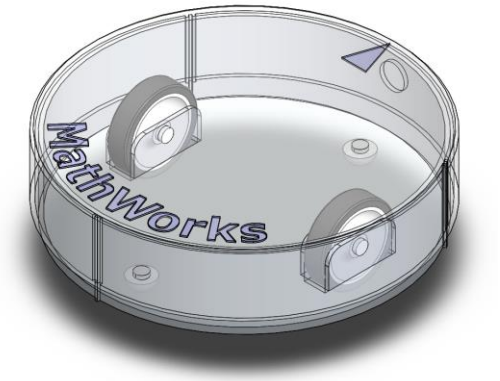
UAV/AGV、空飛ぶクルマ開発に向けた MathWorksソリューションの概要ご紹介

MathWorks Japan
アプリケーションエンジニアリング部
アプリケーションエンジニア
能戸 フレッド・Fred Noto

自律システムの普及...AGV（Automated Ground Vehicle）



Clearpath Robotics Accelerates Algorithm Development for Industrial Robots

Challenge

Shorten development times for laser-based perception, computer vision, fleet management, and control algorithms used in industrial robots

Solution

visualize ROS data,
ly the latest advances

An OTTO self-driving vehicle from Clearpath Robotics.

up to 50%
improved
hms quickly

"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

https://jp.mathworks.com/company/user_stories/clearpath-robotics-accelerates-algorithm-development-for-industrial-robots.html..html

Israel Aerospace Industries Develops DO-178B Level B Certified Software for a Hybrid-Electric Aircraft Tractor

Challenge

Develop control software for the world's first certified aircraft-taxiing vehicle

Solution

Use Model-Based Design to model the control loops, application logic, and plant; run simulations and HIL tests; and generate DO-178B certified production code

Results

- Development time halved
- 50% of models reused
- DO-178B certification simplified

The TaxiBot from Israel Aerospace Industries

"We initially intended to develop only control loops with Model-Based Design, but the process proved so efficient that we decided to use Model-Based Design for the application layer as well. Being able to run a model, see that it's working right, and then generate certifiable code is a big advantage."
- Zeev Gabbin, Israel Aerospace Industries

https://jp.mathworks.com/company/user_stories/israel-aerospace-industries-develops-do-178b-level-b-certified-software-for-a-hybrid-electric-aircraft-tractor.html

自律システムの普及...UAV（Unmanned Air Vehicle）

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
- Development time reduced by one year or more
- Coding errors eliminated
- 80% model reuse achieved

"With Model-Ba
than 95% of co
used the test fli
isolate remaini
more reliable, a
- Marko Thaler, A

https://jp.mathworks.com/company/user_stories/airnamics-develops-unmanned-aerial-system-for-close-range-filming-with-model-based-design.html

Korean Air Speeds UAV Flight Control Software Development and Verification with Model-Based Design

Challenge
Develop and verify flight control software for unmanned aerial vehicles

Solution
Use Model-Based Design to design and simulate flight control laws and operational logic, generate and verify production code, and conduct HIL tests

Results
- 100% of run-time errors in handwritten code identified and eliminated
- Development effort reduced by 60%
- Costly flight tests minimized

A Korean Air unmanned aerial vehicle.

"The model reuse and efficiency improvements enabled by MATLAB and Simulink save time and lower costs. We estimate a time savings of more than 50% is achievable with Model-Based Design compared with hand-coding, and the advantages of Model-Based Design increase with the complexity of the project."
- Jungho Moon, Korean Air

https://jp.mathworks.com/company/user_stories/korean-air-speeds-uav-flight-control-software-development-and-verification-with-model-based-design.html

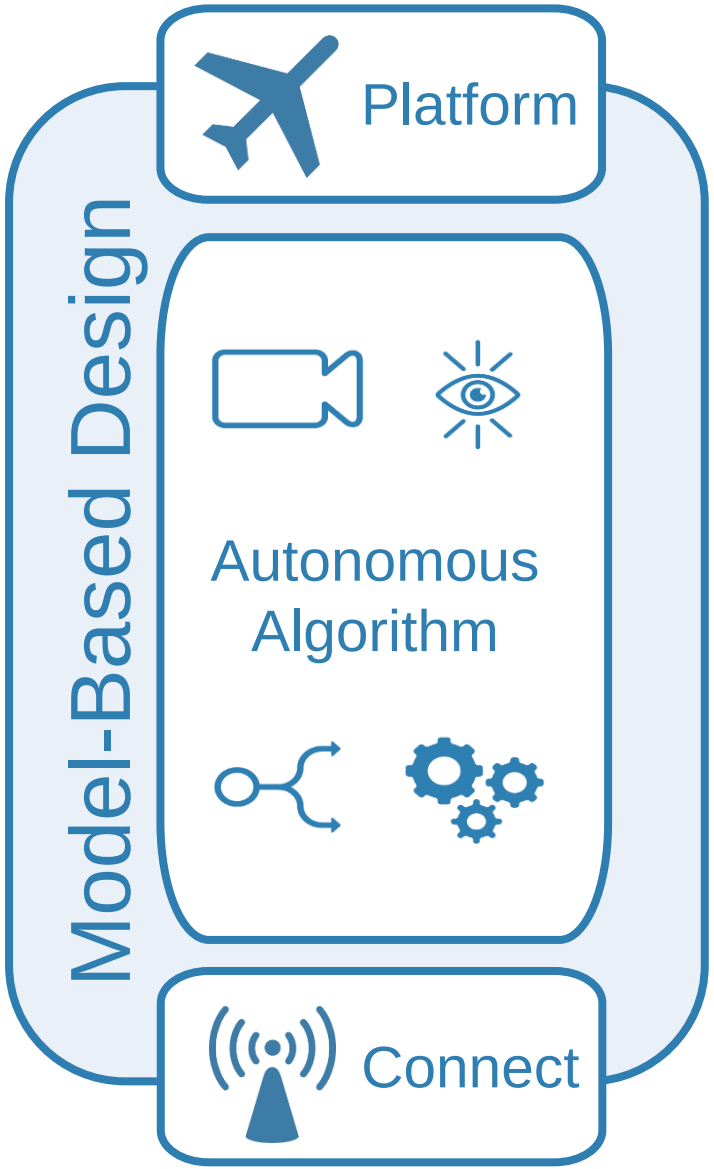
自律システムの普及…「空飛ぶクルマ」に向けて

自律アルゴリズムの役割が増加しつつ、より高いシステム安全性が要求されます

テクノロジー

テクノロジーの融合により
「空飛ぶクルマ」を実現

自律システム開発に対する提案



本日のトピックス

- 自律システムの開発プロセス
- システム安全性の評価（DO-178C）

メッセージ

MATLAB®/Simulink®の統一環境での自律システム開発・評価

自律システム開発プロセス

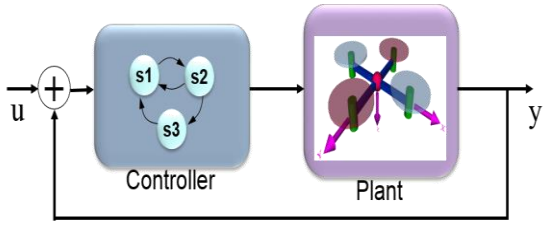
Platform
機体ダイナミクスと制御アルゴリズム

Autonomous Algorithm
自律アルゴリズム

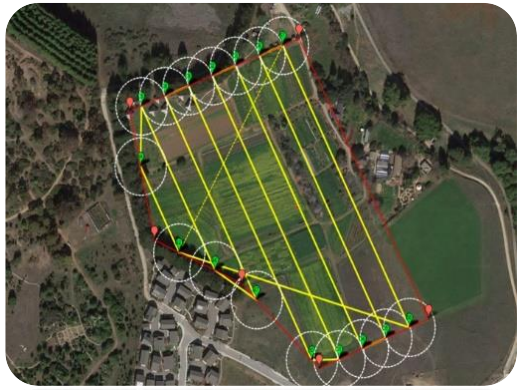
Test & Refine in Simulation
シミュレーション

Test & Refine on Real H/W
実装＆実機実験

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

自律システム開発プロセス

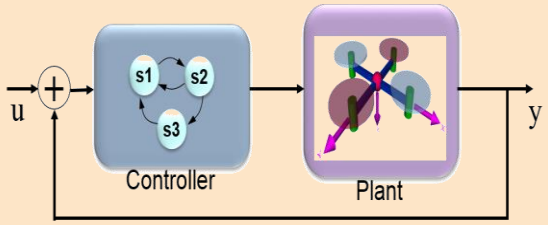
Platform
機体ダイナミクスと
制御アルゴリズム

Autonomous
Algorithm
自律アルゴリズム

Test & Refine
in Simulation
シミュレーション

Test & Refine
on Real H/W
実装＆実機実験

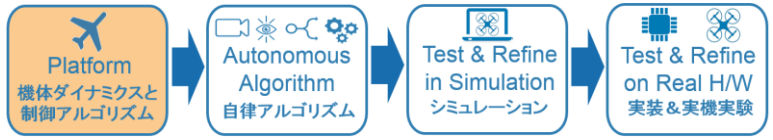
ステップ①：
機体ダイナミクスの理解と飛行制御アルゴリズムの設計



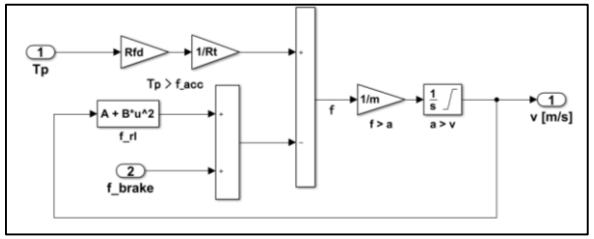
ステップ②：
ビジョン、レーダ、知覚
アルゴリズムの設計・検証

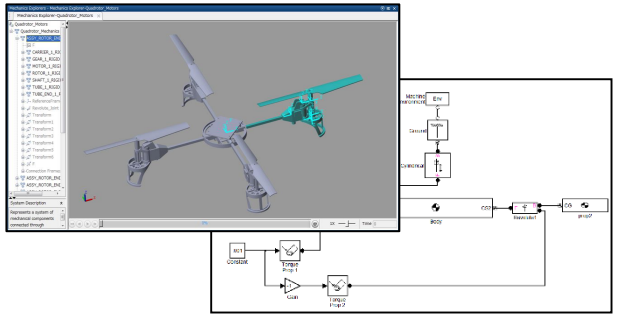
ステップ③：
アルゴリズムの
検証と実装

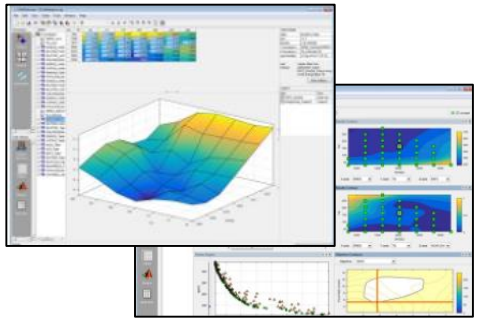
MATLAB/Simulinkによるプラントモデリング

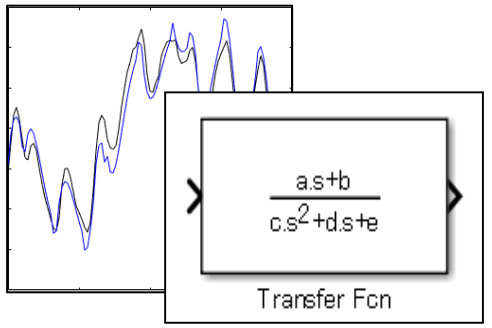


Simulink環境で様々な作成方法を柔軟に対応









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

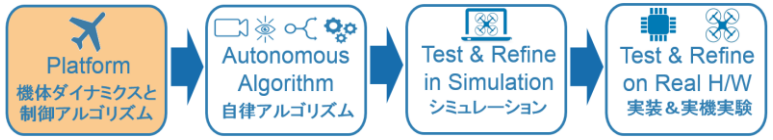
パラメータチューニング
Simulink Design Optimization™

