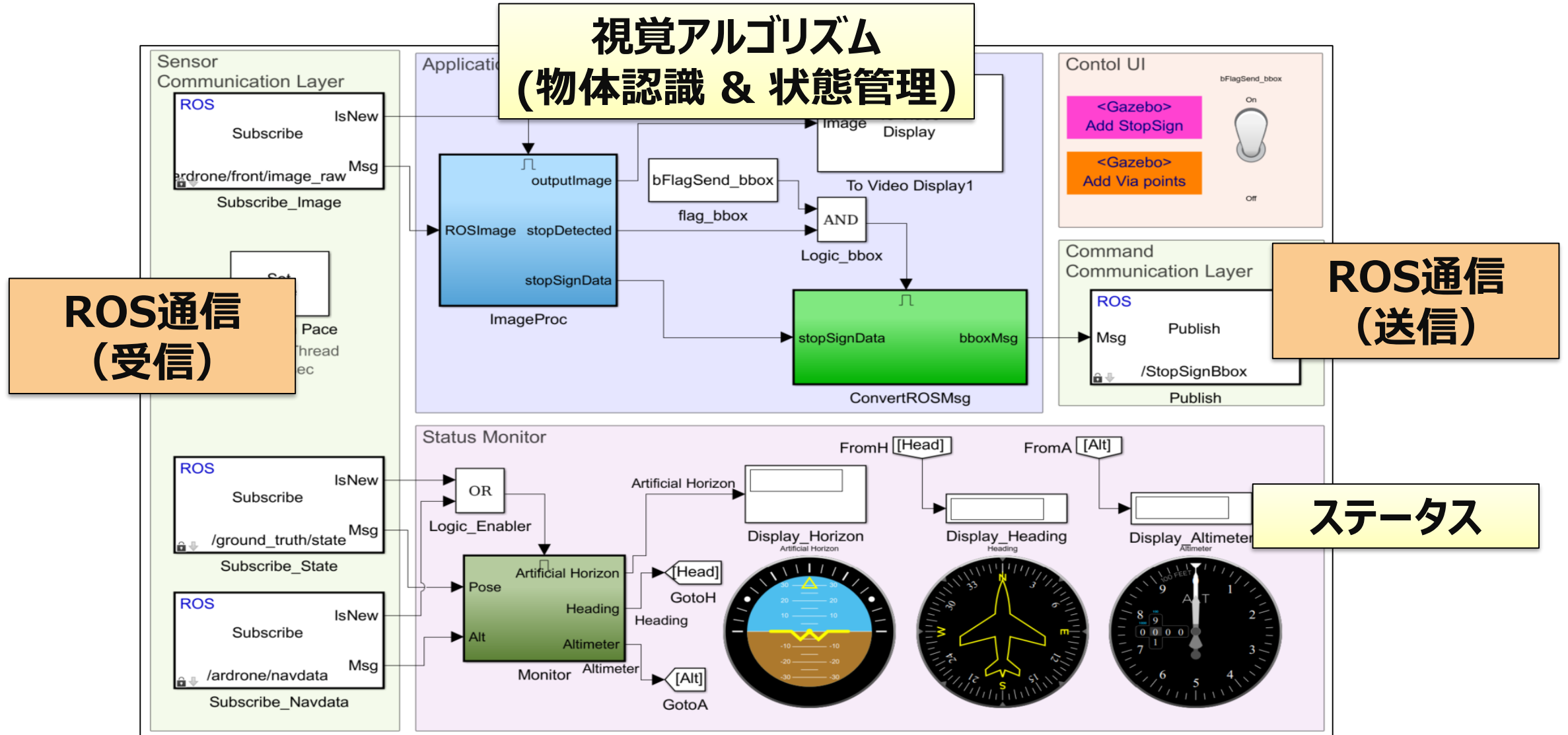
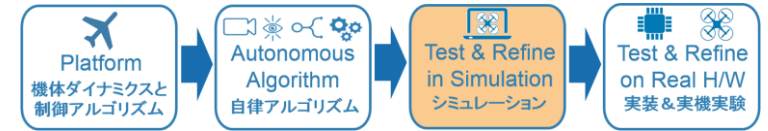
Robotics System Toolboxを使用したROS連携



