



# MATLAB EXPO 2022 JAPAN

## 無人航空機の自律飛行ソフトウェア開発における Model-Based Design適用事例

株式会社SUBARU 航空宇宙カンパニー  
研究部 空力制御設計課 梅沢 翔

# Agenda

1. SUBARUの事業紹介
2. 無人航空機の開発概況とMBD適用事例
3. 標準化に向けた活動
4. 適用範囲の拡大に向けて

# Agenda

1. SUBARUの事業紹介
2. 無人航空機開発の概況とMBD適用事例
3. 標準化に向けた活動
4. 適用範囲の拡大に向けて

# 自己紹介

梅沢 翔 (うめざわ しょう)

2007年 株式会社SUBARU 航空宇宙カンパニー入社  
(旧 富士重工業株式会社)

<主な担当業務>

固定翼および回転翼タイプの無人航空機に関する

- ✓ 運動シミュレーションの構築
- ✓ 自律飛行アルゴリズムの開発(経路生成・追従・飛行制御など)
- ✓ 上記に関するソフトウェア開発、各種試験



出典:陸上自衛隊HPより引用



# SUBARUの事業内容

飛行機研究所 (1917~)  
(のちの中島飛行機)



日本最大規模の大手航空機メーカーであった、中島飛行機を前身とする

富士重工業 (1953~)



スバル自動車部門

Automotive Business



航空宇宙カンパニー

Aerospace Company



株式会社SUBARU  
(2017~)

# SUBARUの国内主要拠点



航空宇宙カンパニー  
土地面積：668,000㎡  
従業員数：2,210人  
(2022年3月現在)



《本工場》



《南工場》飛行場地区



《南第2工場》



《半田工場》



《半田西工場》



《木更津事業所》



# SUBARU 航空宇宙カンパニーの事業① ヘリコプター

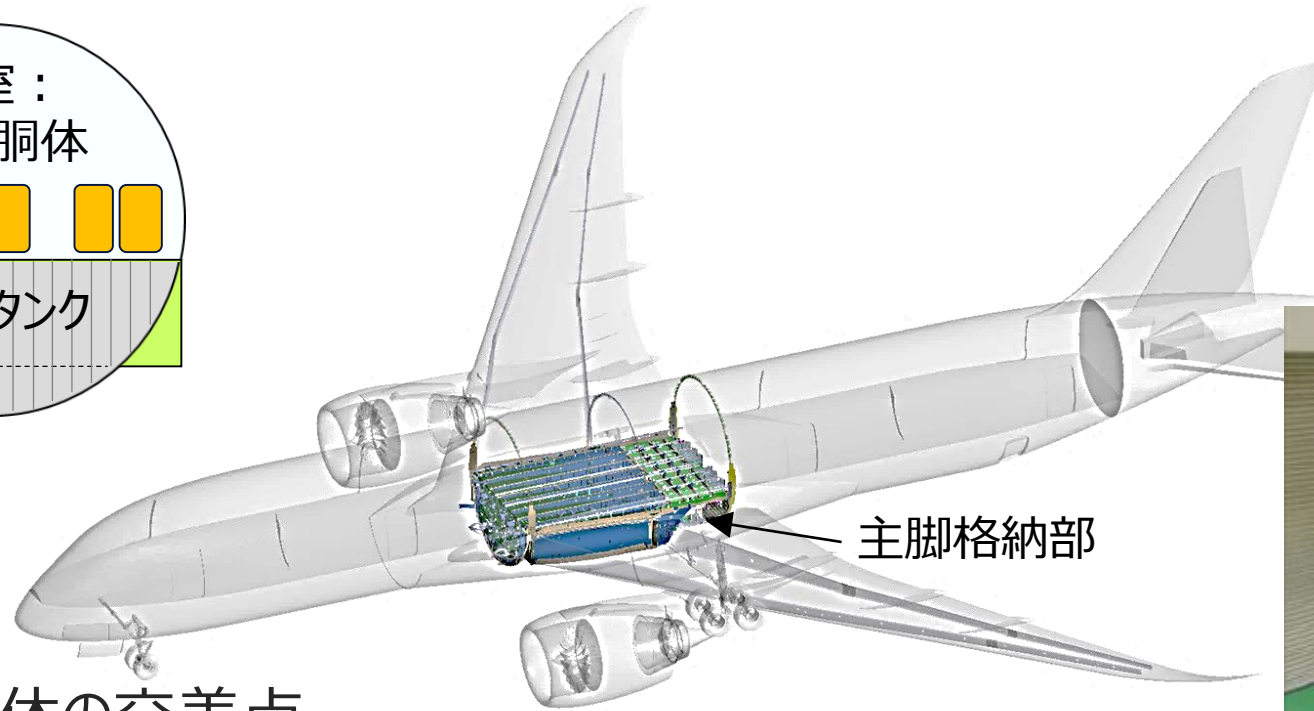
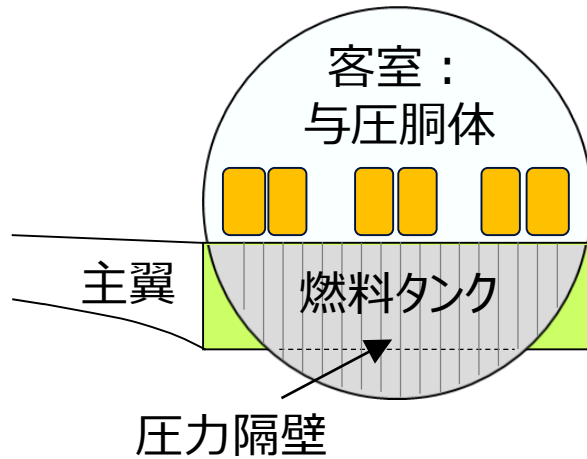
- 陸上自衛隊向けUH-2、民間機SUBARU BELL 412EPXを同時並行開発
- 米国ベル社との国際共同開発プロジェクト(2018年に試作機の初飛行を成功)
- UH-2量産と共に、民間機事業を展開
  - ⇒ 捜索救難・任務輸送・消防防災等の多用途にて社会のニーズに対応

SUBARU BELL 412EPX

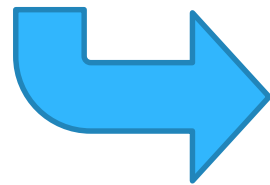


UH-2

# SUBARU 航空宇宙カンパニーの事業② 旅客機：中央翼



- 主翼と胴体の交差点  
⇒ 荷重最大、全システムが集積
- 多機能構造
- 安全と品質が第一  
⇒ 燃料タンク



- BOEINGへ大型機材を供給  
⇒ 777、787、777X
- 設計・開発・製造にフル参画



# SUBARU 航空宇宙カンパニーの事業③ 無人航空機

- ・ 50年以上に渡り無人航空機の研究開発を継続
- ・ 任務の多様化、機能の高度化に向けて、様々な技術を研究・開発中

1970年

1980年

1990年

2000年

2010年

2020年

標的型のターゲットドローンから  
自律・任務型の無人航空機へ

標的



出典: 日本産業用無人航空機協会HPより引用

農薬散布



出典: 陸上自衛隊HPより引用



偵察



自動離着陸

出典: 防衛装備庁HPより引用

警戒監視



# Agenda

1. SUBARUの事業紹介
- 2. 無人航空機の開発概況とMBD適用事例**
3. 標準化に向けた活動
4. 適用範囲の拡大に向けて

# 無人航空機の社会実装

- ・ 経済産業省において、2015年頃より「空の産業革命に向けたロードマップ」が策定
- ・ 無人航空機の社会実装に向けて、  
2022年には**有人地帯(まずは離島・山間部)での“目視外飛行”**の実現が掲げられている