統計的手法
Model-Based Calibration Toolbox™

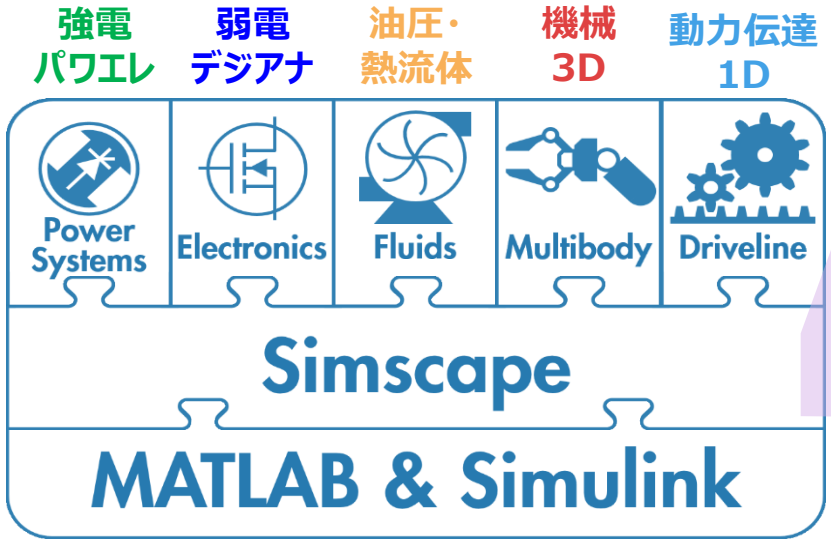
ニューラルネットワーク
Neural Network Toolbox™

システム同定
System Identification Toolbox™

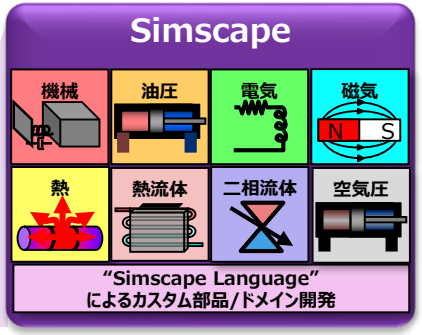
Simscapeによる制御対象の物理モデリング

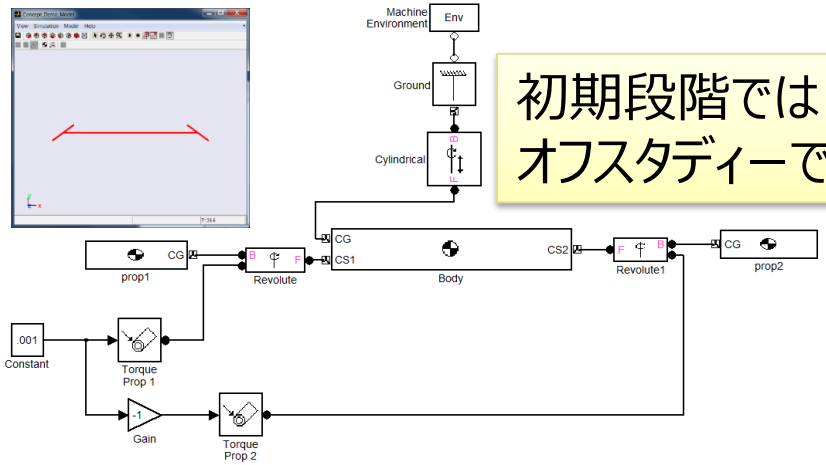


数式不要で直感的にコンポーネントの追加・削除が容易

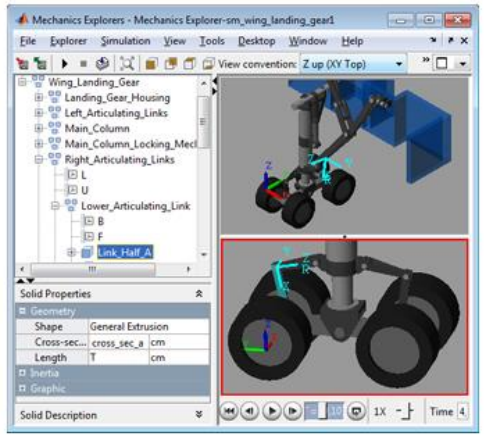


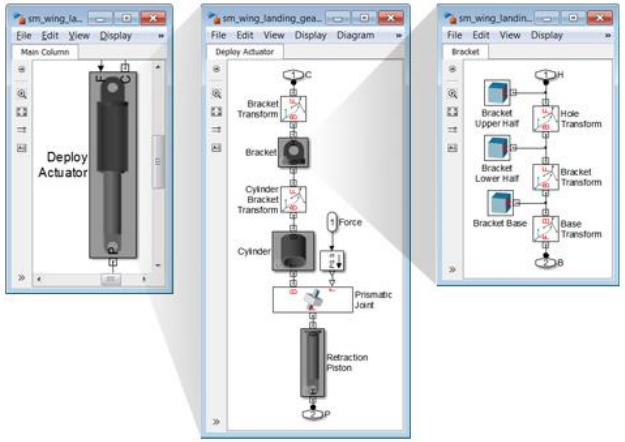
基本部品ライブラリ

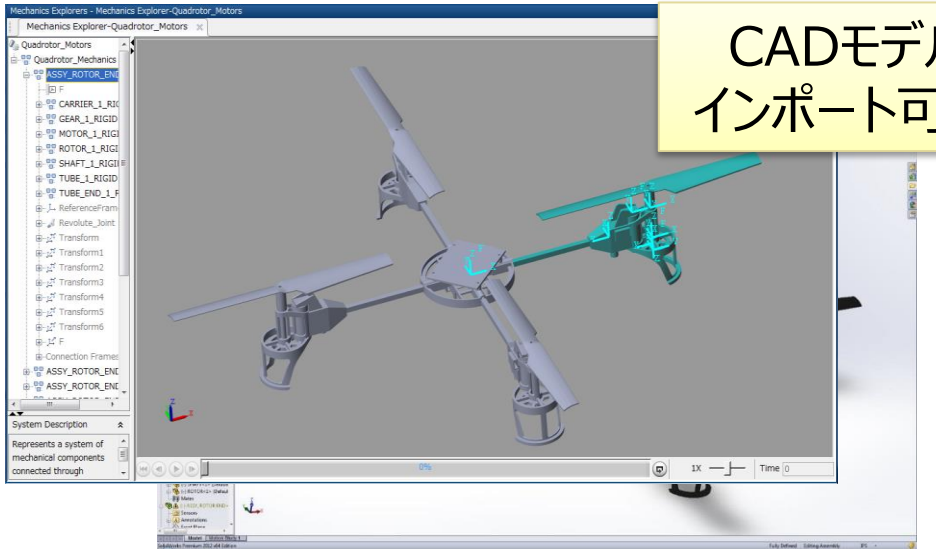




初期段階ではトレード
オフスタディーで活用





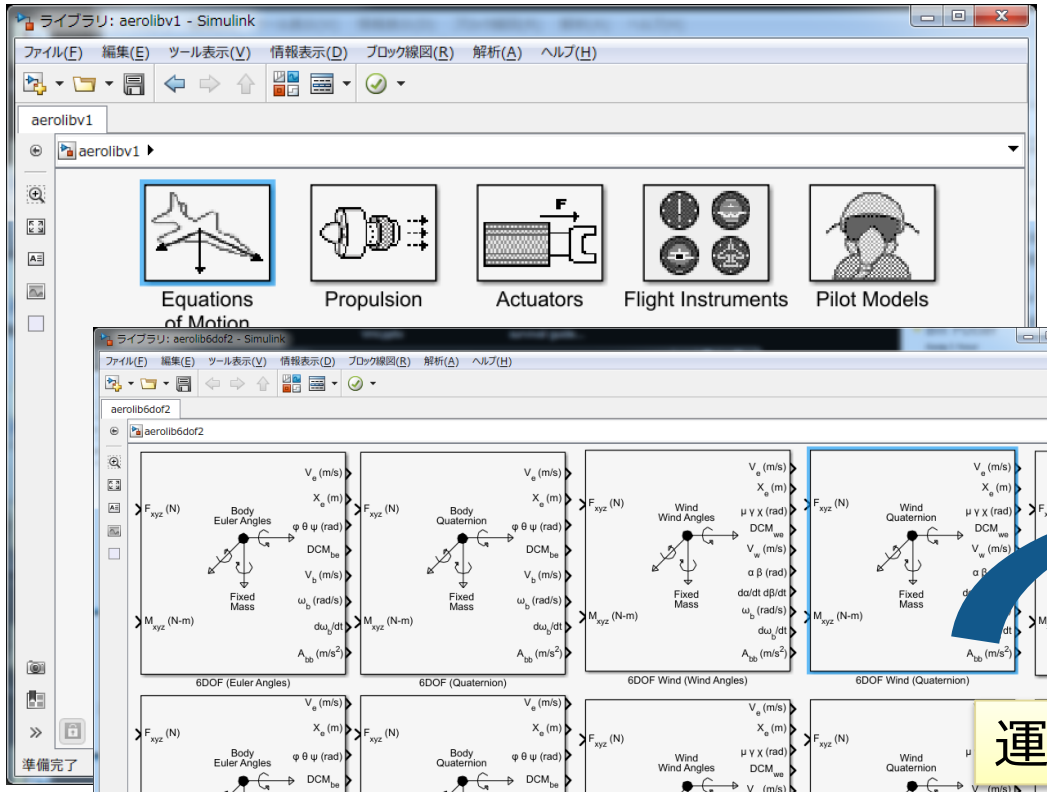


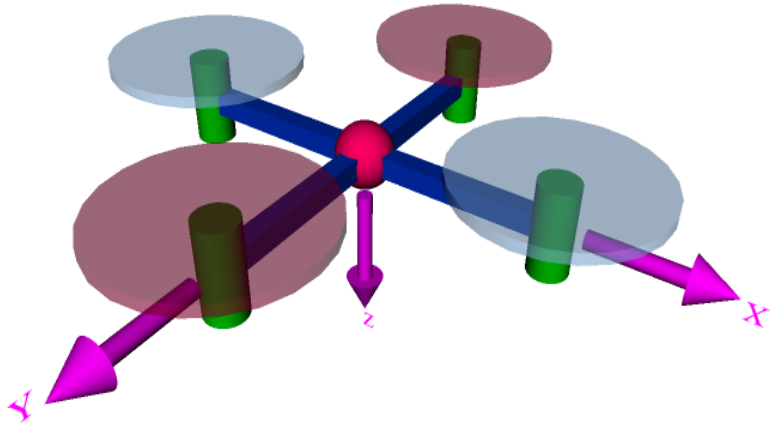
CADモデル
インポート可能

Aerospace Blockset™でのダイナミクスモデリング

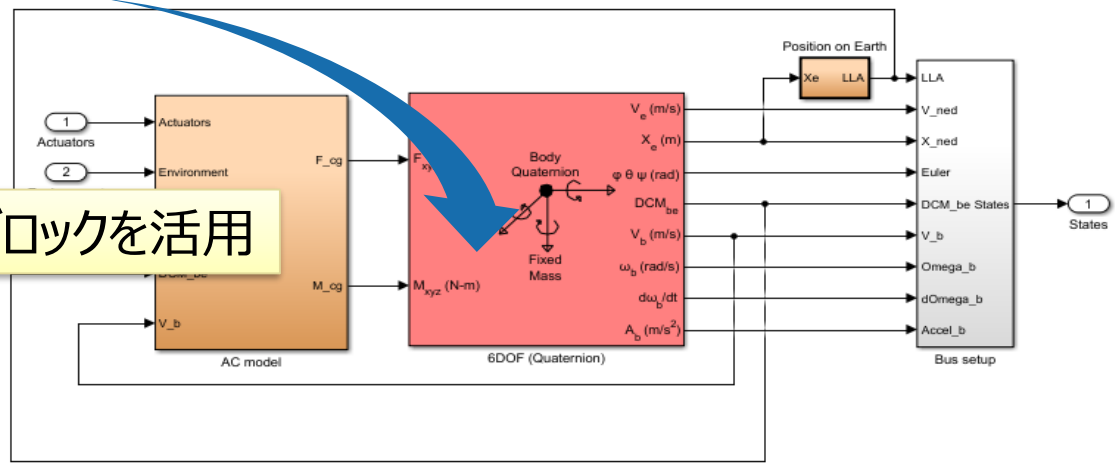
Platform 機体ダイナミクスと制御アルゴリズム → Autonomous Algorithm 自律アルゴリズム → Test & Refine in Simulation シミュレーション → Test & Refine on Real H/W 実装＆実機実験