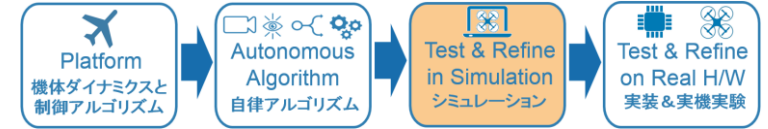
ROSとの連携によるロボティクス開発加速化



MATLAB・Simulink・ROSによるシステム設計



自律アルゴリズムのシミュレーション・検証



Robotics System ToolboxのROS連携機能による自律アルゴリズムの検証

The image shows a dual-view environment. On the left, the MATLAB/Simulink interface displays a Simulink model for an autonomous system. The model includes a Sensor Communication Layer with ROS subscribers for image and state data, an Application Layer with image processing and control logic, and a Status Monitor with various display widgets. On the right, the ROS + Gazebo environment is shown within a VMware Workstation. The Gazebo window displays a 3D simulation of a quadcopter in a virtual world with buildings, a stop sign, and a barrier. A property window on the left of the Gazebo window shows details for the 'ground_plane' link.

MATLAB/Simulink

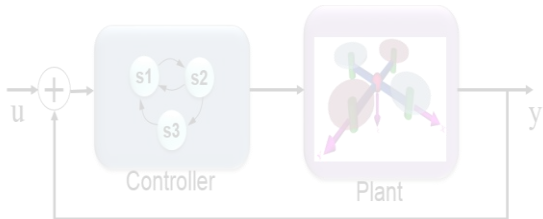

ROS + Gazebo

アジェンダ: UAV開発プロセス



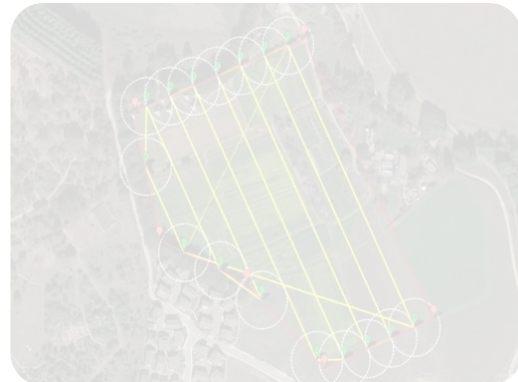
Platform
機体ダイナミクスと
制御アルゴリズム

ステップ①:
機体ダイナミクスの
理解と飛行制御
アルゴリズムの設計


**Autonomous
Algorithm**
自律アルゴリズム

ステップ②:
ビジョン、レーダ、知覚
アルゴリズムの設計




**Test & Refine
in Simulation**
シミュレーション

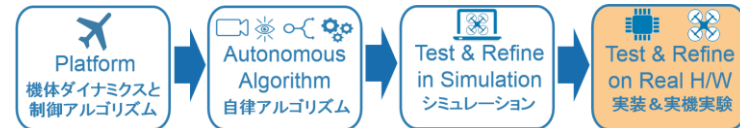
ステップ③:
アルゴリズムの
検証と実装

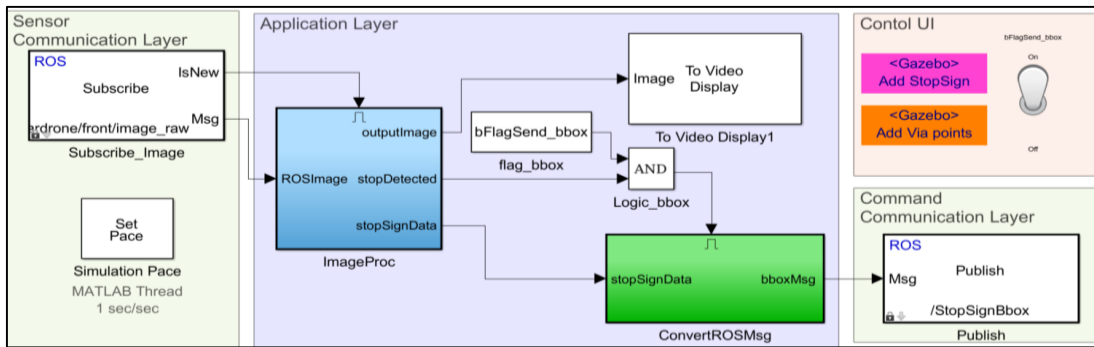
**Test & Refine
on Real H/W**
実装 & 実機実験



コード生成・実装



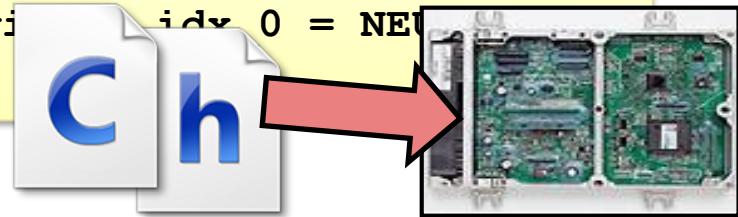
作成したモデルを自動コード生成機能を使用して容易に実装



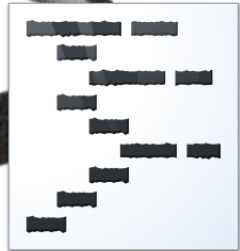
```

if (LeftPos > 4.0F) {
    rtb_Switch1_idx 0 = LOCK;
} else if (LeftPos < 4.0F) {
    rtb_Switch1_idx 0 = NEW;
} else {
    rtb_Switch1_idx 0 = NEUTRAL;
}
    
```