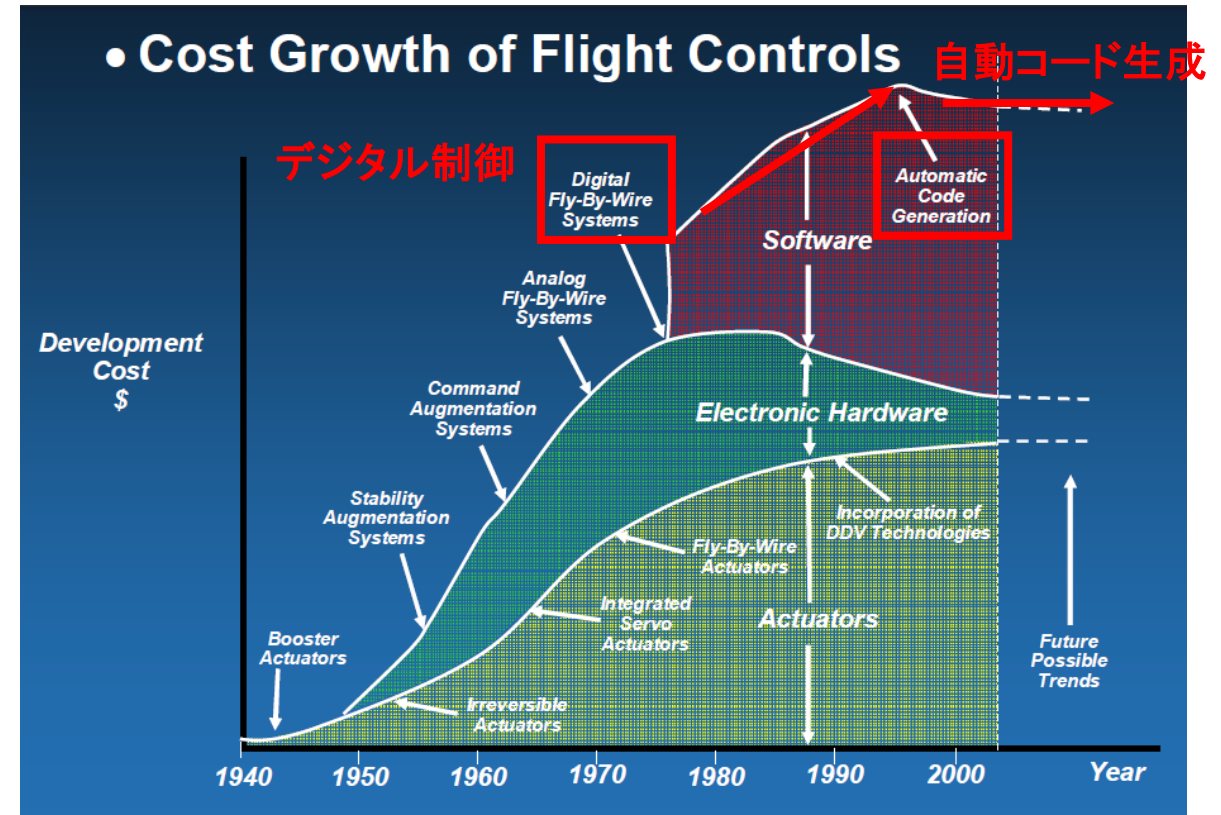
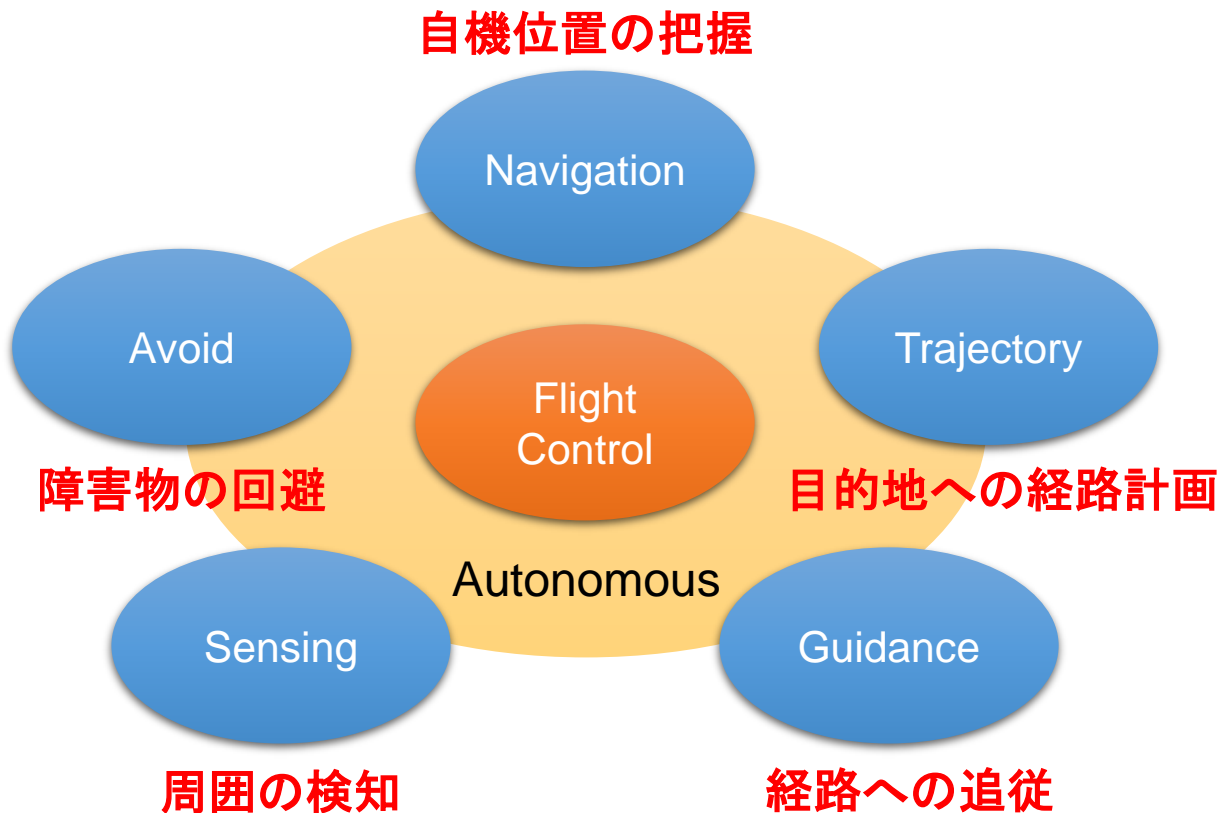


2022年～

出典: 無人航空機のレベル4の実現のための新たな制度の方向性について  
[https://www.kantei.go.jp/jp/singi/kogatamujinki/kanminkyougi\\_dai15/siryou1.pdf](https://www.kantei.go.jp/jp/singi/kogatamujinki/kanminkyougi_dai15/siryou1.pdf)

# 無人航空機の自律化

- ・ 目視外飛行(=地上パイロット不在)を実現するには、無人航空機の自律化が必須となる
- ・ 他機や障害物に対する自律的な衝突回避等、安全性を担保するためのソフトウェア機能が拡大、より一層の高品質化、開発の効率化が求められる



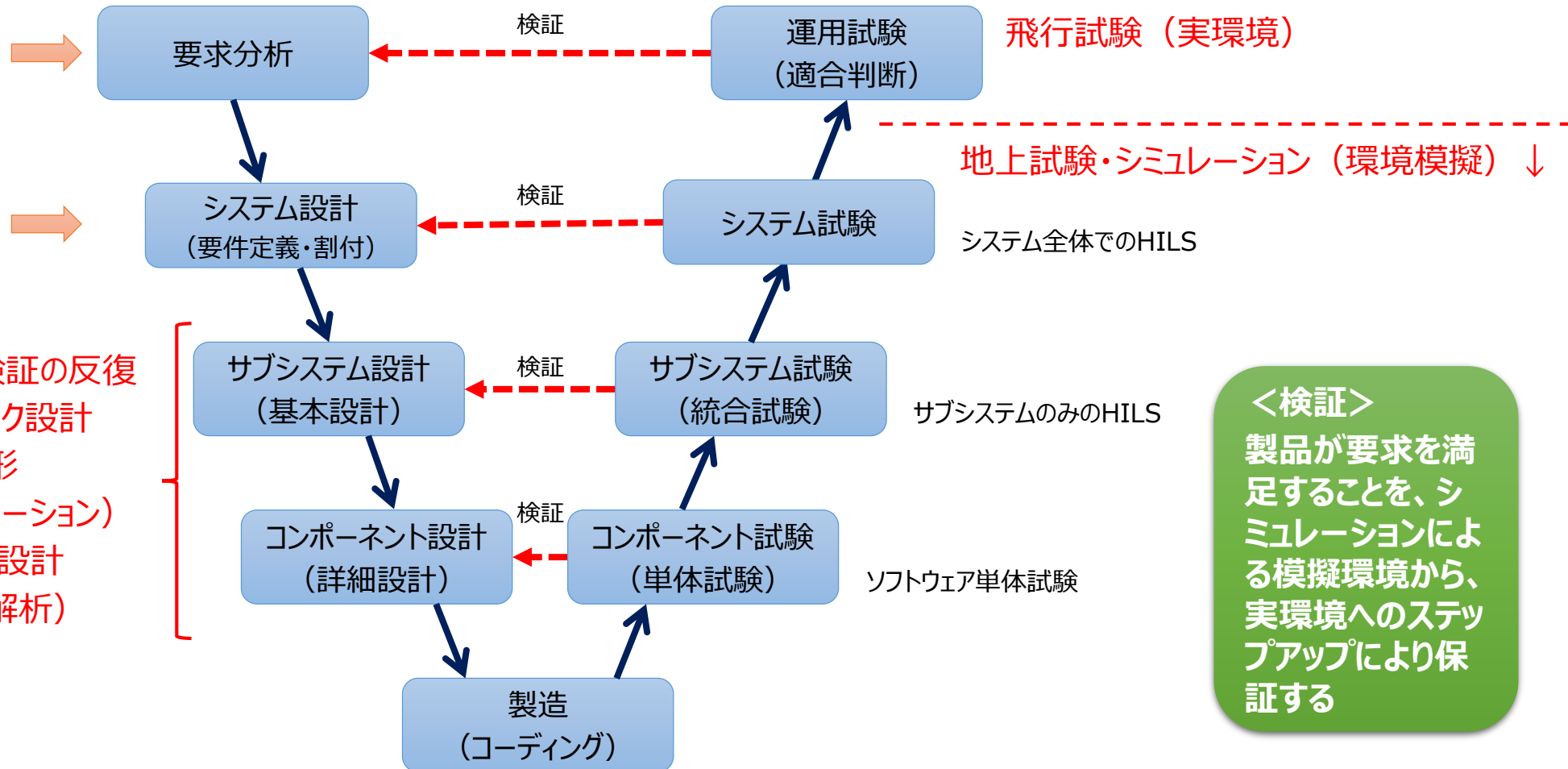
出典: Rapid Deployment of Aerospace Flight Controls  
 (Edward L. Burnett, Lockheed Martin Aeronautics Company, The MathWorks Aerospace and Defense Conference, 14 June, 2006)  
[https://it.mathworks.com/content/dam/mathworks/mathworks-dot-com/solutions/aerospace-defense/files/2006/madc\\_2006\\_03\\_burnett\\_lm.pdf](https://it.mathworks.com/content/dam/mathworks/mathworks-dot-com/solutions/aerospace-defense/files/2006/madc_2006_03_burnett_lm.pdf)