航空機コンポーネンツのモデリングを簡略化・効率化

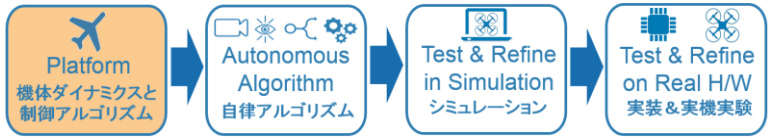




運動方程式ブロックを活用



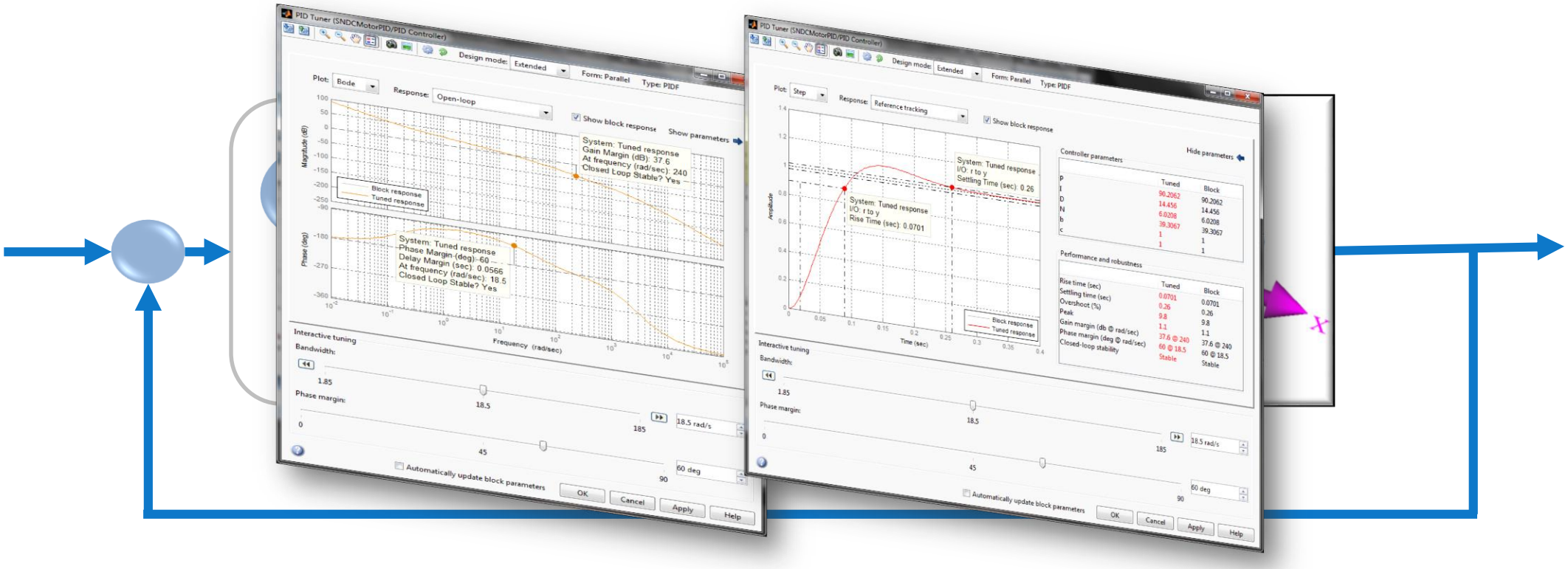
制御アルゴリズムの設計



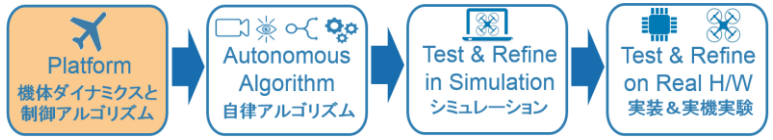
プラントとコントローラを同一環境で組み合わせたシステムレベルの動作検証

制御系設計ツールのご活用による性能向上

Simulink Control Design™ 、Control System Toolbox™、Robust Control Toolbox™



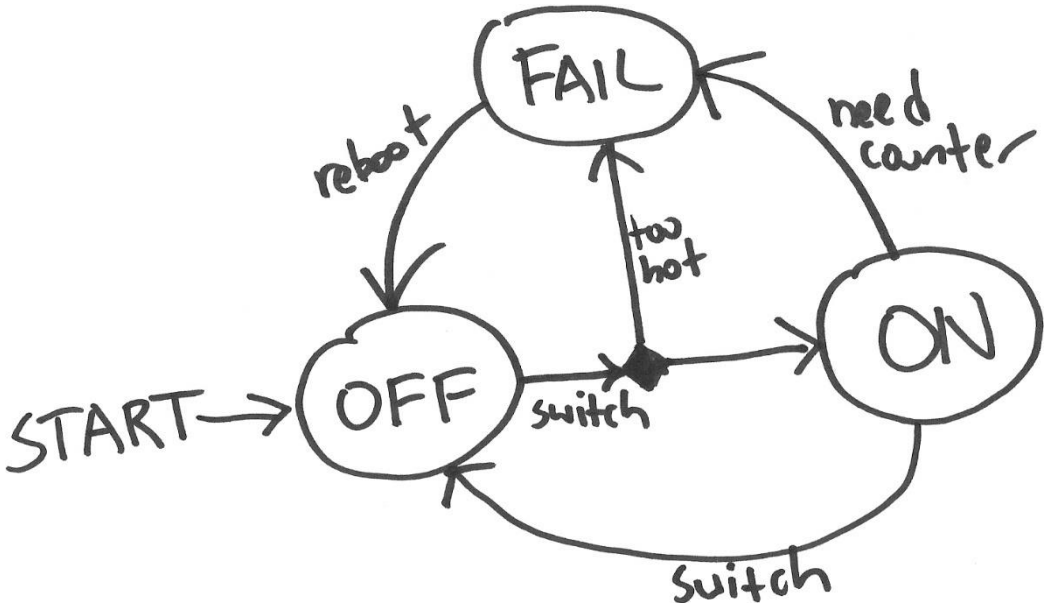
制御アルゴリズムの設計

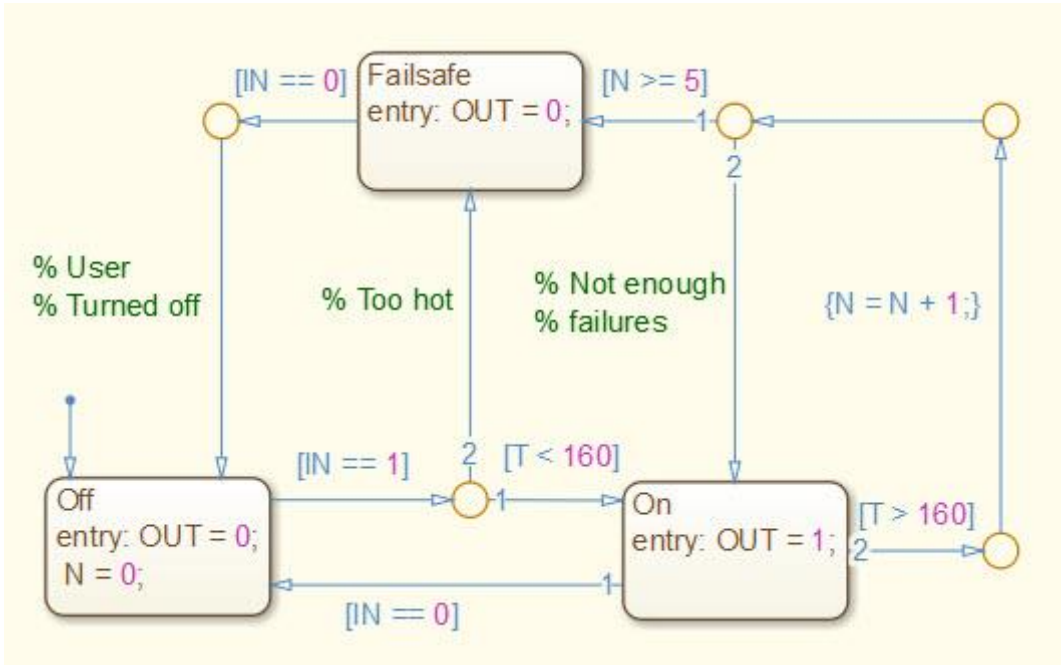


Stateflow®を活用したモードロジック等の瞬間的変化の設計を容易化

・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間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

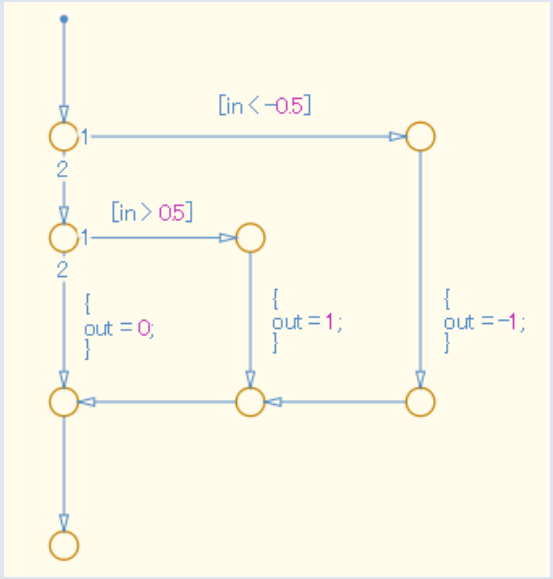




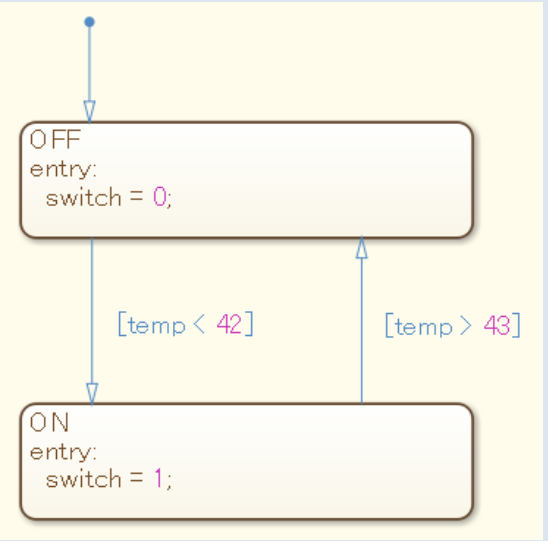
Stateflowによる色々なロジック表現

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

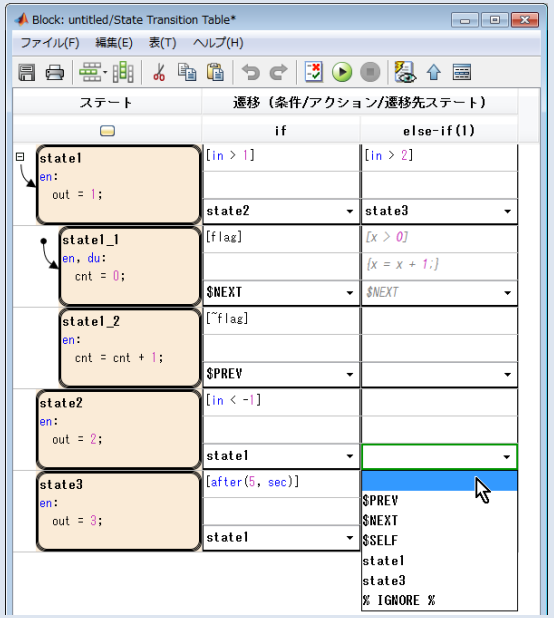
フローチャート



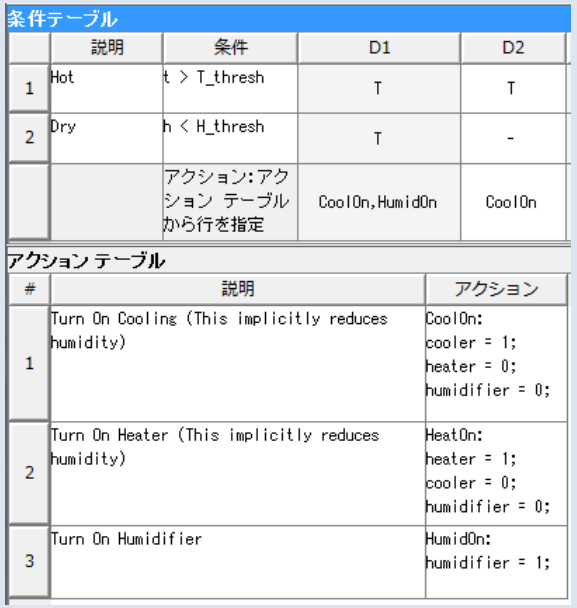
状態遷移図



状態遷移表



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

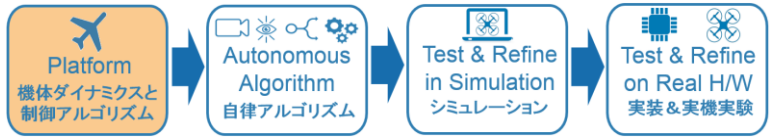
#	説明	アクション

1	Turn On Cooling (This implicitly reduces humidity)	CoolOn:
		cooler = 1;
		heater = 0;
		humidifier = 0;

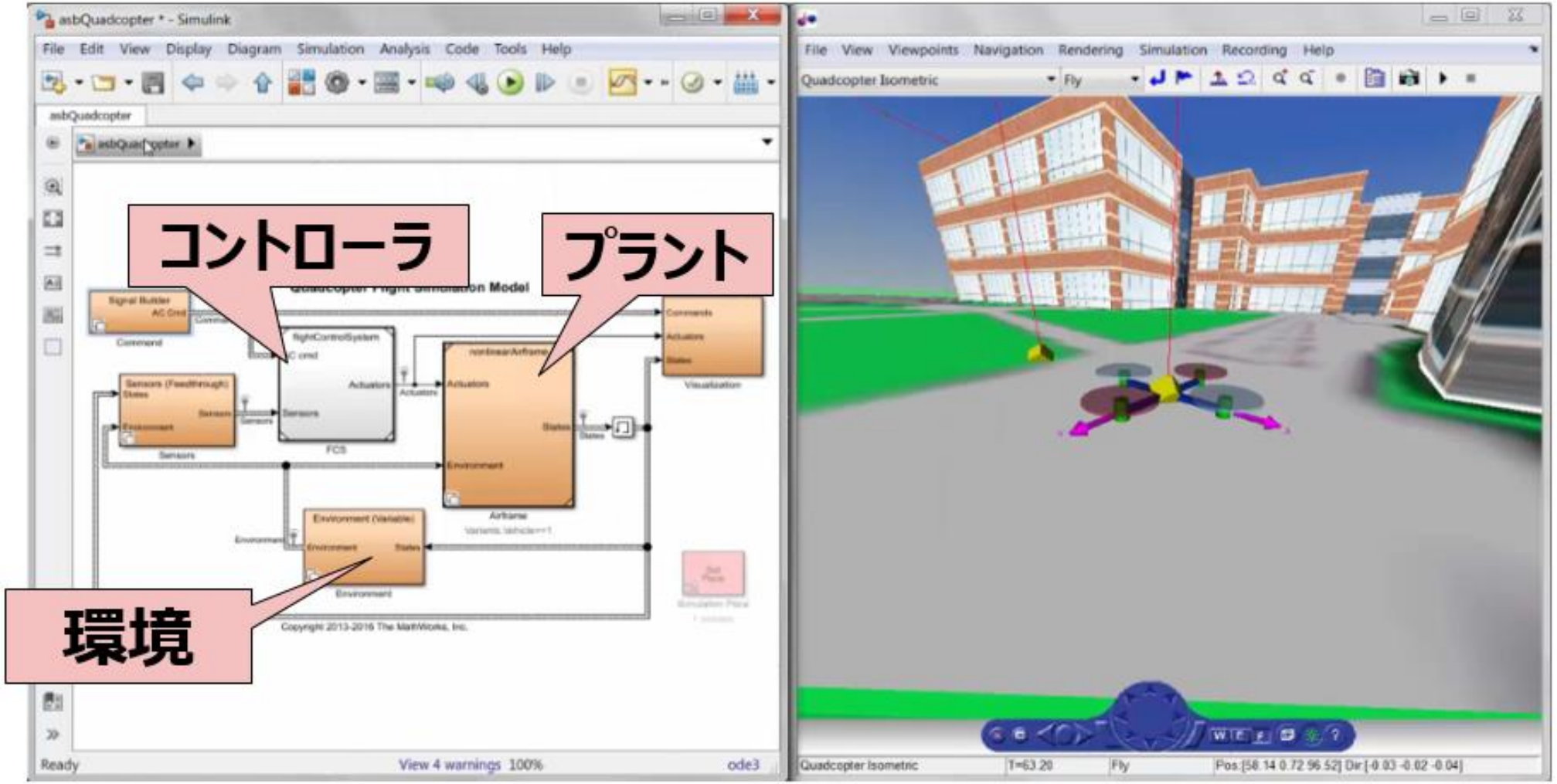
2	Turn On Heater (This implicitly reduces humidity)	HeatOn:
		heater = 1;
		cooler = 0;
		humidifier = 0;

3	Turn On Humidifier	HumidOn: humidifier = 1;

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



自律システム開発プロセス

Platform
機体ダイナミクスと制御アルゴリズム

Autonomous Algorithm
自律アルゴリズム

Test & Refine in Simulation
シミュレーション

Test & Refine on Real H/W
実装＆実機実験

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

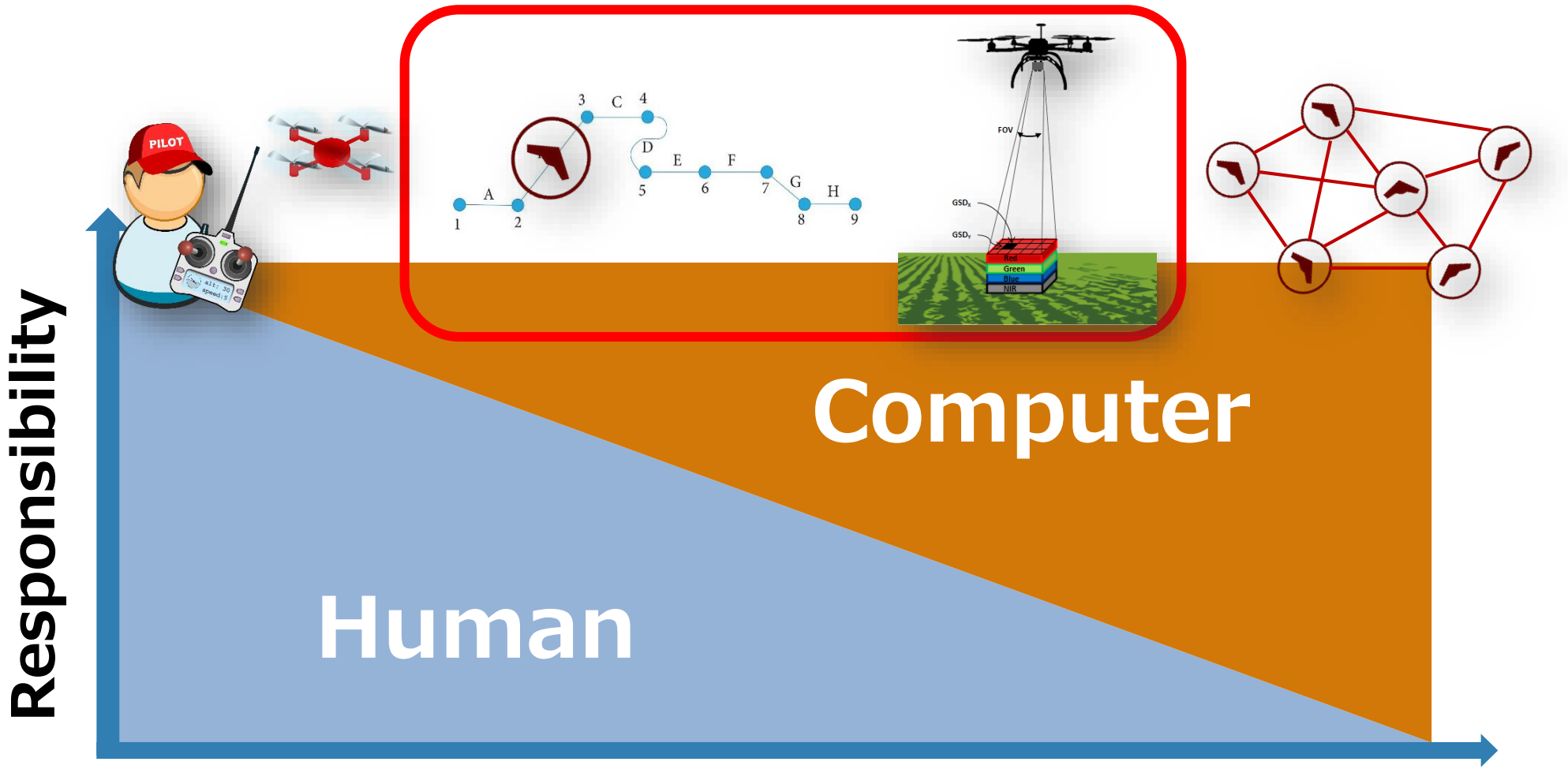
ステップ②：
ビジョン、レーダ、知覚アルゴリズムの設計

ステップ③：
アルゴリズムの検証と実装

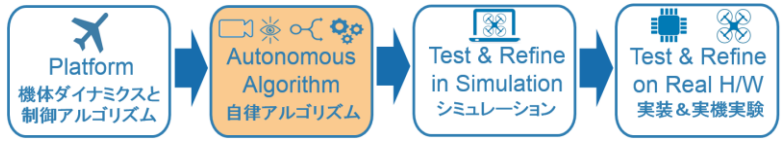
自律テクノロジーのトレンド

Platform 機体ダイナミクスと制御アルゴリズム → Autonomous Algorithm 自律アルゴリズム → Test & Refine in Simulation シミュレーション → Test & Refine on Real H/W 実装＆実機実験