C/C++コード生成



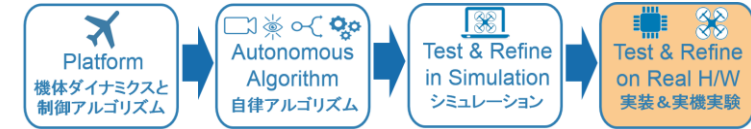
ROSノードC/C++コード生成



ROS ノード

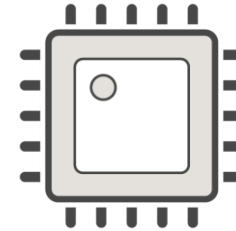
コード生成レポート

コード生成・実装 関連ツール



MATLAB Coder™

- MATLABプログラムからのC/C++コード生成
- スタンドアロン・ライブラリのアプリ作成



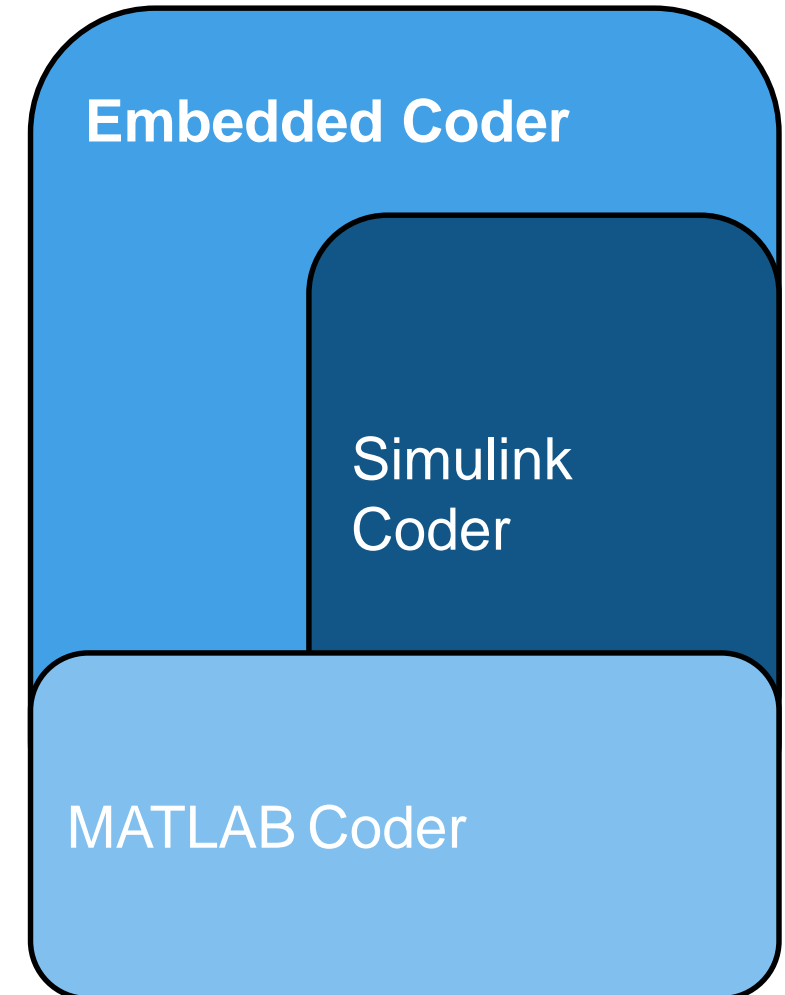
Simulink Coder™

- Simulink/StateflowモデルからのC/C++コード生成
- RCP/HIL試験用コードの生成

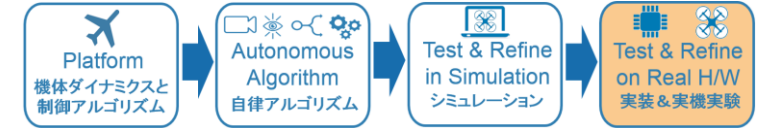
Embedded Coder™

- 組み込み実装に適した効率的なC/C++コードを自動生成
- プロセッサに合わせたコード最適化・サポートパッケージ
- モデル & 生成コード間トレーサビリティ
- モデル・生成コード間等価性検証 (B2Bテスト)

※Robotics System Toolboxと組み合わせることでROSノード用コード生成可能



ハードウェアサポートパッケージ



Products Solutions Academia Support Community Events

Hardware Support

Hardware Support ▾
🔍

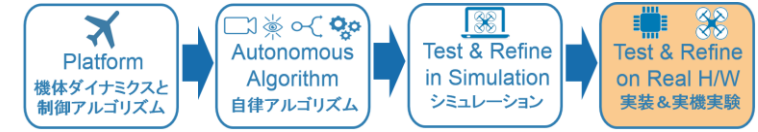
Overview | Search Hardware Support | Request Hardware Support

[Trial software](#)
[Contact sales](#)

⏪
Results 1 - 3 of 3
⏩

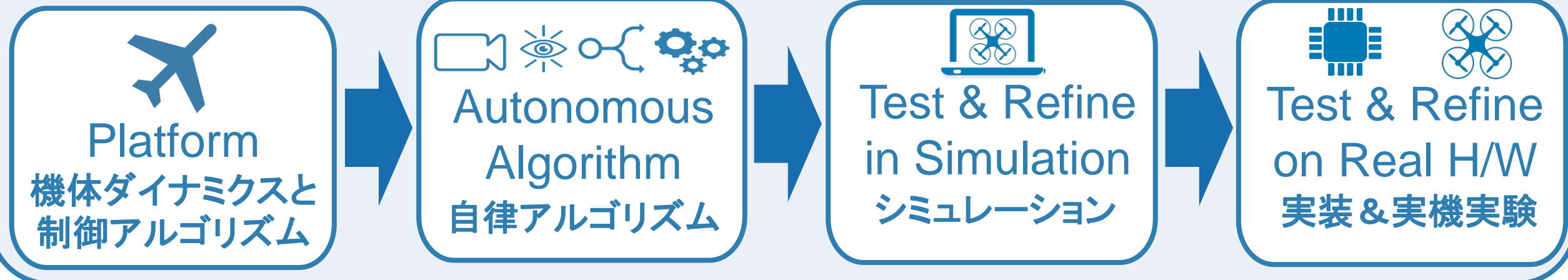
<p>Refine by Product</p> <p>Embedded Coder 2</p> <hr/> <p>Refine by Product Family and Category</p> <p>Simulink Product Family 3</p> <hr/> <p>Refine by Vendor</p> <p>3D Robotics 1</p> <p>PARROT 2</p> <hr/> <p>Refine by Application</p> <p>Control Systems 2</p> <p>Embedded Systems 2</p> <p>Image Processing and Computer Vision 1</p> <hr/> <p>Refine