# 無人航空機の開発プロセス

- ・システム開発の一般的な流れであるV字モデルに基づき、システムの設計と検証を行う
- ・飛行試験（実環境）前に、早期段階からシミュレーション（模擬環境）を活用し、システム品質を高める

・顧客要求  
・アーキテクチャフレームワーク  
・制御設計スペック

スケールモデルやCAE解析による  
モデル基礎、詳細データ取得



**<設計>**  
顧客要求・スペックを元に要件を定義し、シミュレーションで見通しを得ながら、設計を固める

**設計と検証の反復**

- ① ロジック設計  
(非線形シミュレーション)
- ② 制御設計  
(線形解析)

**<検証>**  
製品が要求を満足することを、シミュレーションによる模擬環境から、実環境へのステップアップにより保証する

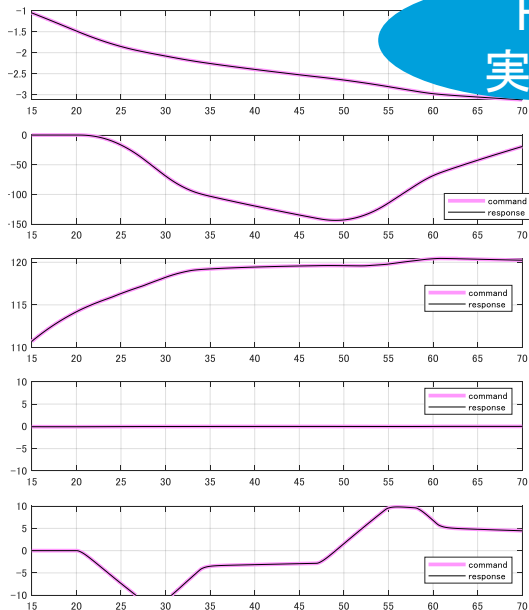
# 従来のソフトウェア開発における課題

- ・ ロジック設計(非線形シミュレーション)と制御設計(線形解析)を別々の開発環境で実施
- ・ 開発環境の統合は、プログラムの処理速度や実装に課題があり、困難であった

ロジック設計  
(非線形シミュレーション)

処理速度  
早い

HW  
実装可

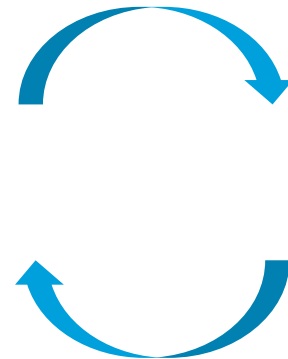


社内独自の  
シミュレーションプログラムを構築

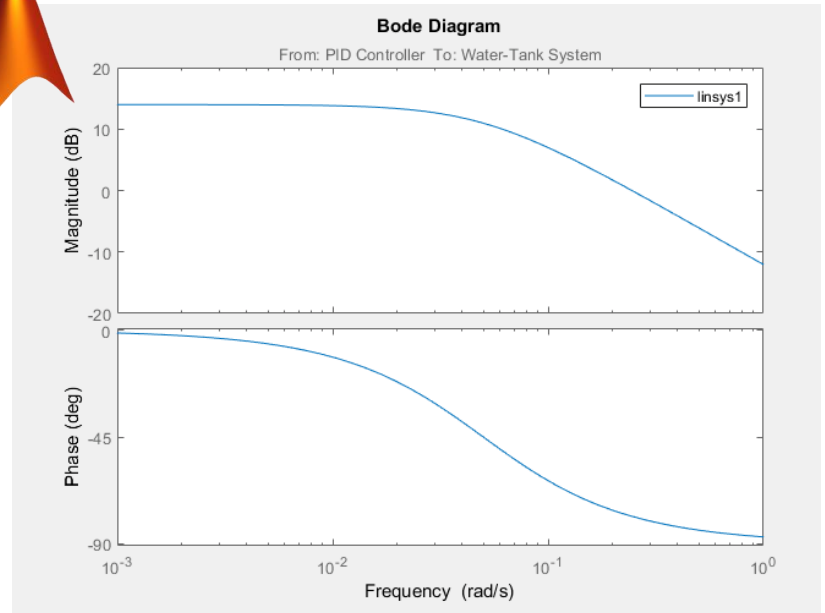
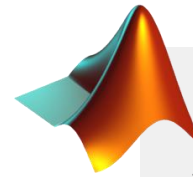
制御設計(線形解析)

解析容易

線形化



設計結果

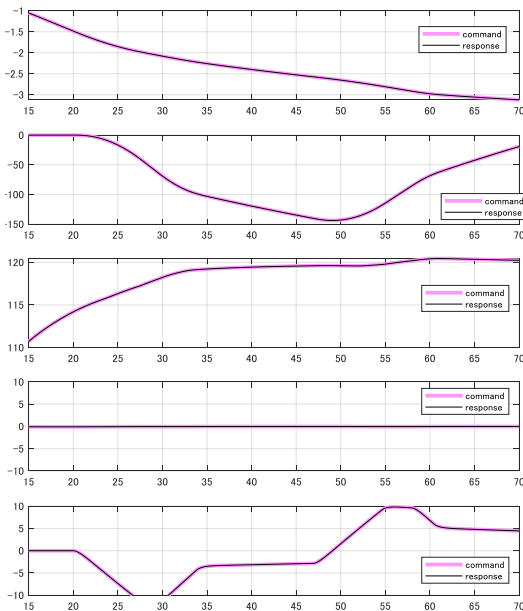


MATLAB<sup>®</sup>、Simulink<sup>®</sup>により実施

# 従来のソフトウェア開発における課題

- ・ ロジック設計に必要なシミュレーションプログラムの開発に、コーディングスキルを要していた
- ・ ロジックの実装、評価までに時間がかかるため、試作としては非効率

## ロジック設計 (非線形シミュレーション)



ロジック検討書



プログラム開発



Fortran、  
C/C++言語  
スキル



シミュレーション実行・ロジック評価

問題なければハードウェア実装

# MBD導入の経緯

- ～1990年 非線形シミュレーション、線形解析をそれぞれ他ツールで実施
  - 1990年～ 線形解析にMATLAB/Simulinkを導入
  - 2000年～ S-Functionをベースに、非線形シミュレーションにも活用を検討(但し、実用化に至らず)
  - 2010年～ ソフトウェア開発プロセス全般に、MBD導入を計画
- オートコード適用、非線形シミュレーションをMATLAB/Simulinkに移行**

1990年～

2000年～

.....