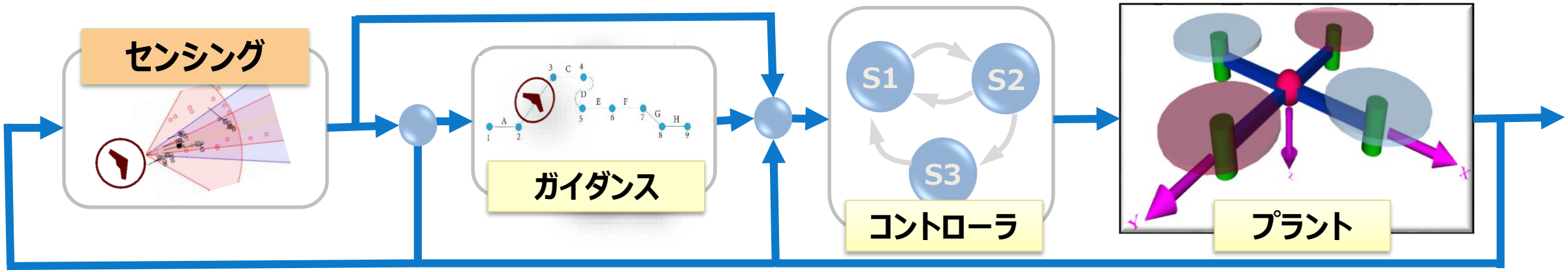
設計したいアルゴリズムに応じて様々なデザインスタイルを活用できます



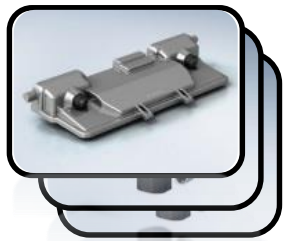
自律アルゴリズムの追加

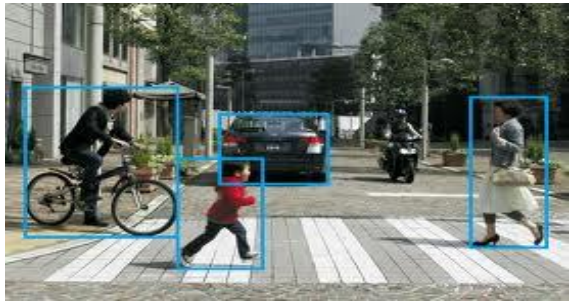


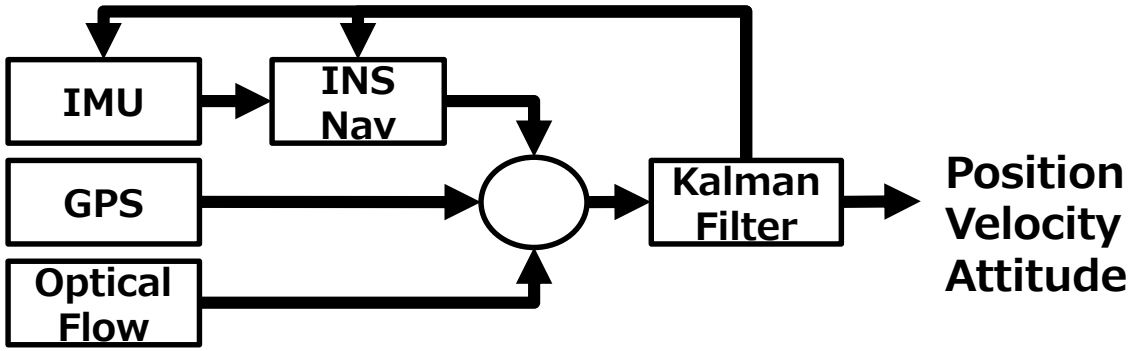
ガイダンス＆視覚アルゴリズムによる自律機能の増加



カメラ,
レーダー,
LiDAR,
GPS,
IMU,
, …



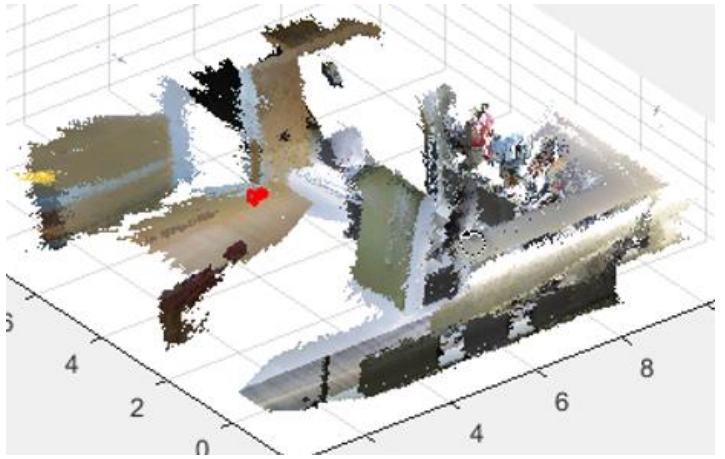


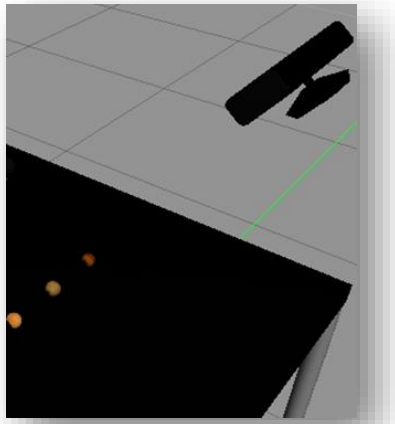


センサーデータの取得

Image Acquisition Toolbox™

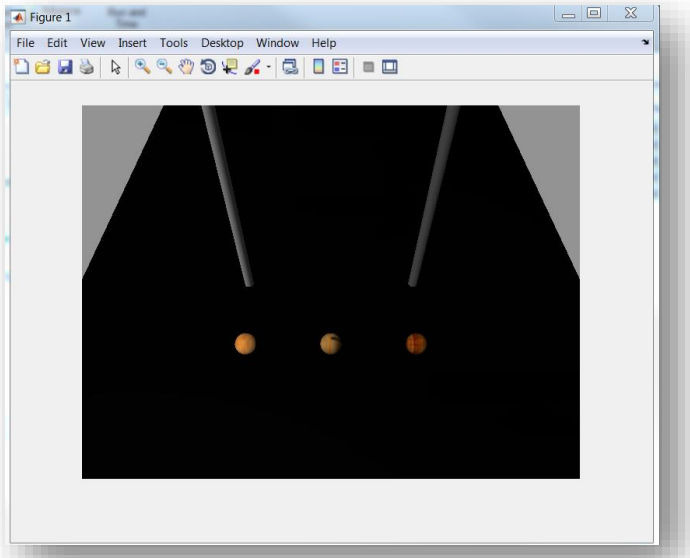
直接ハードウェアからデート取得





MATLAB/Simulink

保存データの読み込み

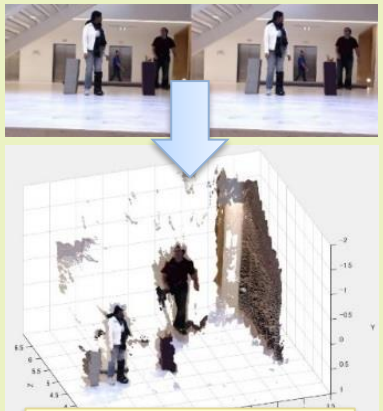


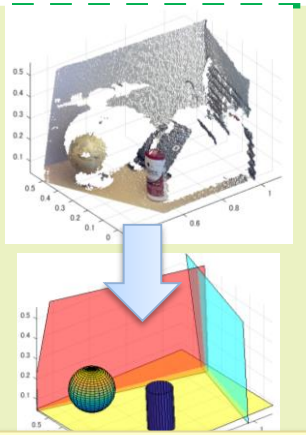
Robotics System Toolbox™

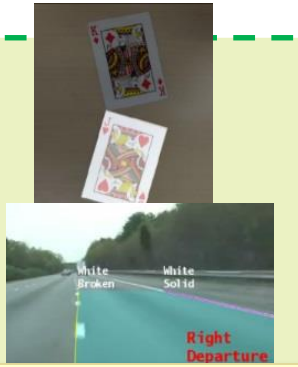
ROSによるデータ通信

MATLABによる認識ソリューション

Platform 機体ダイナミクスと制御アルゴリズム → Autonomous Algorithm 自律アルゴリズム → Test & Refine in Simulation シミュレーション → Test & Refine on Real H/W 実装＆実機実験

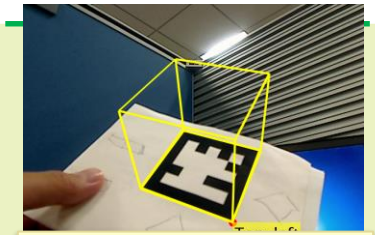

ステレオビジョン

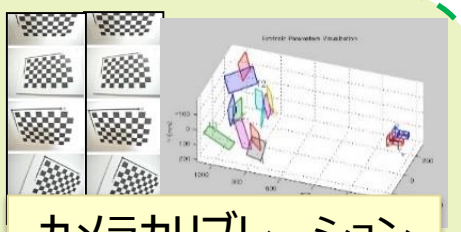

3次元点群処理



物体の検出

文字認識(OCR)


AR(拡張現実)


カメラカリブレーション

顔、人物認識

Image Processing Toolbox™
Computer Vision System Toolbox™

画像検索、分類(BoF)


機械学習による分類

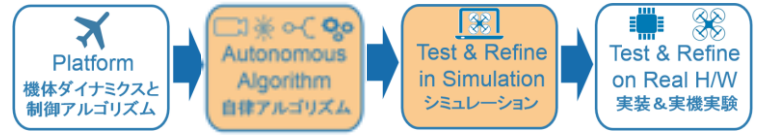
ディープラーニング (CNN/Faster R-CNN)

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™

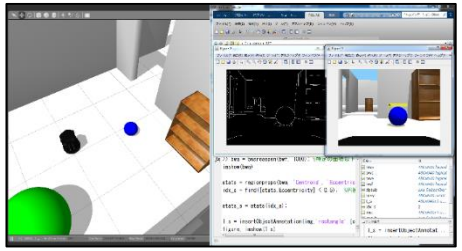
MATLABとROSによる自律制御システム開発



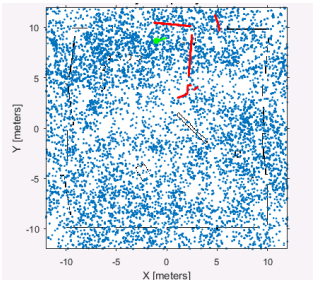
MATLAB/Simulinkの柔軟な開発環境とROSを連携しロボティクス開発を加速化

Robotics System Toolbox

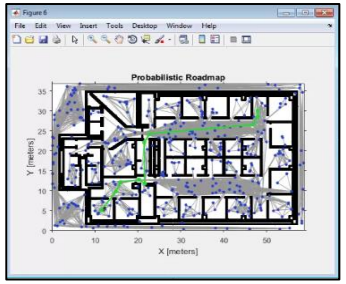
- ロボットアルゴリズム開発の支援
  - 移動体向けアルゴリズム
    - 障害物データ表現、位置推定、パスプランニング
  - ロボットアーム向けアルゴリズム
    - ツリー構造表現、逆運動学解析
  - ユーティリティ関数
    - オイラー角、クォータニオン、座標変換
- ROSのインターフェイス提供
  - MATLABをROSマスター、ノードとして起動
  - 直接ROSネットワークに接続して検証
  - ROSノード生成
    - SimulinkモデルからC++ ROSノードを生成



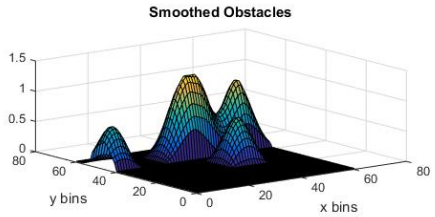
シミュレータや実機とのROS連携



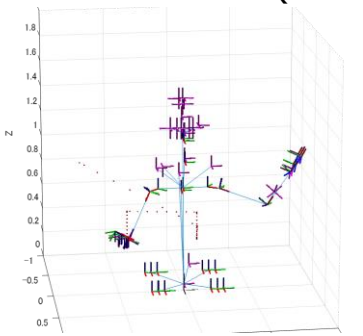
自己位置推定(AMCL)



確率的ロードマップ法(PRM)



衝突回避(VFH+)