by Protocol or Standard</p> <p>Bluetooth 1</p> <p>USB 1</p>	<div style="margin-bottom: 15px;"> <p>PARROT Minidrones Support from Simulink 📄</p> <p>Highlights include: Automatically deploy algorithms designed in Simulink to PARROT Minidrones for flight control, optical flow, and image processing Learn about the sensors and algorithms needed to...</p> <p>Vendors: PARROT</p> <p>Tags: Support Package Installer Enabled, Run on Target Hardware, C/C++ Code Generation, Project-Based Learning, MathWorks Supported</p> </div> <hr/> <div style="margin-bottom: 15px;"> <p>Pixhawk PX4 Support from Embedded Coder</p> <p>Automatically build and deploy flight control algorithms to the Pixhawk Autopilot using Simulink and Embedded Coder</p> <p>Vendors: 3D Robotics</p> <p>Tags: RTOS, C/C++ Code Generation, Project-Based Learning</p> </div> <hr/> <div> <p>AR.Drone 2.0 Support from Embedded Coder</p> <p>...helicopter manufactured by PARROT® Is designed to be controlled from a mobile device, such as a tablet or phone Has a 720p front camera with 93° field of view lens, with 30 frames per second recording rate ...</p> <p>Vendors: PARROT</p> <p>Tags: C/C++ Code Generation, Project-Based Learning</p> </div>
--	---