2010年～

非線形シミュレーション



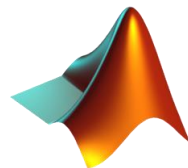
社内独自のシミュレーションプログラム



線形解析



他メーカーツールで実施



**MATLAB/Simulink導入**

・シミュレーション  
実行速度の壁  
(使いやすさ)

・オートコードの  
可読性・信頼性の壁  
(効率、品質)

**MATLAB/Simulinkに  
社内独自のシミュレーションプログラムを移行**

従来環境と  
の比較検証  
を実施

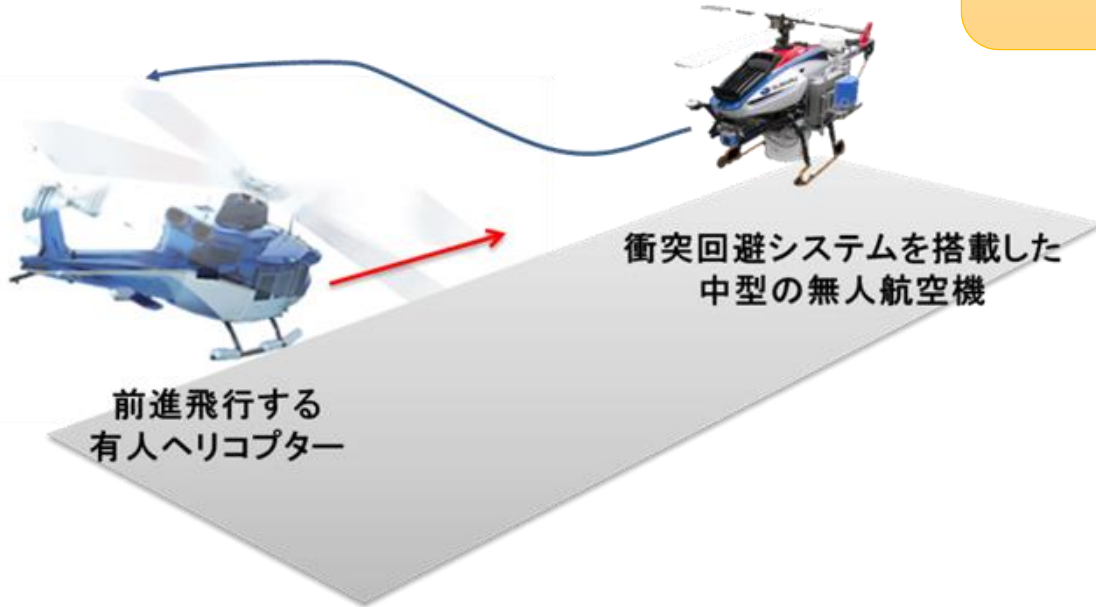


# MBD適用事例

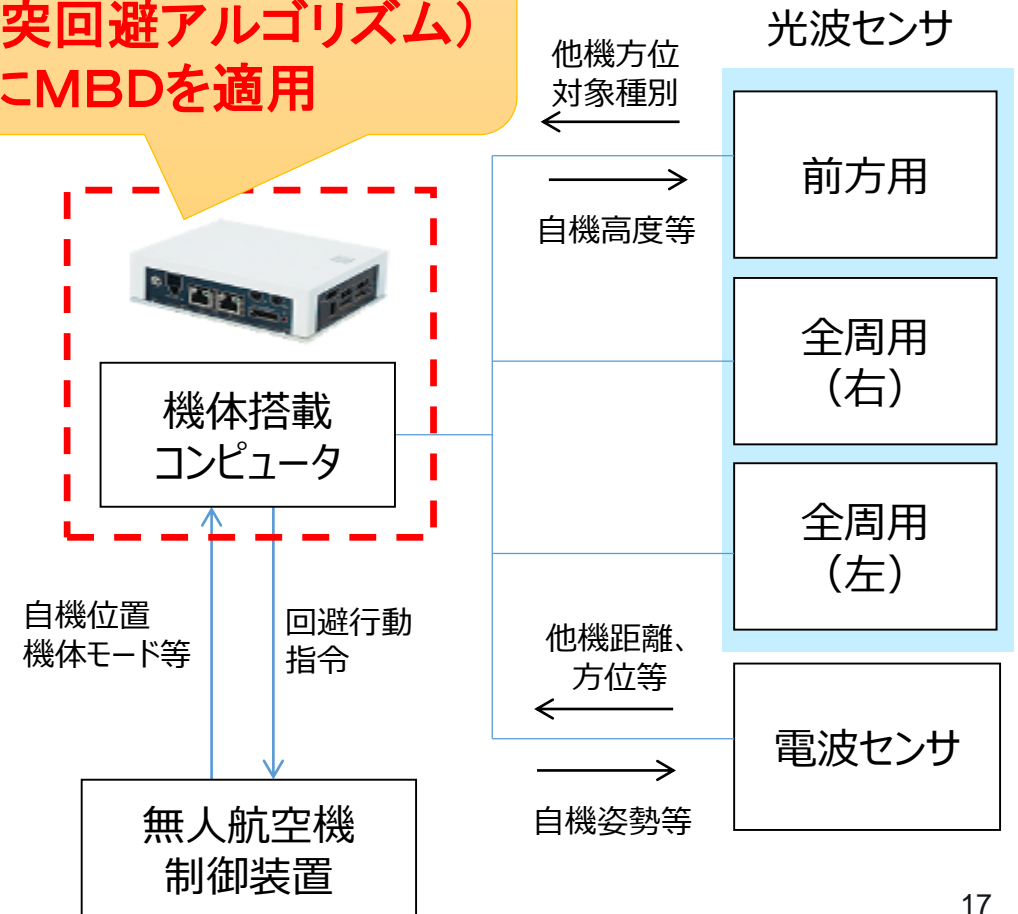
この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務 (JPNP17004) の結果得られたものです。

## NEDO委託業務：ロボット・ドローンが活躍する省エネルギー社会の実現プロジェクト 無人航空機の運航管理システム及び衝突回避技術の開発 (2017年～2019年度)

機体搭載コンピュータに実装するソフトウェア (衝突回避アルゴリズム) の開発にMBDを適用



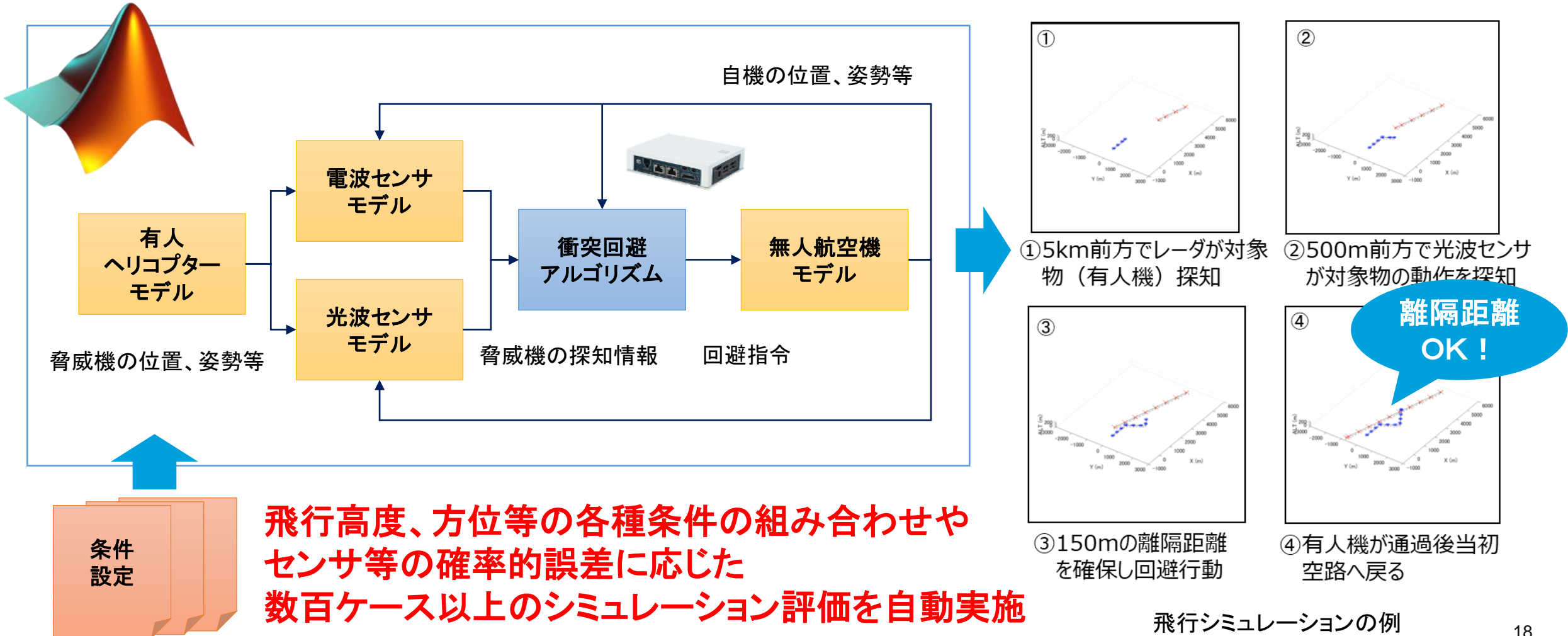
相対速度100km/hでの無人航空機の衝突回避試験を実施  
—搭載した各種センサで有人ヘリコプターを感知し、自律的に回避—