ロボットマニピュレーターアルゴリズム開発

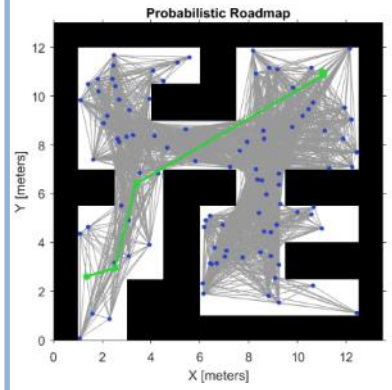
Robotics System Toolboxを使用したROS連携

Platform 機体ダイナミクスと制御アルゴリズム → Autonomous Algorithm 自律アルゴリズム → Test & Refine in Simulation シミュレーション → Test & Refine on Real H/W 実装＆実機実験

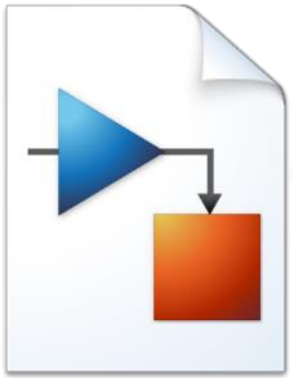
ROSとの連携によるロボティクス開発加速化

MATLAB

MATLABコード



すぐに使える
組み込み
アルゴリズム



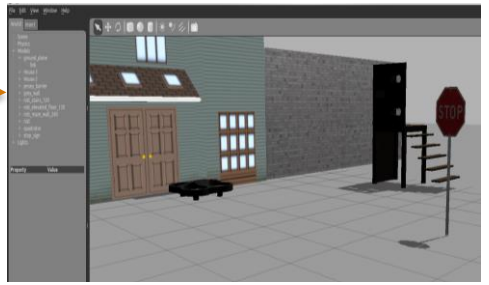
Simulinkモデル

ネットワーク

ROS

Cコード生成による
ROSノード生成・組み込み

実機



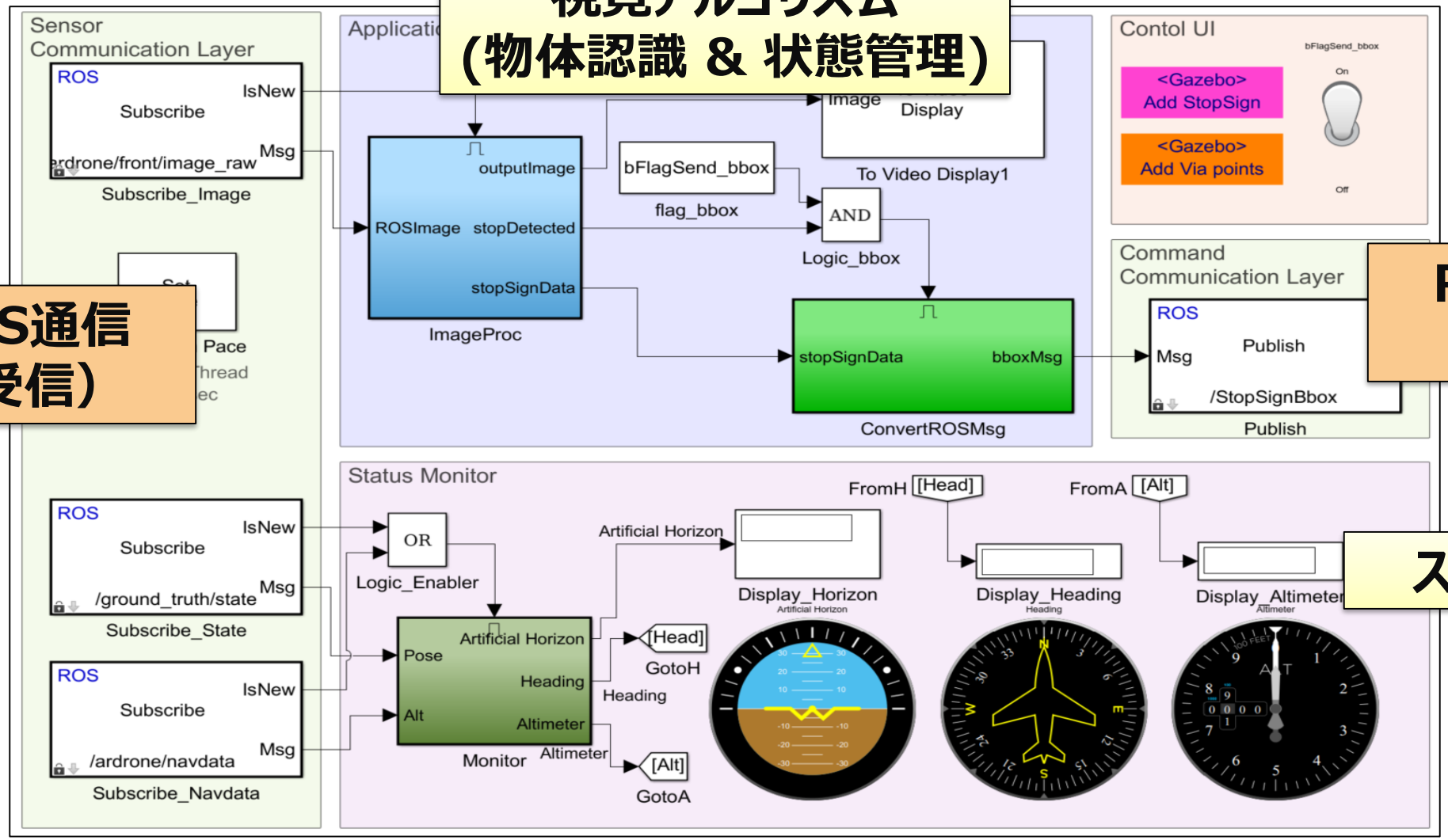
シミュレーション環境

ROS ノード

MATLAB・Simulink・ROSによるシステム設計

Platform 機体ダイナミクスと制御アルゴリズム → Autonomous Algorithm 自律アルゴリズム → Test & Refine in Simulation シミュレーション → Test & Refine on Real H/W 実装＆実機実験

視覚アルゴリズム (物体認識 & 状態管理)

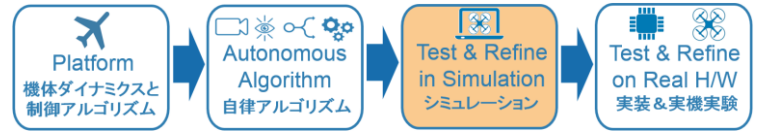


ROS通信（受信）

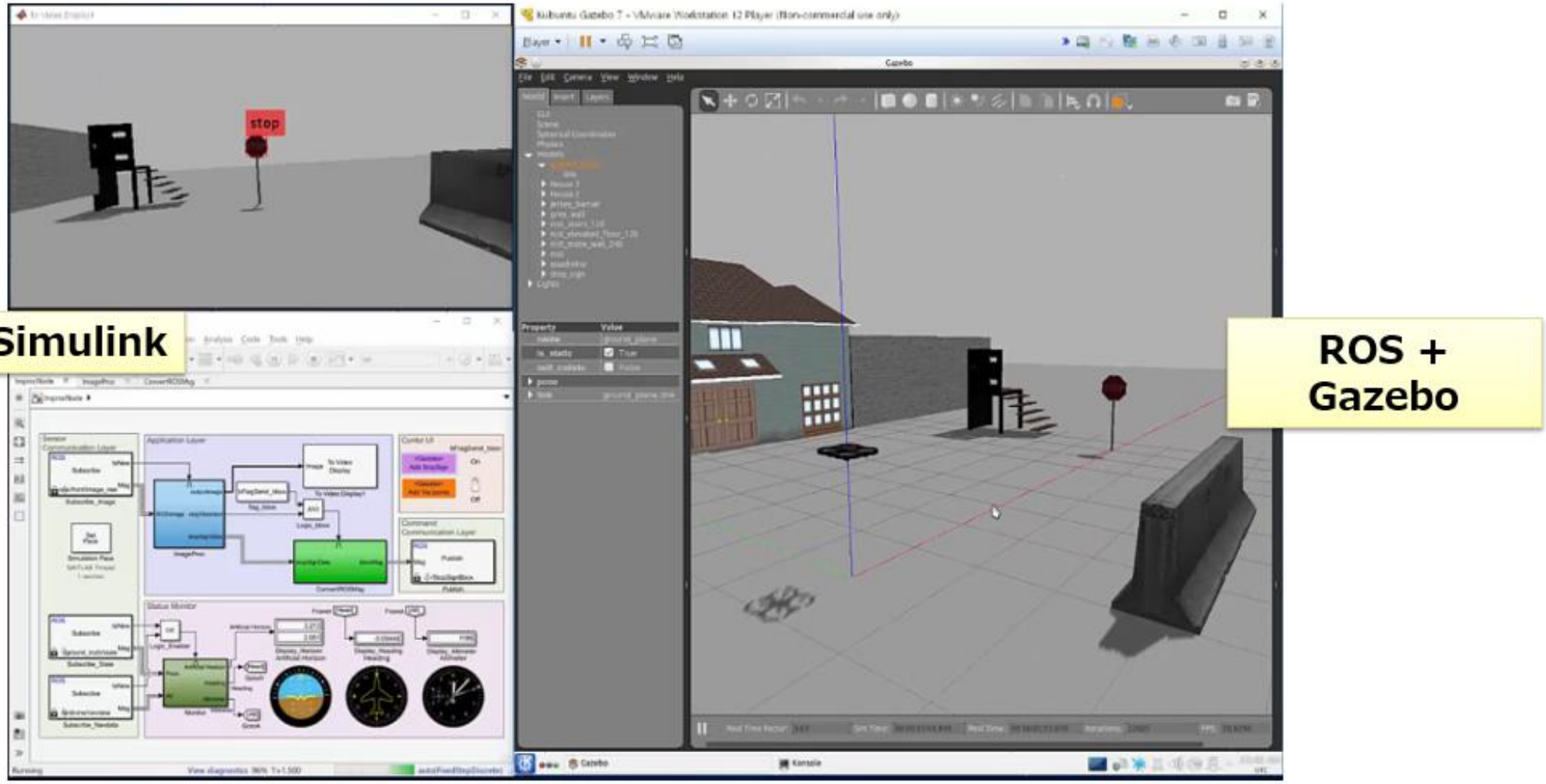
ROS通信（送信）

ステータス

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



自律システム開発プロセス

Platform
機体ダイナミクスと制御アルゴリズム

Autonomous Algorithm
自律アルゴリズム

Test & Refine in Simulation
シミュレーション

Test & Refine on Real H/W
実装＆実機実験

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

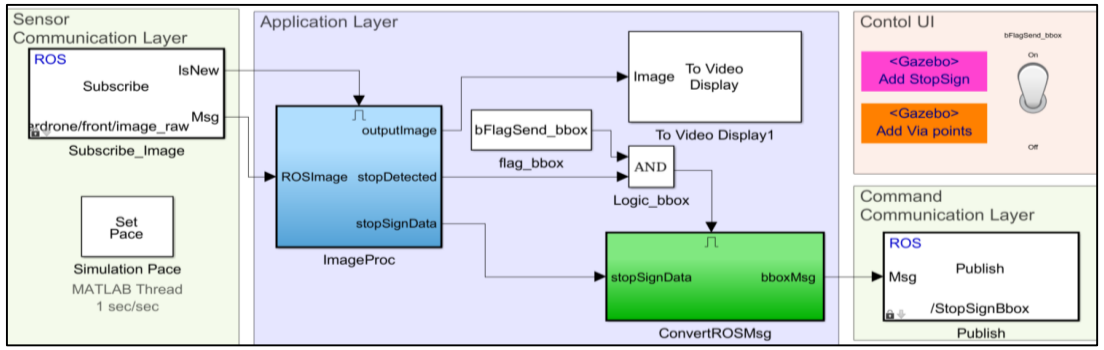
ステップ②：
ビジョン、レーダ、知覚アルゴリズムの設計

ステップ③：
アルゴリズムの検証と実装

コード生成・実装

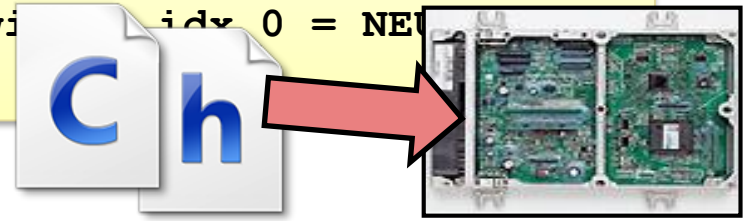
Platform 機体ダイナミクスと制御アルゴリズム → Autonomous Algorithm 自律アルゴリズム → Test & Refine in Simulation シミュレーション → Test & Refine on Real H/W 実装&実機実験

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



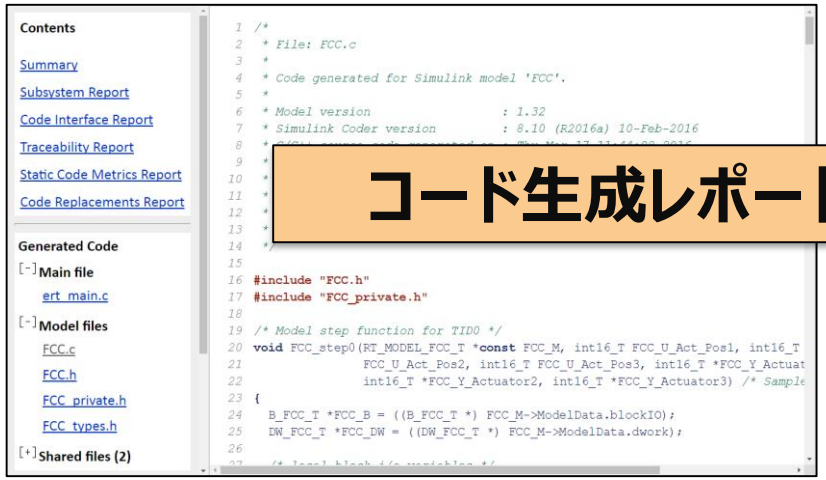

```
if (LeftPos > 4.0F) {
  rtb_Switch1_idx_0 = LOCK;
} else if (Left
  rtb_Switch1_i
} else {
  rtb_Swi        idx_0 = NE
}
```