自動コード生成の実装

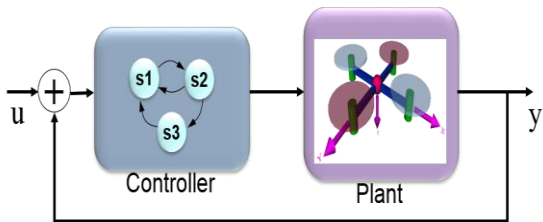


UAV開発プロセス

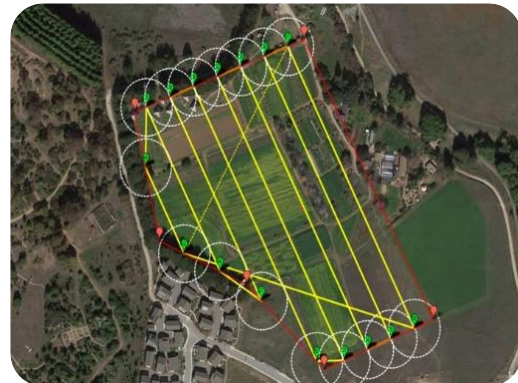
MATLAB/Simulink環境でのModel-Based Design



ステップ①:
機体ダイナミクスの
理解と飛行制御
アルゴリズムの設計



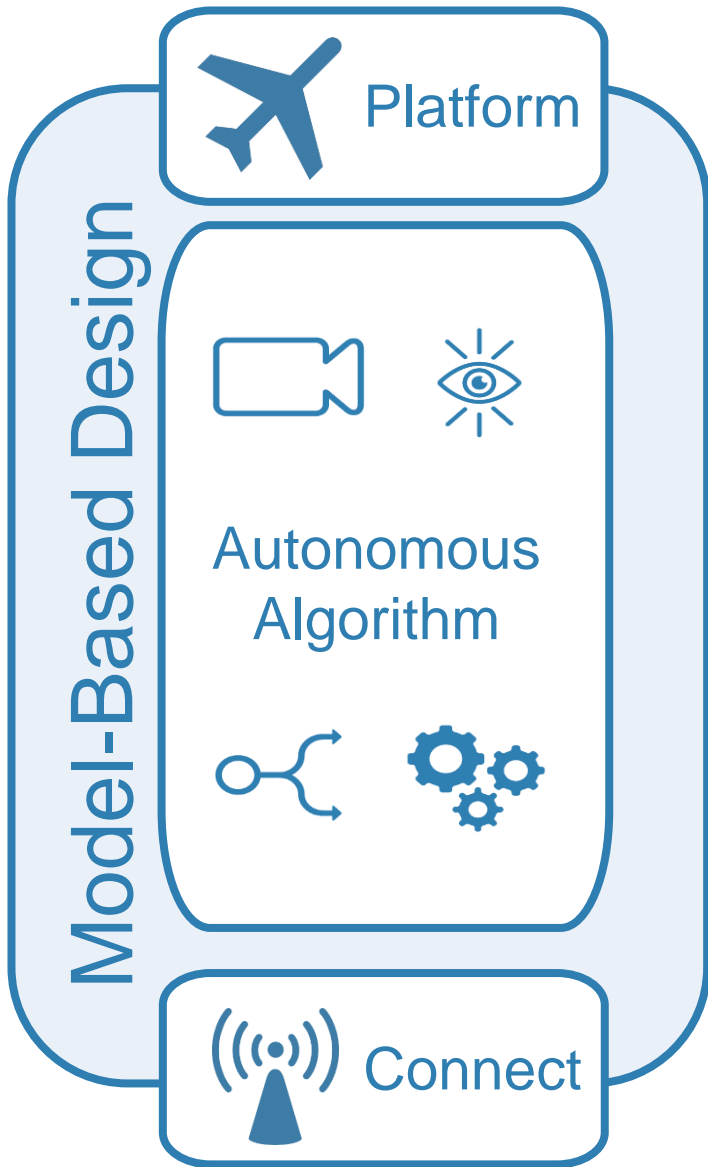
ステップ②:
ビジョン、レーダ、知覚
アルゴリズムの設計



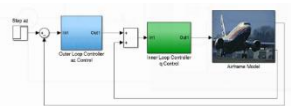
ステップ③:
アルゴリズムの
検証と実装



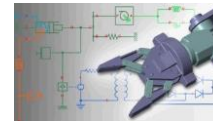
MATLAB/Simulinkは無人航空機システムに必要な要素を網羅



Control System Toolbox



Simscape



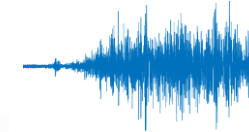
Simulink Real-Time



HW Support Packages



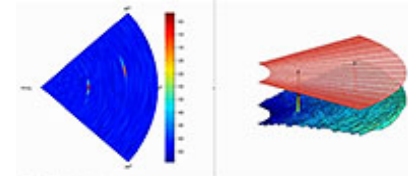
Data Acquisition Toolbox



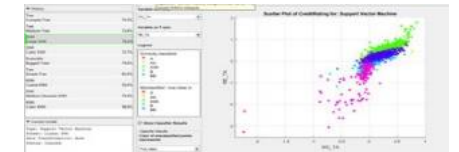
Computer Vision



Phased Array



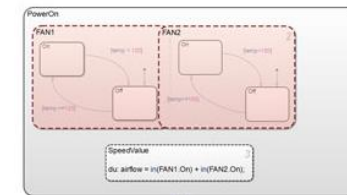
Statistics and Machine Learning



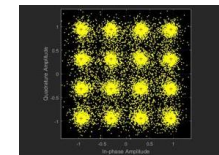
Robotics System Toolbox



Stateflow



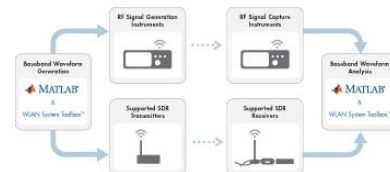
WLAN System Toolbox



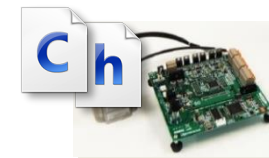
Robotics System Toolbox

ROS

Communications Toolbox



Embedded Coder



Autonomous System with MATLAB/Simulink

Airnamics Develops Unmanned Aerial System for Close-Range Filming with Model-Based Design

Challenge

Design and develop an unmanned aerial camera motion system for close-range aerial filming

Solution

Use Model-Based Design with MATLAB and Simulink to accelerate the design, debugging, and implementation of the vehicle's fly-by-wire and flight management system software

Results

- Time-to-market shortened by up to an order of magnitude
- Test flight anomalies quickly resolved
- Debugging time reduced from weeks to hours



Airnamics co-founders Marko Thaler and Zoran Bjelić with the R5 MSN1 prototype after its first flight.

“With Model-Based Design our three-engineer team found more than 95% of control software bugs before the first flight. We used the test flights to increase our Simulink models’ fidelity and isolate remaining bugs with high precision. The result is a safer, more reliable, and higher-quality product.”