# MBD適用事例

この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務(JPNP17004)の結果得られたものです。

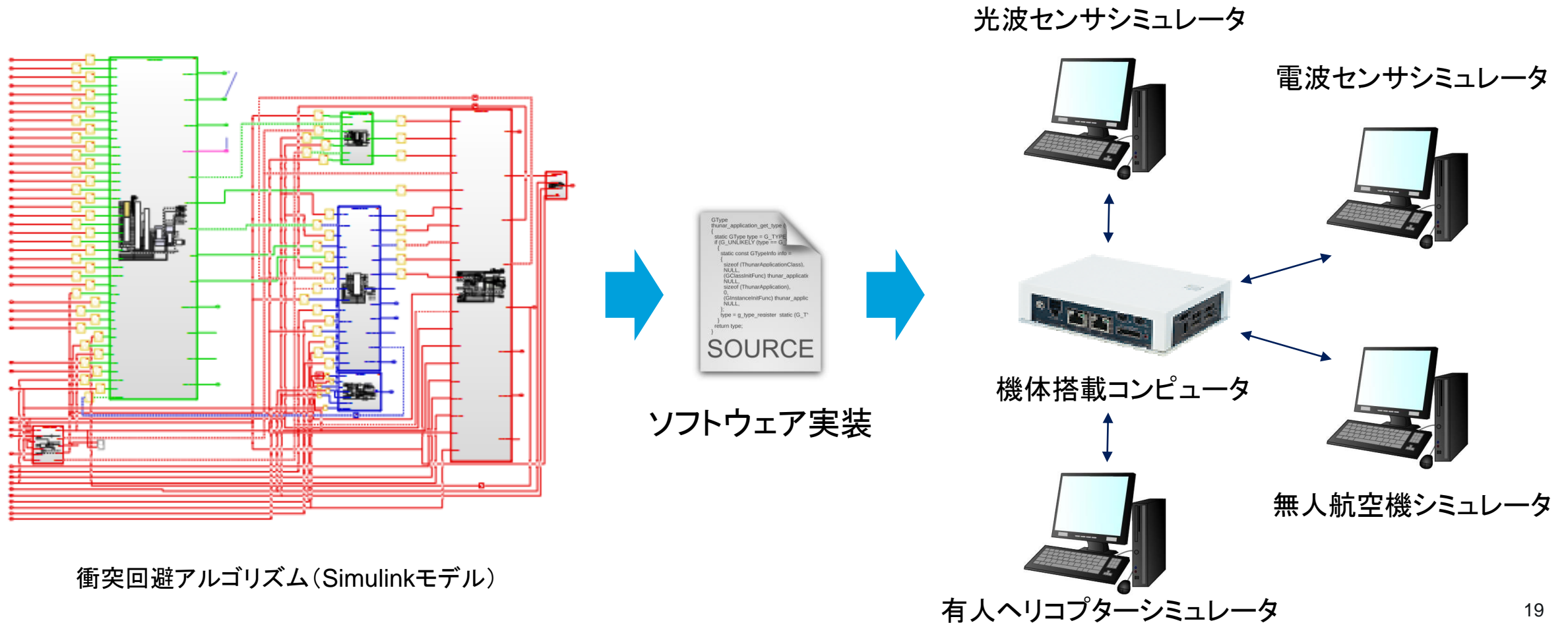
MATLAB/Simulinkで、各種センサや無人航空機の挙動をモデル化  
MILS (Model In the Loop Simulation) により、衝突回避アルゴリズムの妥当性を検証



# MBD適用事例

この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構(NEDO)の委託業務(JPNP17004)の結果得られたものです。

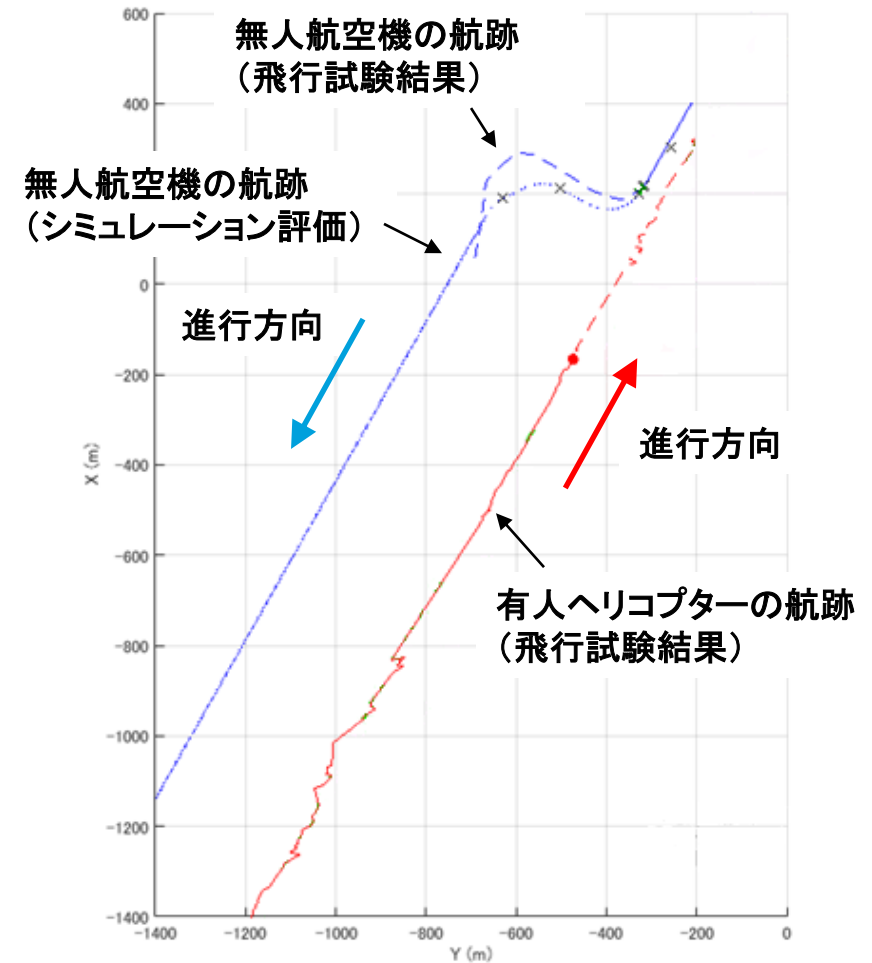
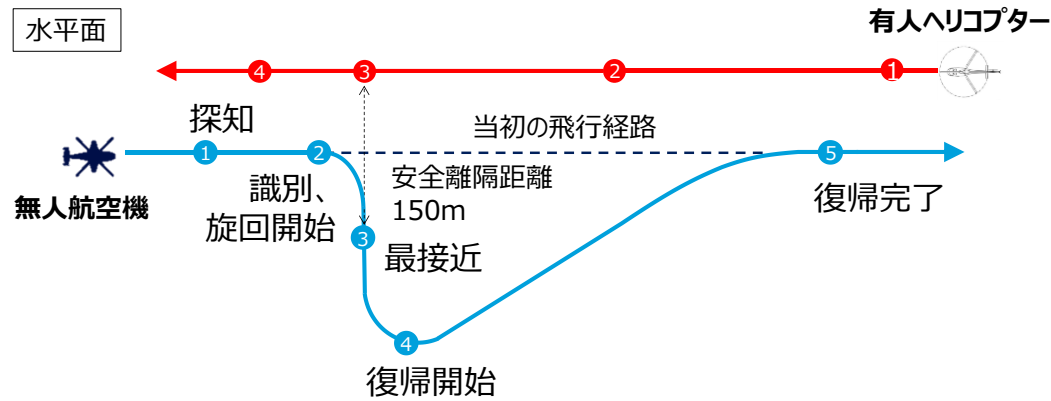
- ・ 妥当性が検証された衝突回避アルゴリズム (Simulinkモデル) から Embedded Coder<sup>®</sup>により、Cコードを生成し、機体搭載コンピュータにソフトウェアを実装
- ・ HILS (Hardware In the Loop Simulation) による飛行前の最終検証を実施



# MBD適用事例

この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務 (JPNP17004) の結果得られたものです。