C/C++コード生成



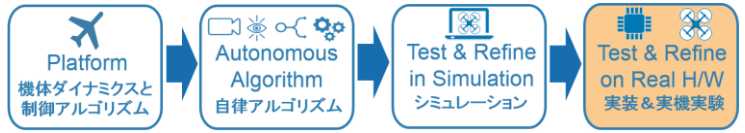
ROSノードC/C++コード生成

ROS ノード



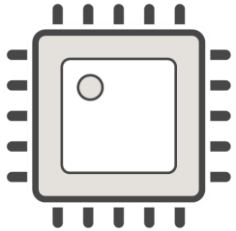
コード生成レポート

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



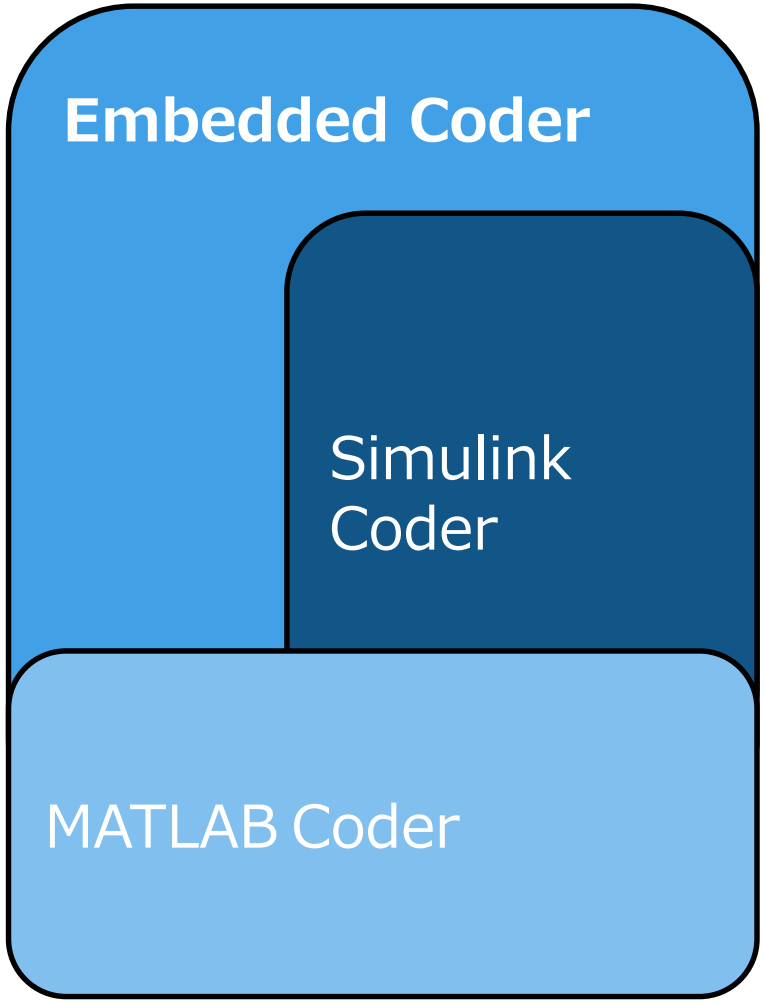
Simulink Coder™

- 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

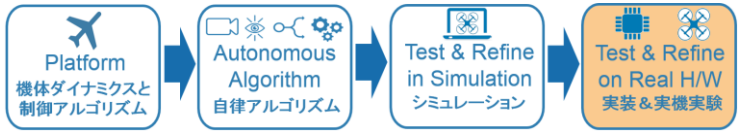
Embedded Coder™

- 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



自動コード生成の実装


Platform
機体ダイナミクスと
制御アルゴリズム
Autonomous
Algorithm
自律アルゴリズム
Test & Refine
in Simulation
シミュレーション
Test & Refine
on Real H/W
実装＆実機実験

自律システム開発プロセス

MATLAB/Simulink環境でのModel-Based Design

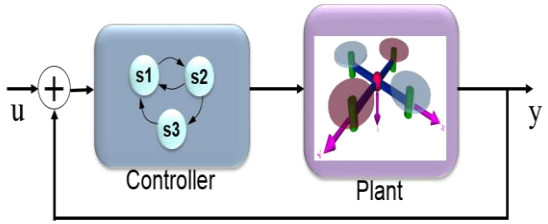
Platform
機体ダイナミクスと制御アルゴリズム

Autonomous Algorithm
自律アルゴリズム

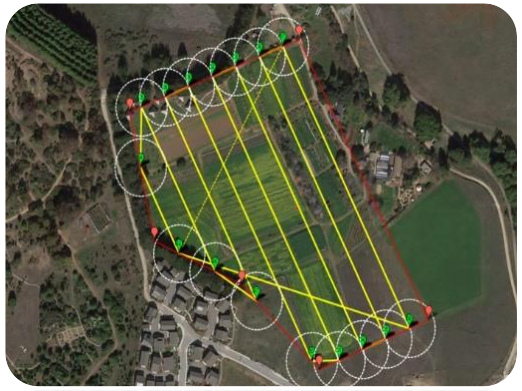
Test & Refine in Simulation
シミュレーション

Test & Refine on Real H/W
実装＆実機実験

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

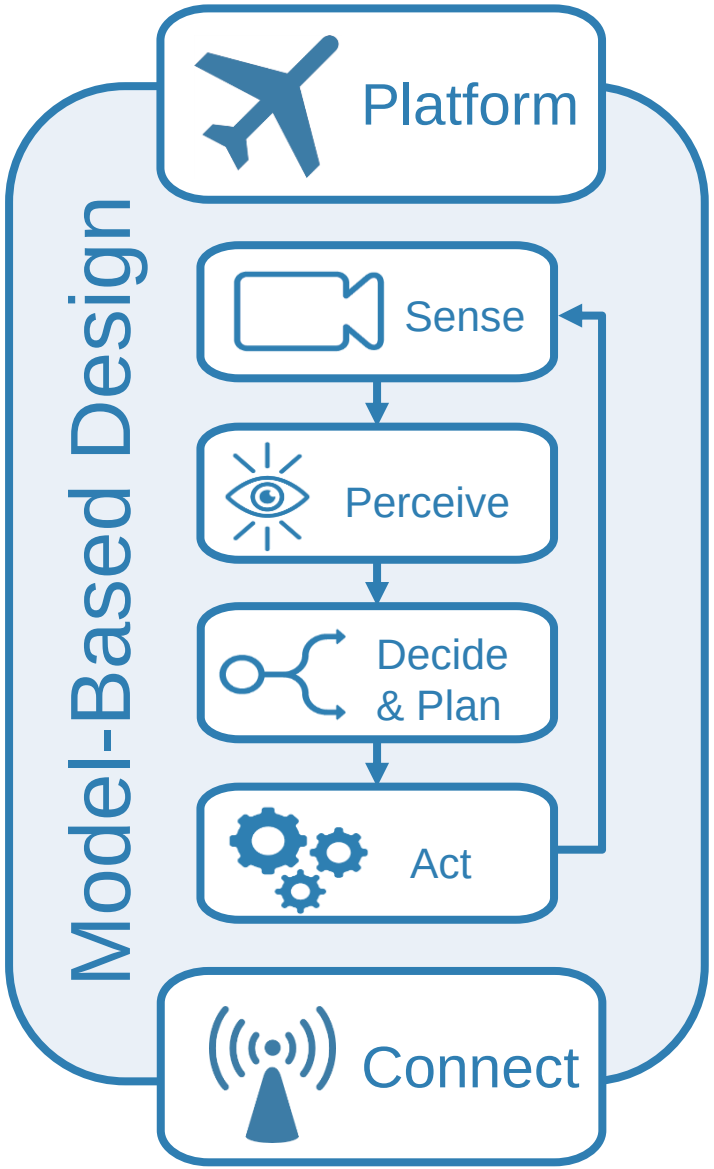


ステップ②：
ビジョン、レーダ、知覚アルゴリズムの設計



ステップ③：
アルゴリズムの検証と実装

自律システム開発に対する提案



本日のトピックス

- 自律システムの開発プロセス
- システム安全性の評価（DO-178C）

メッセージ

MATLAB/Simulinkの統一環境での自律システム開発・評価

RTCA/DO-178Cについて

“Software Considerations in Airborne Systems and Equipment Certification”

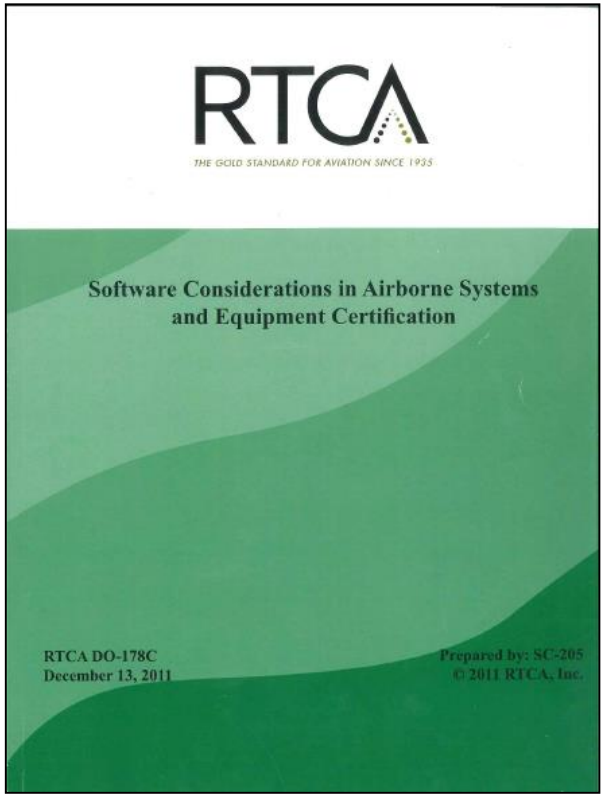
目的：

航空機システム・機器を対象としたソフトウェア開発のガイドライン

- 意図した機能を確保
- 意図しない機能を回避

考慮事項：

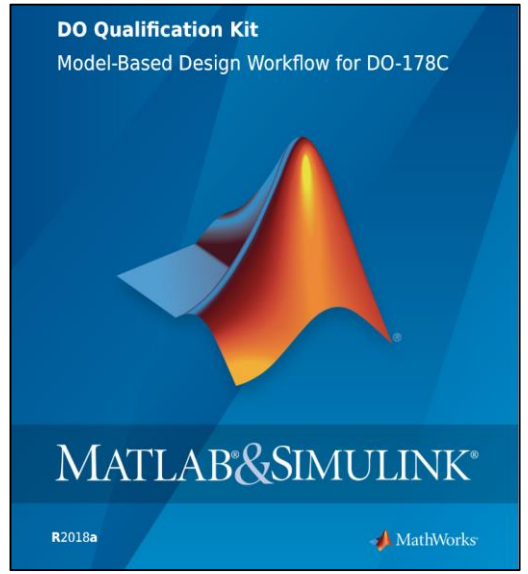
- ソフトウェアライフサイクル目標
- 開発および検証アクティビティの説明
- 目標達成に向けた必要な成果物の説明



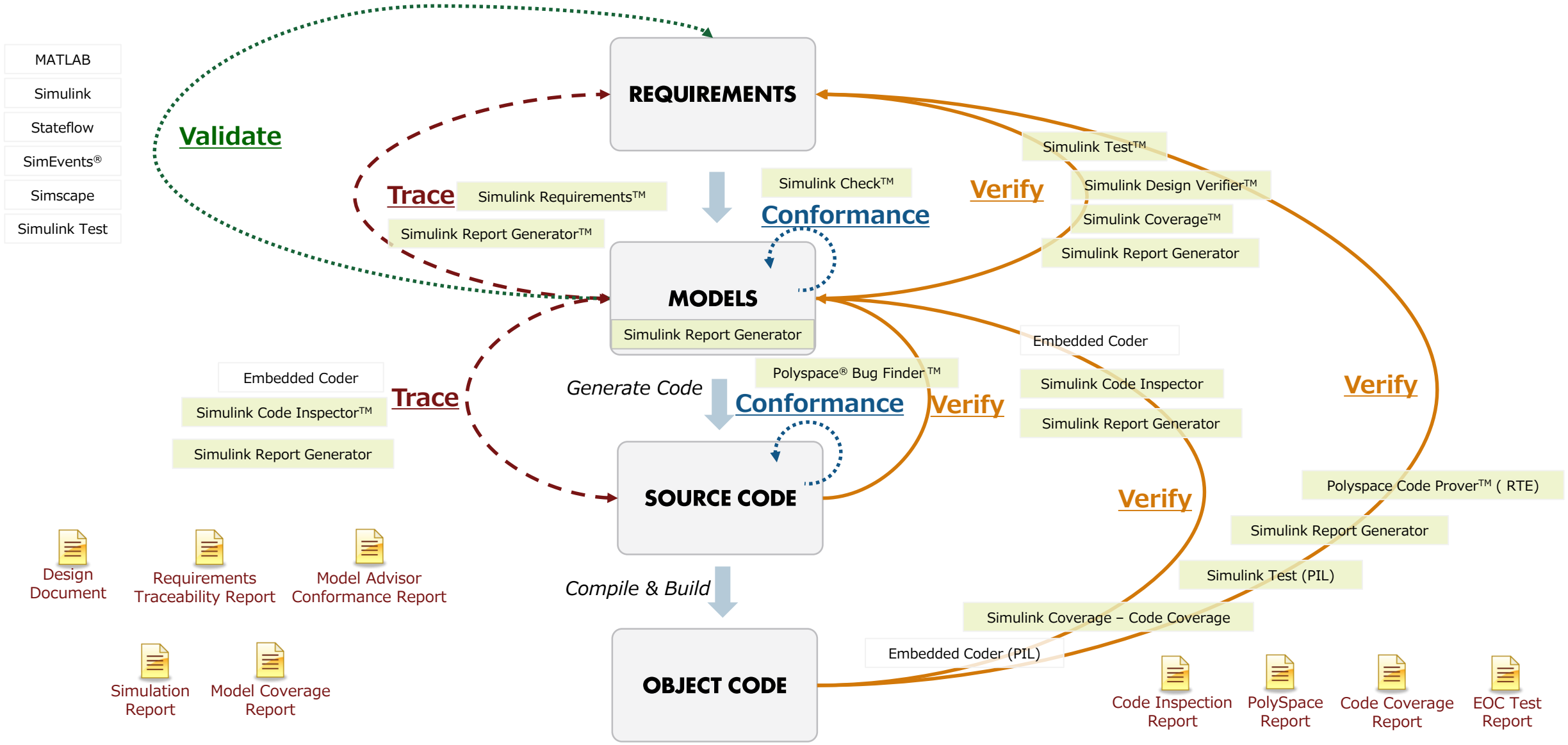
http://www.rtca.org/

DO Qualification Kit

- DO-178C/DO-331推奨ワークフロー
- ツール認定プランやツール動作要求を提供
- テストケースのモデルとコード、テスト手法、テスト結果の活用のドキュメント
- テストケースのマッピング・トレーサビリティの生成（リスト化）