Marko Thaler
Airnamics

Autonomous System with MATLAB/Simulink

BAE Systems Controls Develops Autopilot for Unmanned Aerial Vehicle Using MathWorks Tools

Challenge

Enable teams working in separate locations to design a sophisticated UAV autopilot system quickly and inexpensively

Solution

Use MathWorks tools, modify existing software designs with Model-Based Design, and automatically generate embedded control code

Results

- Design and rework costs substantially reduced
- Testing cycle time minimized
- Coding errors and manual documentation work minimized



An Eagle 150 unmanned aerial vehicle flight.

(Image courtesy of Composites Technology Research Malaysia.)

“MATLAB and Simulink greatly reduced development cycle time and cut system software design and testing costs by 50%.”

**Feng Liang
BAE Systems Controls**

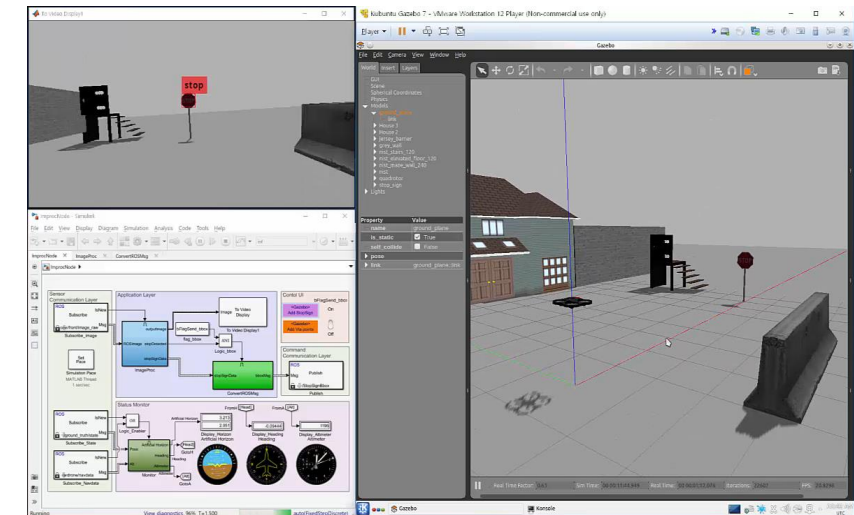
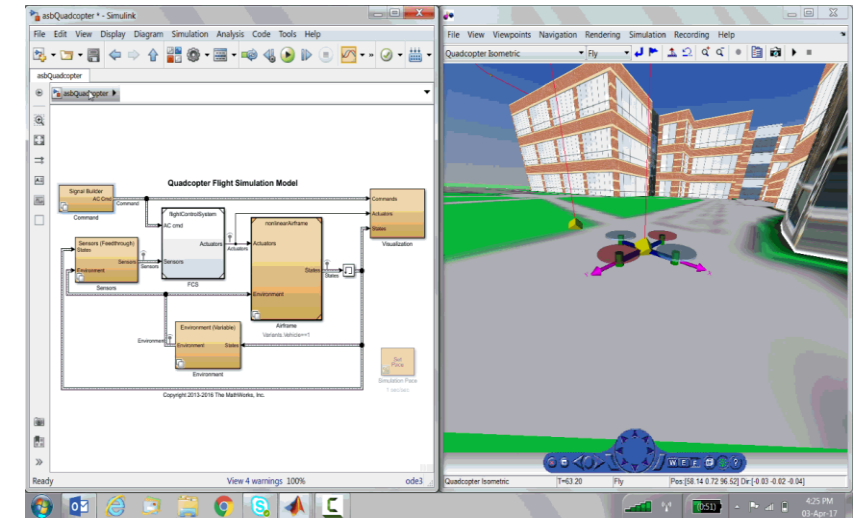
まとめ

**自律アルゴリズムの設計・検証には
システムシミュレーションが重要！**

MATLAB/SimulinkによるUAV設計・開発のサポート:

- プラント・コントローラモデリングによるダイナミクス評価
- ガイダンス・視覚アルゴリズム設計・検証の容易化
- シミュレーションでの動作確認後に自動コード生成・実装

**MATLAB/Simulink:
UAV設計・開発の統一環境！**



Thank You For Your Attention
ご清聴ありがとうございました

© 2017 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.