飛行試験において、  
シミュレーション評価通りに、無人航空機が自律的な回避機動を行うことを確認した



# 製品活用に向けた課題

技術開発の効率化は図れたものの、製品活用(量産化)に向けては課題有り

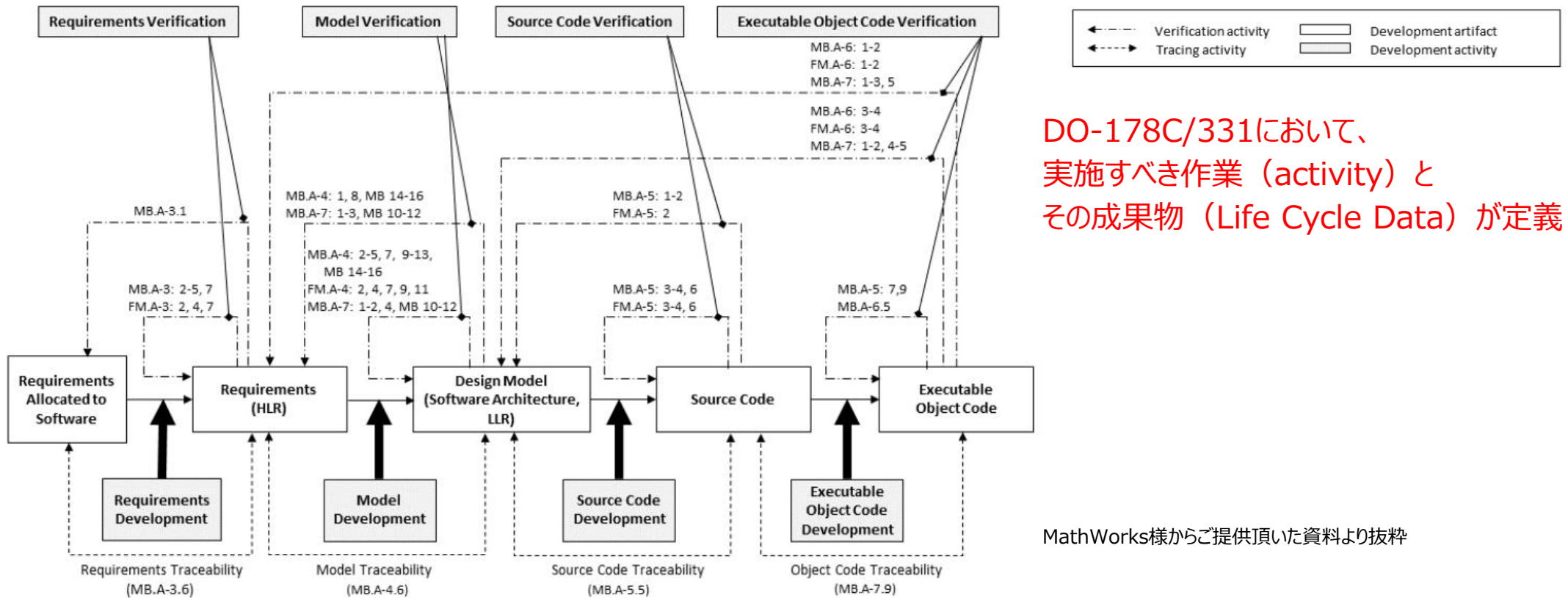
- ✓ **モデルの作り方**が作業者によって異なり、統一されていない  
⇒ 作成者以外には分かりづらく、**可読性**／**再利用性**に懸念あり
- ✓ コード生成を前提とした、**モデルの検証方法**が定まっていない  
⇒ オートコードとハンドコードを結合する際に、**不具合による手戻り**が発生している
- ✓ **コード生成**に必要なパラメータ項目が多く、設定方針が定まっていない  
⇒ コード生成されたソースコードの**品質バラツキ**に懸念あり



“モデル作成”、“モデル検証”、“コード生成”における社内標準の策定が必要！！

# 製品活用に向けた課題

社内標準の策定にあたっては、民間航空機用ソフトウェアの世界的規格である、DO-178C/331に準拠したプロセスとする（社内のみで通用する、ローカルな標準とならないように）



DO-178C/331において、実施すべき作業（activity）とその成果物（Life Cycle Data）が定義

MathWorks様からご提供頂いた資料より抜粋

- DO-178C ... 民間航空機に搭載されるソフトウェアの耐空性を保証するために必要な認証規格
- DO-331 ... 上記に対して、MBDを適用する際に必要な補足規格

# Agenda

1. SUBARUの事業紹介
2. 無人航空機の開発概況とMBD適用事例
- 3. 標準化に向けた活動**
4. 適用範囲の拡大に向けて





# MBDプロセス評価支援プログラム

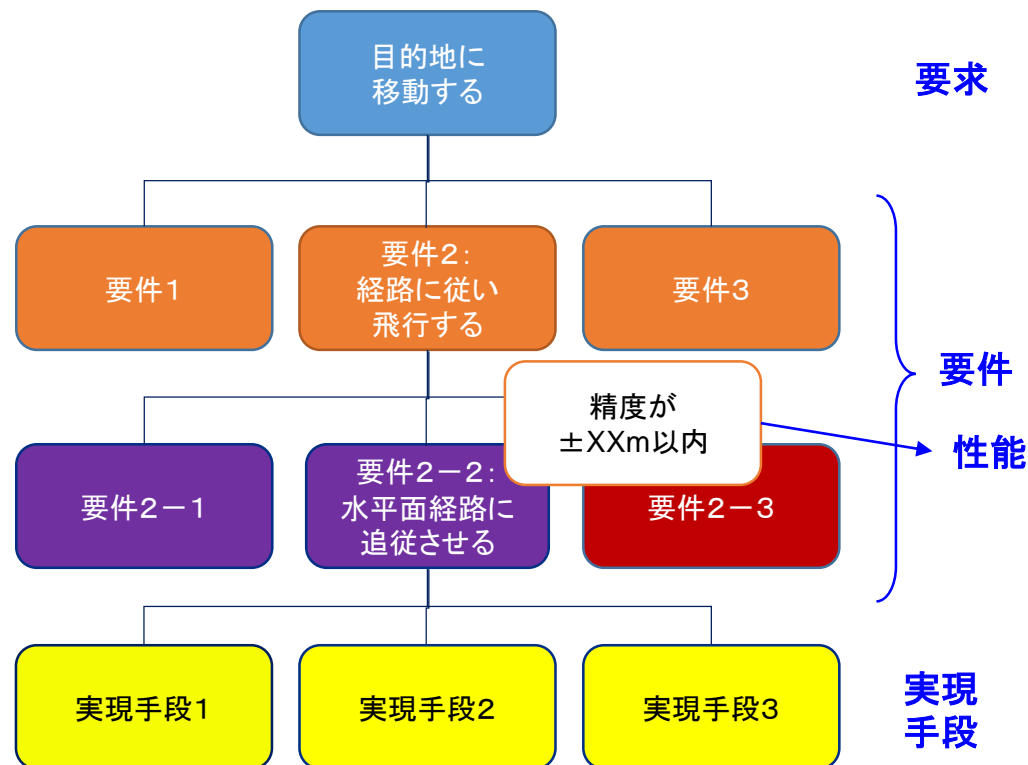
モデル作成から検証、コード生成とその検証まで、一通りの試行を実施  
ツールの使用方法を習熟すると共に、DO-178C/331の特徴的な点についてコメント頂いた

大項目	中項目	実施項目	メリット
モデル作成 (M)	仕様整理	M1. ソフトウェア要求仕様の整理	<u>シミュレーション活用</u> <ul style="list-style-type: none"> <li>機能要求の妥当性確認</li> <li>要求仕様の誤解や不整合の発見</li> <li>仕様ミス of 早期発見</li> <li>量子化誤差の確認</li> </ul>
		M2. ソフトウェア要求仕様の可視化と詳細化	
	モデル作成	M3. アーキテクチャーモデルの作成	
		M4. コンポーネントおよび詳細設計モデルの作成と機能確認	
		M5. コード生成モデルの作成と機能確認	
モデル検証 (V)	モデル検証	V1. 単体レベルモデルの検証(Verification)	<u>機能検証</u> <ul style="list-style-type: none"> <li>要求シナリオ下での機能検証作業</li> <li>ツールによる効率的なモデル検証作業</li> <li>ツールによるモデルカバレッジ測定および補完</li> <li>仮想環境による多様なシミュレーションおよび検証</li> <li>再現困難な状況下での検証</li> <li>テスト実行の自動化による試験業務の効率化</li> </ul>
		V2. 統合レベルモデルの検証(Validation)	
コード生成 (C)	コード作成	C1. 自動コード生成	<u>自動コード生成</u> <ul style="list-style-type: none"> <li>ソフト開発効率化</li> <li>ソフト品質の安定化</li> <li>プログラマ工数削減</li> <li>コードの動作保証・性能評価</li> <li>仕様とコードの一致性確保</li> </ul>
	コード検証	C2. モデルとコードの一致性確認など	