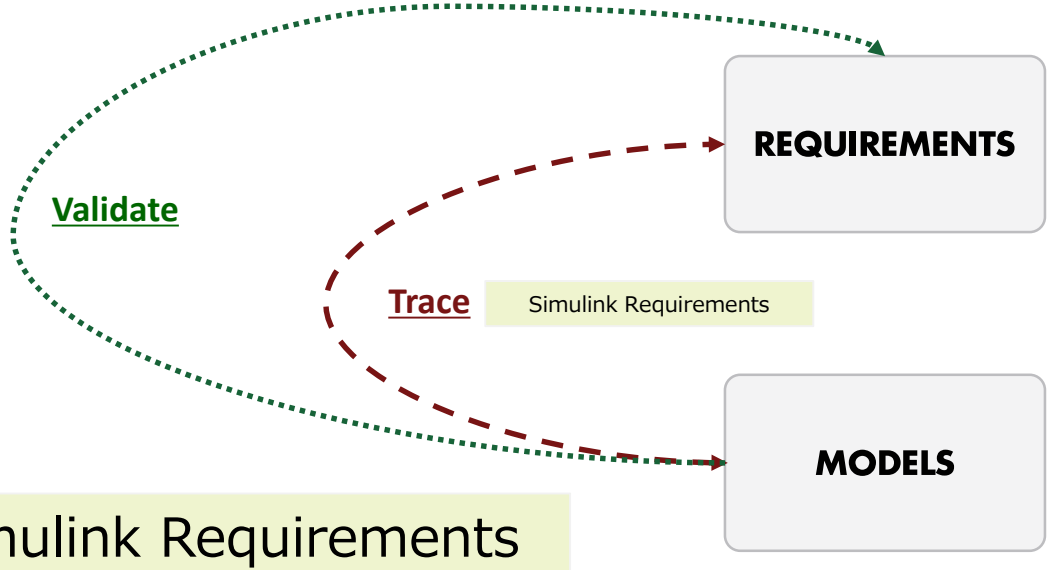


DO-178C 高信頼性ソフトウェア開発ワークフロー

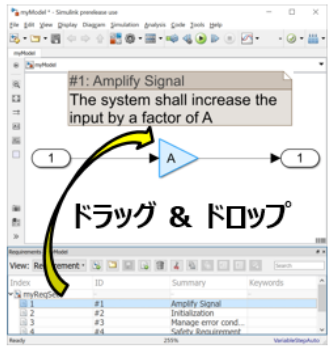
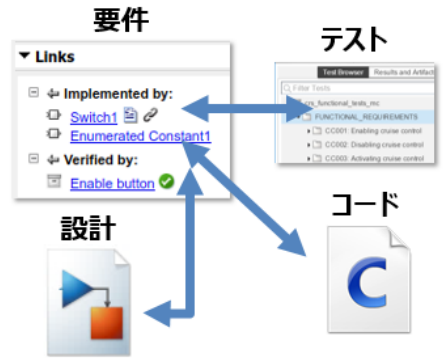


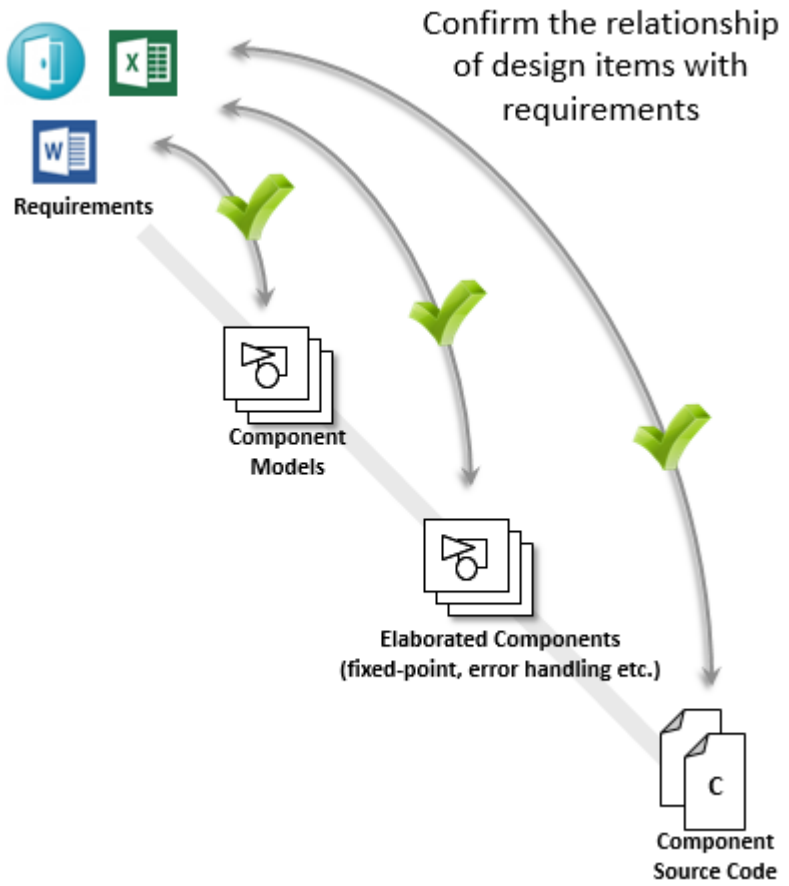
高信頼性ソフトウェア開発ワークフロー

MATLAB
Simulink
Stateflow
SimEvents
Simscape
Simulink Test



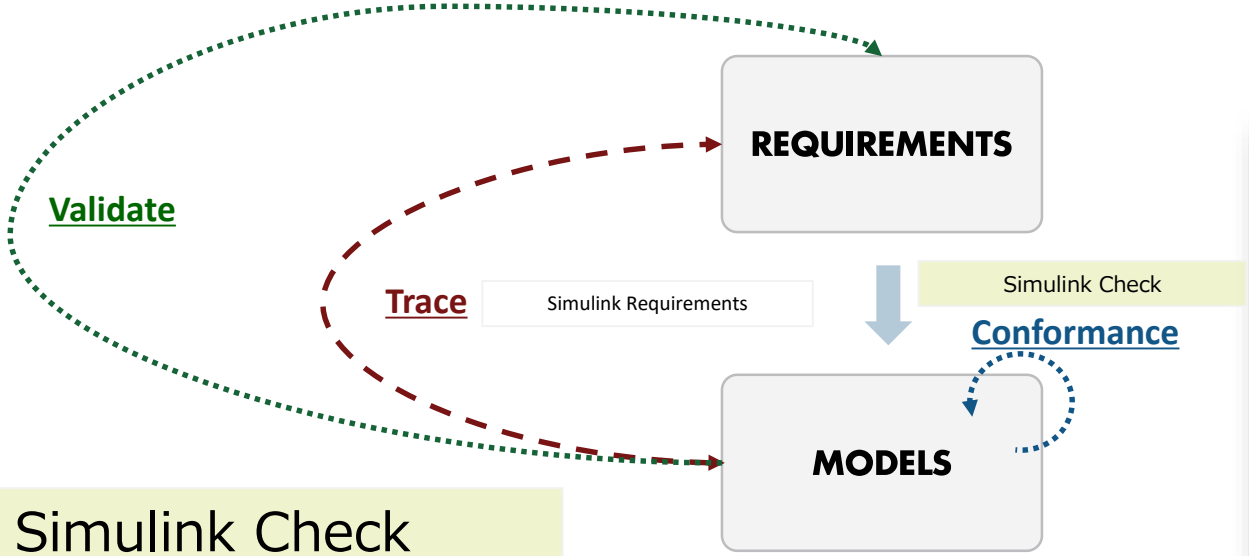
Simulink Requirements

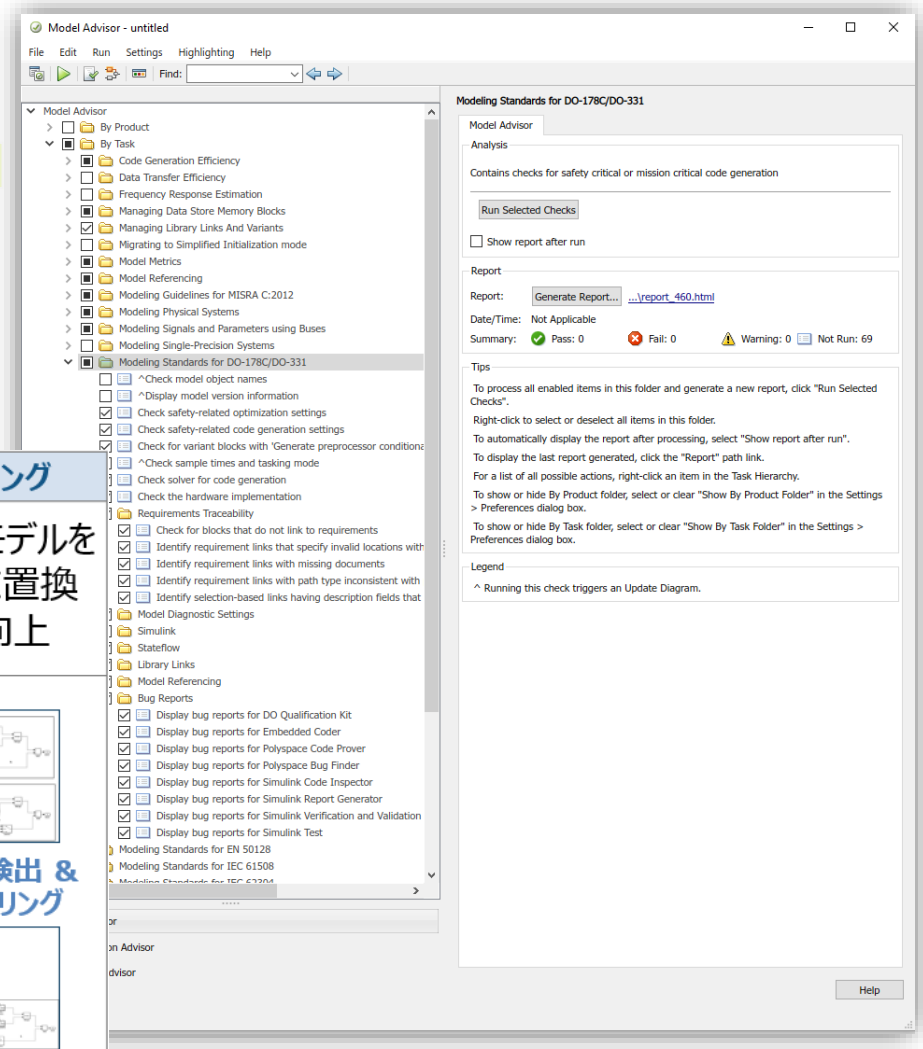
要件エディタ	要件パースペクティブ	要件トレーサビリティ
Simulink上で外部要件と内部要件の管理と分析	要件をドラッグ＆ドロップしてリンクを作成	要件と設計、コード、テスト間で双方向にトレーサビリティを確保
		



高信頼性ソフトウェア開発ワークフロー

MATLAB
Simulink
Stateflow
SimEvents
Simscape
Simulink Test



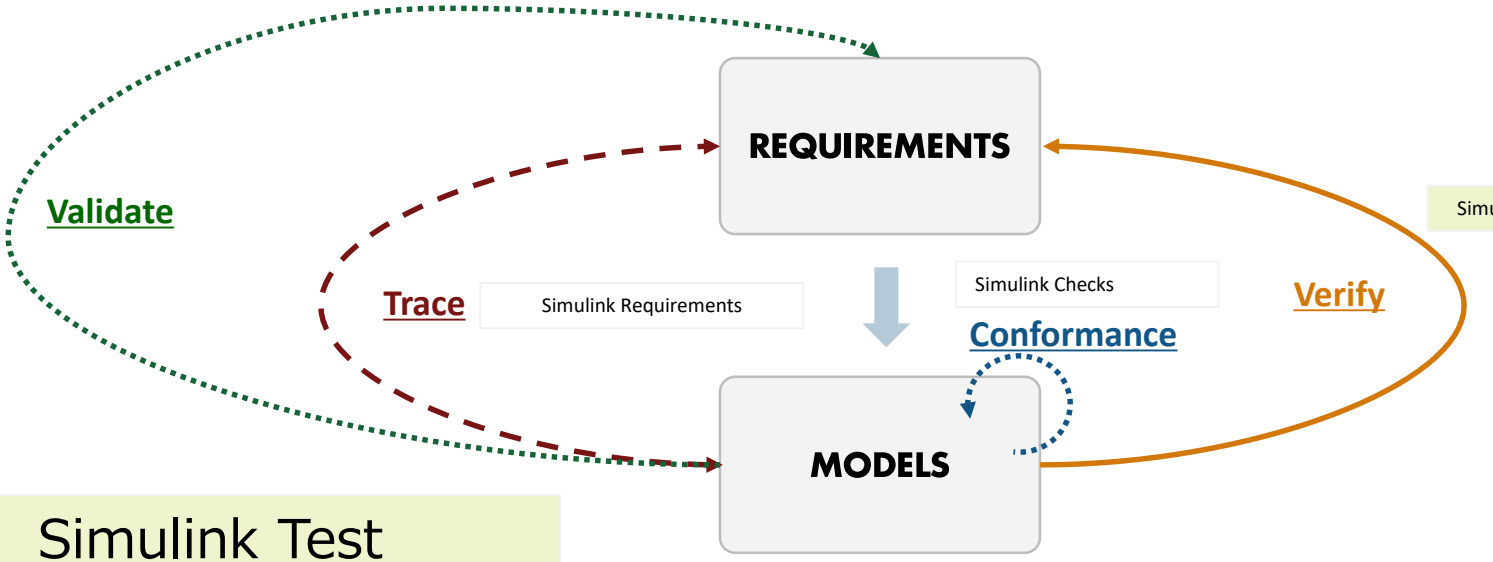


Simulink Check

モデルアドバイザー	モデル編集時チェック	モデルメトリクス	モデルリファクタリング
ガイドラインに準拠したモデル記述になっていることを確認	モデル編集時に潜在的な問題を早期発見	モデルのサイズ、複雑さ、可読性を定量化してモデルの品質を測定	設計が重複したモデルをライブラリブロックに置換して再利用性を向上
• GUIのモデルチェック • 自動修正 • レポート生成 • カスタムチェックの追加		メトリクスダッシュボード • サイズ • アーキテクチャ • モデルガイドラインの準拠	元のモデル クローン検出 & リファクタリング リファクタリングモデル

高信頼性ソフトウェア開発ワークフロー

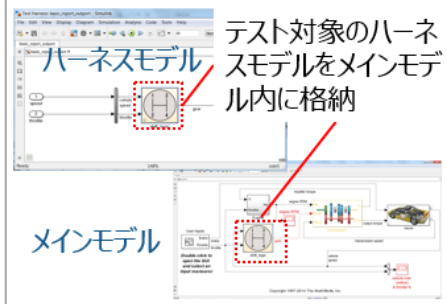
MATLAB
Simulink
Stateflow
SimEvents
Simscape
Simulink Test