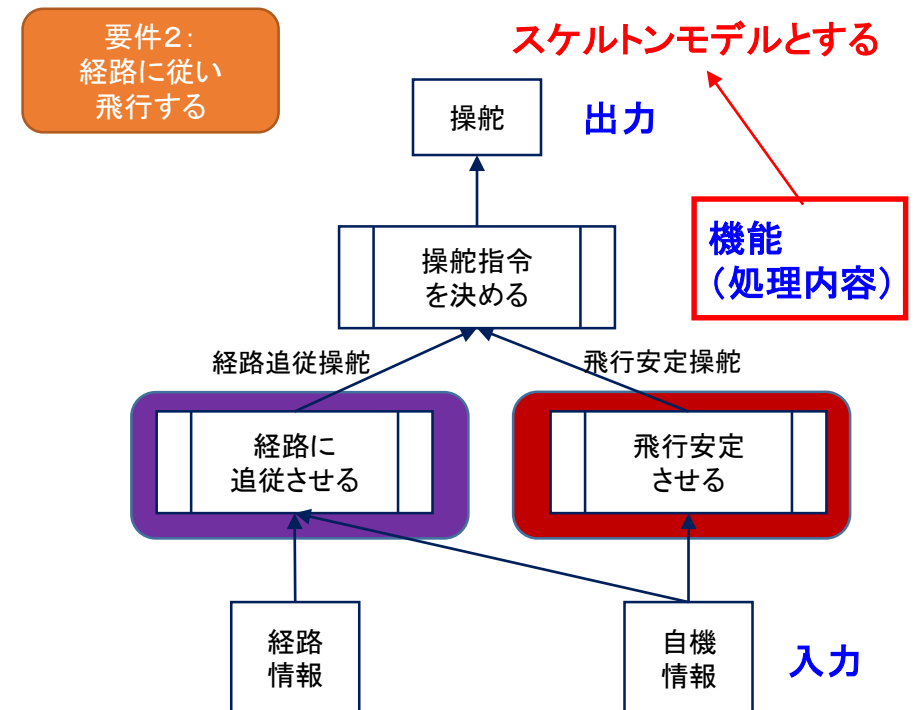
## <仕様整理>

- ・ソフトウェア要求の整理(要求図の検討)、モデルアーキテクチャの整理(制御構成図の検討)を実施
- ・要求をどこまでブレイクダウンすべきか、要件、実現手段の適切なレベル感について議論を行った

### 要求図の検討



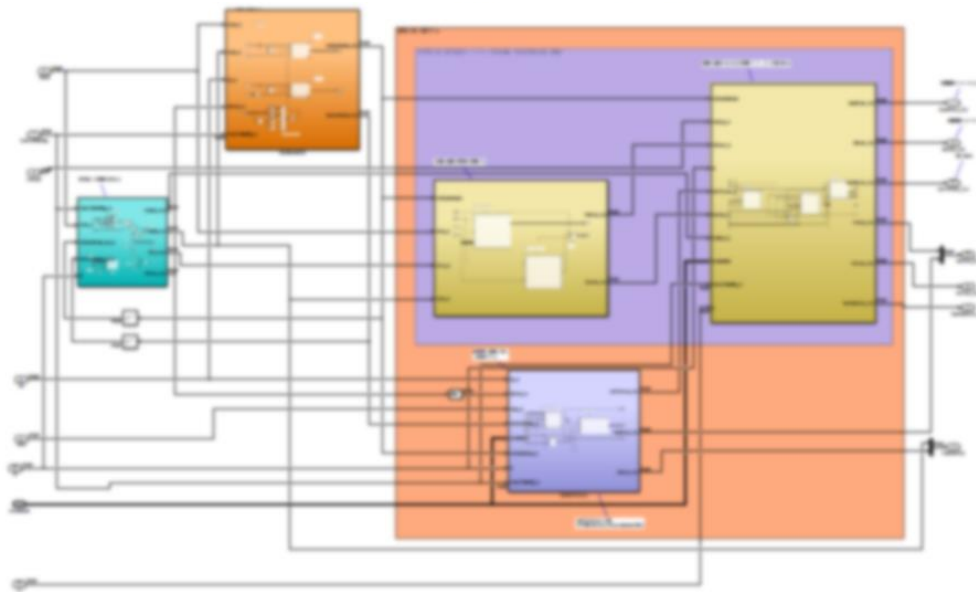
### 制御構成図の検討



## <モデル作成>

- ・ 制御構成図を元に、スケルトンモデル→インターフェイス定義→設計モデルの順にモデルを作成
- ・ モデル単体検証用の入出力期待値の作成を実施
- ・ SDDのレビュー方法(いつ、誰が、何をチェックするか)や、**モデル作成規約**について検討を行った

## SDD (System Design Description)



EXCEL変数管理ツールによる変数(インターフェイス含む)一元管理

No	Type	LabelName	Description(Japanese)	DataType	Min	Max	StorageClass	Cust
	オブジェクト	ラベル名	説明	データ型	最小値	最大値	ストレージクラス	カス
1	Simulink.S	Position	自機位置(Xa,Ya,ALTa)	single	-1000	1000	ImportedExtern	
2	Simulink.S	Velocity	自機対地速度(VNa,VEa,VDa)	single	-100	100	ImportedExtern	
3	Simulink.S	Attitude	自機姿勢角(Phia,Thetaa,Psia)	single	-180	180	ImportedExtern	

規約 ID	b_0009	<b>モデル作成規約の例</b>
規約名	インデックスの使用方法	
参考規約 ID	db_0112	
規約		
サブ ID	記述内容	カスタムパラメーター
a	ベクトル信号の{インデックス モード}は"0 ベース"に統一する。	
【例】		
【誤例】		
インデックスが「0」に統一されていない。		
根拠		
サブ ID	内容	
a	<ul style="list-style-type: none"> <li>・ 統一のインデックスを使用することで可読性が向上する。</li> <li>・ C 言語の配列インデックスに合わせることで可読性が向上する。</li> </ul>	

# モデル検証フェーズ

主に、Simulink Coverage™, Simulink Design Verifier™, Simulink Test™を使用

## <モデル検証>

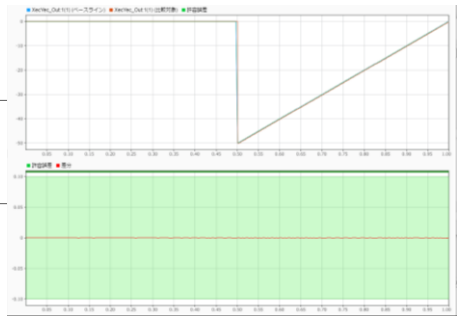
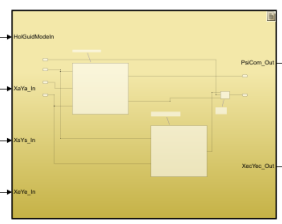
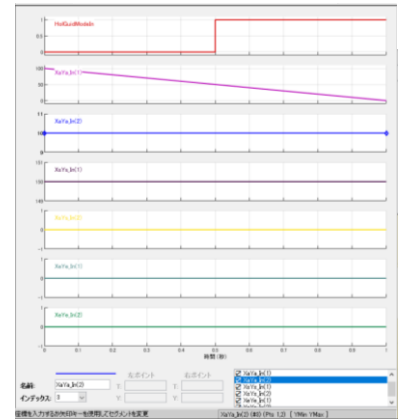
- モデルの設計進捗に応じて、チェックすべきモデルアドバイザーの項目を**モデル検証規約**に整理
- モデルに対する設計エラー検出、単体・統合検証、モデルカバレッジの取得を行うと共に、モデルの実装状況と検証状況に関するトレーサビリティの可視化を行った

モデルアドバイザーのチェック項目	詳細設計モデル	コード生成モデル
Simulink	<a href="#">Simulink のチェック</a>	
Embedded Coder®	<a href="#">Embedded Coder のチェック (Embedded Coder)</a>	
AUTOSAR Blockset	<a href="#">AUTOSAR Blockset のチェック (AUTOSAR Blockset)</a>	
Simulink Coder™	<a href="#">Simulink Coder のチェック (Simulink Coder)</a>	
HDL Coder™	<a href="#">HDL コード アドバイザー チェック (HDL Coder)</a>	
Simulink Code Inspector™	<a href="#">Simulink Code Inspector Checks (Simulink Code Inspector)</a>	
Simulink Check	<a href="#">DO-178C/DO-331 Checks (Simulink Check)</a>	
	<a href="#">IEC 61508, IEC 62304, ISO 26262, ISO 25119, and EN 50128/EN 50657 Checks (Simulink Check)</a>	
	<a href="#">DO-254 標準へのモデルの準拠性のチェック (Simulink Check)</a>	
	<a href="#">High Integrity System Modeling Checks (Simulink Check)</a>	
	<a href="#">Model Advisor Checks for MAB and JMAAB Compliance (Simulink Check)</a>	
	<a href="#">MISRA C:2012 Checks (Simulink Check)</a>	
	<a href="#">Secure Coding Checks for CERT C, CWE, and ISO/IEC TS 17961 Standards (Simulink Check)</a>	
	<a href="#">Model Metrics (Simulink Check)</a>	
	<a href="#">Clone Detection Checks (Simulink Check)</a>	
	<a href="#">Simulink Design Verifier のチェック (Simulink Design Verifier)</a>	
Simulink Requirements™	<a href="#">要件の整合性チェック (Simulink Requirements)</a>	
Simscape™	ドキュメンテーションはモデル アドバイザーでのみ使用できます。チェックのドキュメンテーションを確認するには、モデル アドバイザーでチェックのタイトルを右クリックして「これはなに?」を選択します。	
Simulink Control Design™	<a href="#">Simulink Control Design のチェック (Simulink Control Design)</a>	
IEC Certification Kit	<a href="#">High Integrity System Modeling Checks (Simulink Check)</a>	
DO Qualification Kit	<a href="#">High Integrity System Modeling Checks (Simulink Check)</a>	