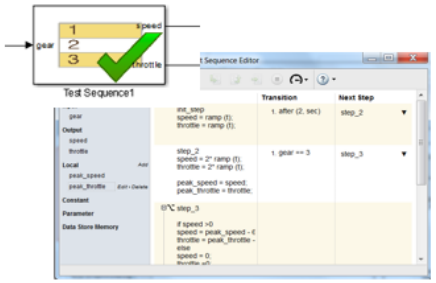


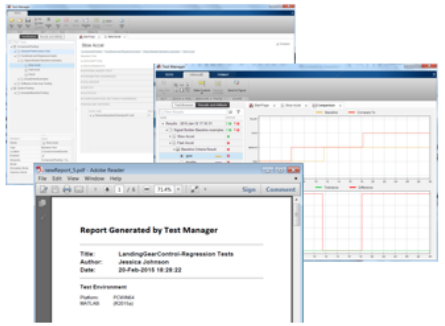
Simulink Test
Simulink Design Verifier
Simulink Coverage

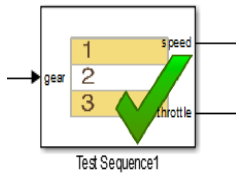
Simulink Test

テストハーネス	テストシーケンスブロック	テストマネージャー
モデル全体・サブシステム単体に対して複数テストハーネスを定義・実行	複雑な入力パターンを状態遷移表で簡潔に表現	MIL/SIL/PILテストの自動化、レポート作成、テスト管理





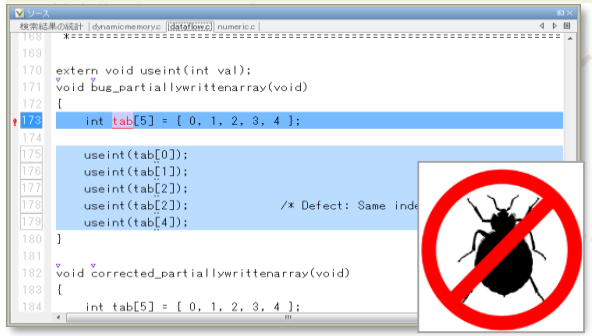




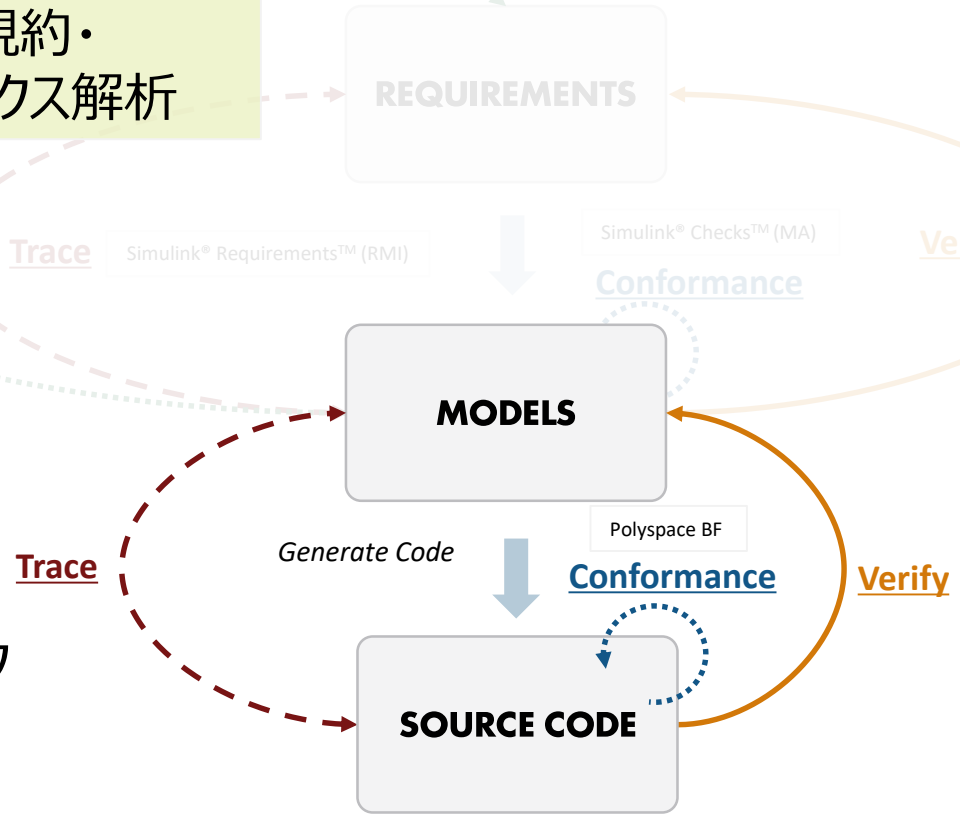
Supported coverage types
- Decision coverage
- Condition coverage
- MC/DC
- Lookup table coverage
- Signal range coverage

高信頼性ソフトウェア開発ワークフロー

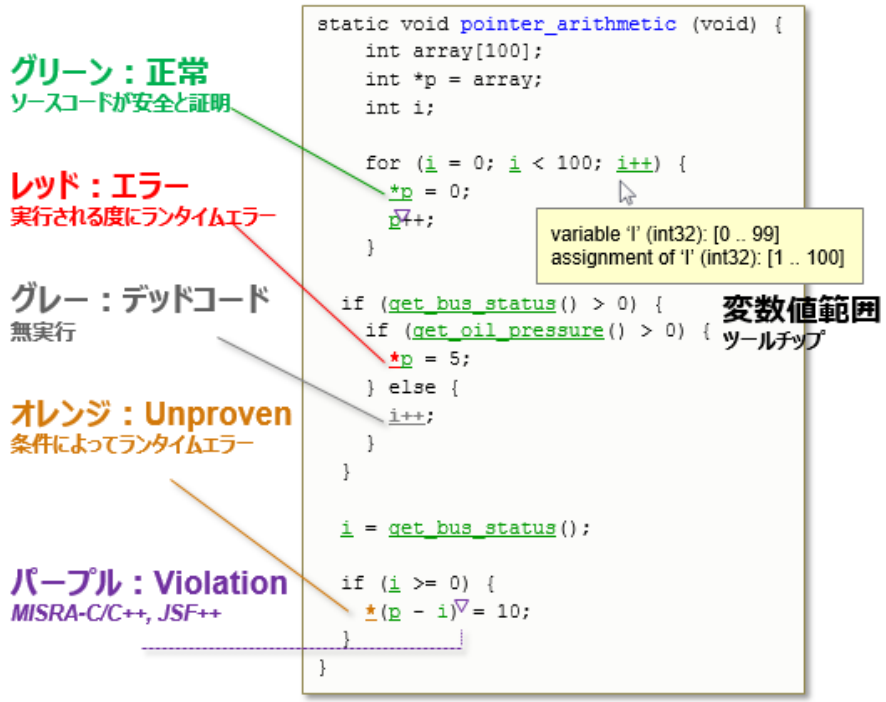
Polyspace Bug Finder
軽量なバグ検出・コード規約・
セキュリティ脆弱性・メトリクス解析



- バグ検出
  - 高速な解析
- コーディングルールチェック
  - MISRA-C準拠
- セキュリティ脆弱性
  - CERT C、CWE
- コードメトリクス解析
  - コード複雑度



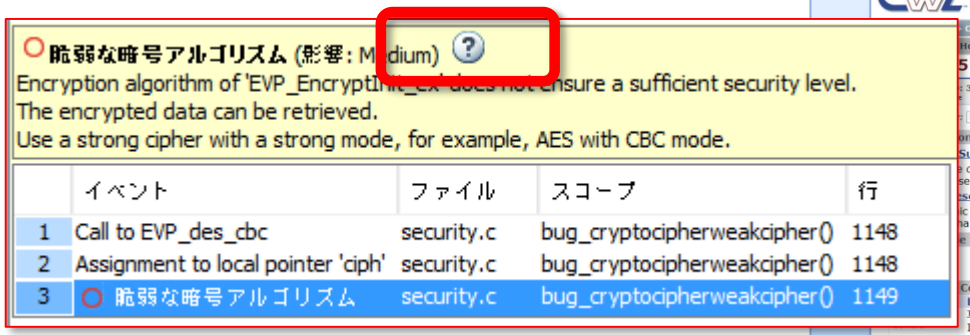
Polyspace Code Prover
全分岐パス解析＆エラーの
存在/不在を証明



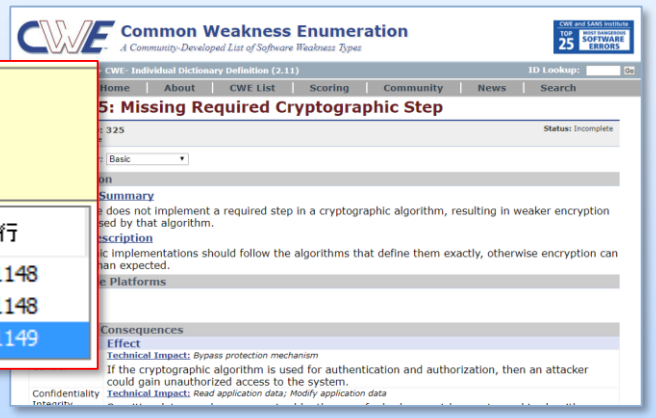
サイバーセキュリティ – 業界活動と標準

自動車業界に見られるコーディング標準＆実践

- **CERT C**：セキュアコーディングスタンダード
- **ISO/IEC TS 17961**：Cセキュアコーディングルール
- **CWE**：共通脆弱性タイプ
- **MISRA-C:2012** Amendment 1：改訂1



CWE Webサイト



https://cwe.mitre.org/data/definitions/325.html?browser=F1help

ドキュメンテーション ＆ 推奨アクション

DO-178C向けツール・機能の概要

*認証可能ツール

- **MATLAB, Simulink, Stateflow, Simscape:**
  - モデル設計、要求検証
- **Simulink Requirements:**
  - 要求とモデル間のトレーサビリティ
- **Simulink Checks:**
  - モデルアドバイザーによるルール準拠チェック
- **Simulink Coverage:**
  - モデル・コードカバレッジ測定
- **Simulink Test:**
  - MIL、SIL、HILによる要求検証
- **Simulink Design Verifier:**
  - 設計エラー検出（ゼロ除算、オーバフロー等）
  - テストケース自動生成
- **DO Qualification Kit**
  - ツール認証サポート
- **Embedded Coder:**
  - C/C++ソースコード自動生成
  - ソースコードとモデルのトレーサビリティ
  - PILモードによる検証
- **Simulink Code Inspector:**
  - ソースコードとモデルの等価性検証
- **Polyspace Bug Finder:**
  - ソースコードMISRA準拠の確認
  - セキュリティガイドライン準拠の確認（CERT C、CWE）
- **Polyspace Code Prover:**
  - 形式手法によるランタイムエラー検証
- **Simulink Report Generator:**
  - 成果物の生成

DO-178C関連ユーザ事例

DO-178B Level A certified

Bell 525:Bell Helicopter Develops World's First Commercial Fly-by-Wire Helicopter

- Hardware Integration time cut by 90% from 10 weeks to 1 week.
- EC generates the right code, and SLCI guarantee that it is traceable to the model.

"We have the same high level of confidence in the quality of the generated code as in the code created using our traditional manual processes"

Read User Story

AW609: Augusta Westland Develops the First Civilian Tiltrotor, Using MBD

- Full collaboration with suppliers via Simulink models
- Flight control system code generated automatically from models
- 40% improvement in design and development time
- Flawless first flight, which went exactly like the simulation

DO-178B Level A under certification

Read Technical Article

M-346: Alenia Aermacchi Develops Autopilot Software with MBD

- Requirements review for certification up to 30% shorter
- Low-level certification activities automated
- Time-to-flight reduced by 20%

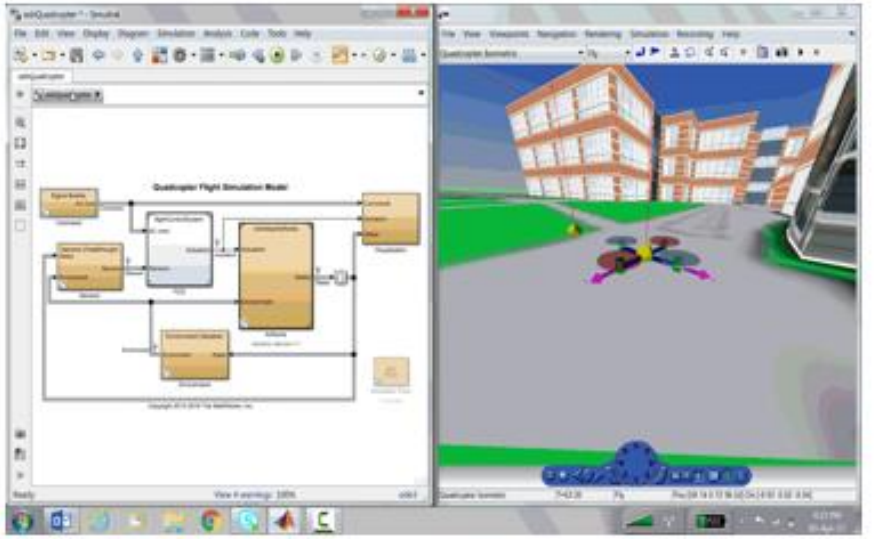
"Result is higher quality, DO-178B certified SW, and faster iterations."

DO-178B Level A certified

Read User Story

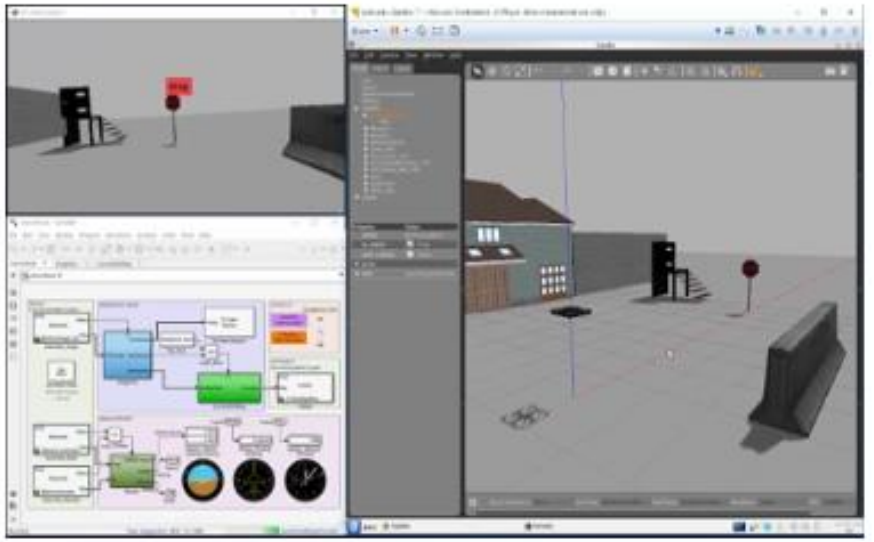
まとめ

**自律システムの設計・検証には
システムシミュレーションが重要！**



MATLAB/Simulinkでの自律システム開発・検証：

- 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
- 豊富な検証機能によるシステム安全性の確保



**MATLAB/Simulink：
自律システム開発・検証の統一環境！**

Thank You For Your Attention

ご清聴ありがとうございました