詳細設計モデル  
コード生成モデル

ごとに  
チェック要否  
を定義

## 単体検証によるモデルカバレッジ取得の例



## 各要件に対するモデル実装、検証状況の可視化の例

Index	ID	Summary	Implemented	Verified
1	#16		<input type="checkbox"/>	<input type="checkbox"/>
2	#18		<input type="checkbox"/>	<input type="checkbox"/>
3	#2		<input type="checkbox"/>	<input type="checkbox"/>
4	#3		<input type="checkbox"/>	<input type="checkbox"/>
4.1	#4		<input type="checkbox"/>	<input type="checkbox"/>
4.1.1	#17		<input type="checkbox"/>	<input type="checkbox"/>
4.1.2	#18		<input type="checkbox"/>	<input type="checkbox"/>
4.2	#6		<input type="checkbox"/>	<input type="checkbox"/>
4.2.1	#19		<input type="checkbox"/>	<input type="checkbox"/>
4.2.2	#20		<input type="checkbox"/>	<input type="checkbox"/>
4.3	#7		<input type="checkbox"/>	<input type="checkbox"/>
4.4	#8		<input type="checkbox"/>	<input type="checkbox"/>
4.5	#9		<input type="checkbox"/>	<input type="checkbox"/>
5	#5		<input type="checkbox"/>	<input type="checkbox"/>

# コード生成フェーズ

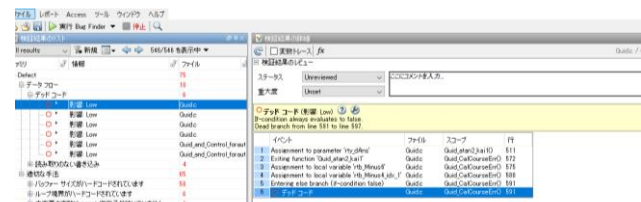
## <コード生成・検証>

- コード生成に必要な各パラメータ設定について良否の議論を行い、**コード生成規約**として整理
- モデルと自動生成されたコードの等価性確認を行うと共に、Back To Backテストやコードカバレッジの取得、静的解析を実施した

カバレッジの結果の集計

解析されたモデル	レポート	SIM モード	実数値...	判定	実行	FUNCTION	FUNCTION CALL
Guid/Subsystem/CalCourseErr		ModelRefSIL	114	79% <span style="color: red;">-</span>	95% <span style="color: blue;">-</span>	100% <span style="color: blue;">-</span>	93% <span style="color: blue;">-</span>
Guid/Subsystem/CalCourseErr		Normal	16	95% <span style="color: blue;">-</span>	100% <span style="color: blue;">-</span>	--	--

## ソースコードの静的解析の例



Web ブラウザ - Guid の SIL カバレッジレポート

Guid の SIL カバレッジレポート

U1.2 Iteration2 17-Dec-2021 15:37:16

### 概要

ファイルの内容/種類	判定	ステートメント	関数	関数呼び出し
1. Guid.c	114 79%	95%	100%	93%
2. ... Guid_atan2_kall	6 80%	86%	100%	100%
3. ... Guid_CalCourseErr	3 75%	98%	100%	91%

### 詳細

#### 1. ファイル Guid.c

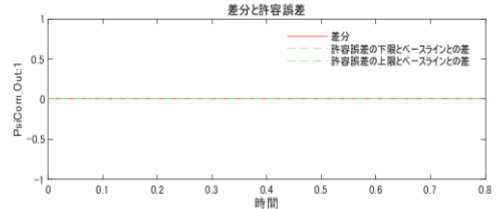
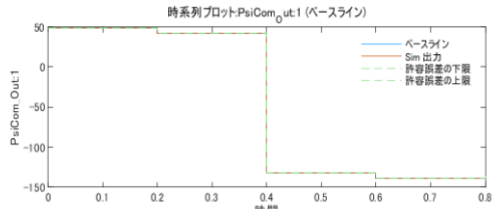
正当化または排除  
関数: Guid\_atan2\_kall (行 499), Guid\_CalCourseErr (行 537)

メトリクス

カバレッジ
導得的複雑度: 114
判定: 79% (11/14) 判定結果
ステートメント: 95% (53/56) カバーされたステートメント
関数: 100% (2/2) カバーされている関数
関数呼び出し: 93% (13/14) カバーされている関数呼び出し

#### 2. 関数 Guid\_atan2\_kall (行 499)

正当化または排除  
ファイル: Guid.c (code)  
モデル オブジェクト: atan2\_kall

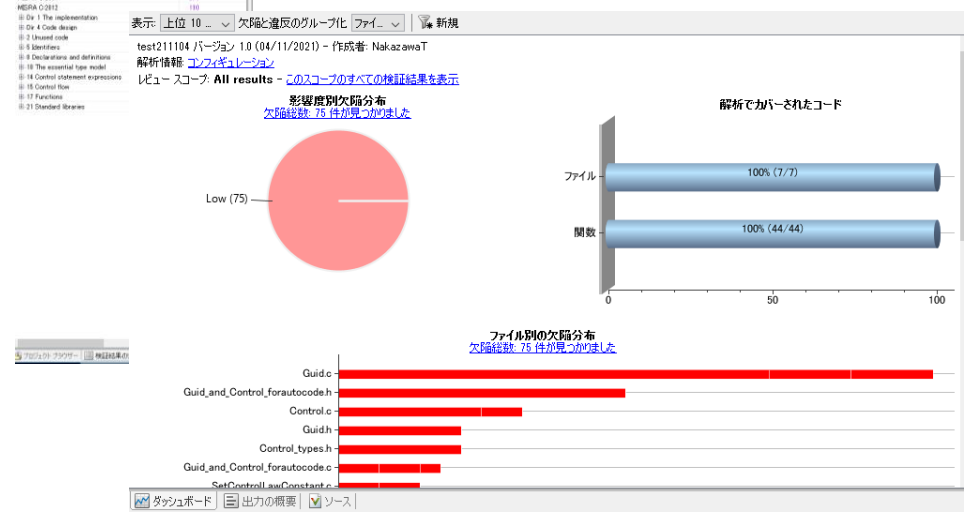


Iteration1

テスト結果情報  
結果タイプ: テスト反復の結果  
観: Guid\_Harness1  
開始時間: 2021/12/17 14:28:56  
終了時間: 2021/12/17 14:29:16  
結果: パス

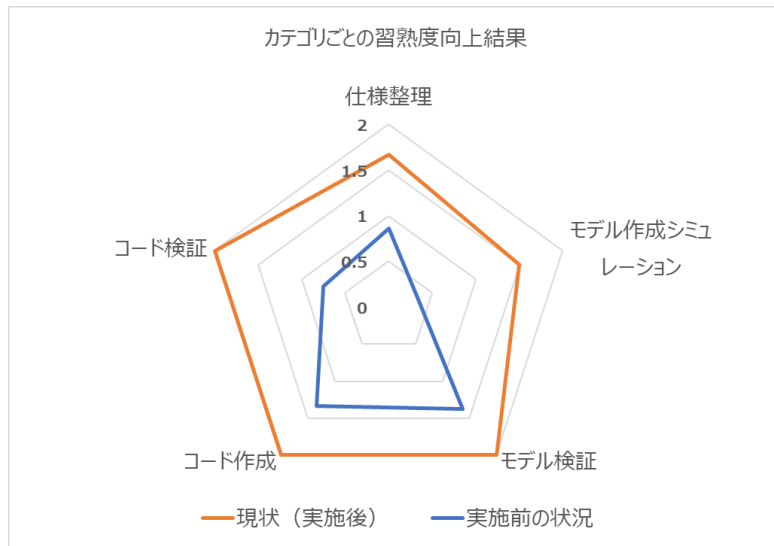
テストケース情報  
名前: Iteration1  
タイプ: 等価性テスト

## コードカバレッジ取得の例



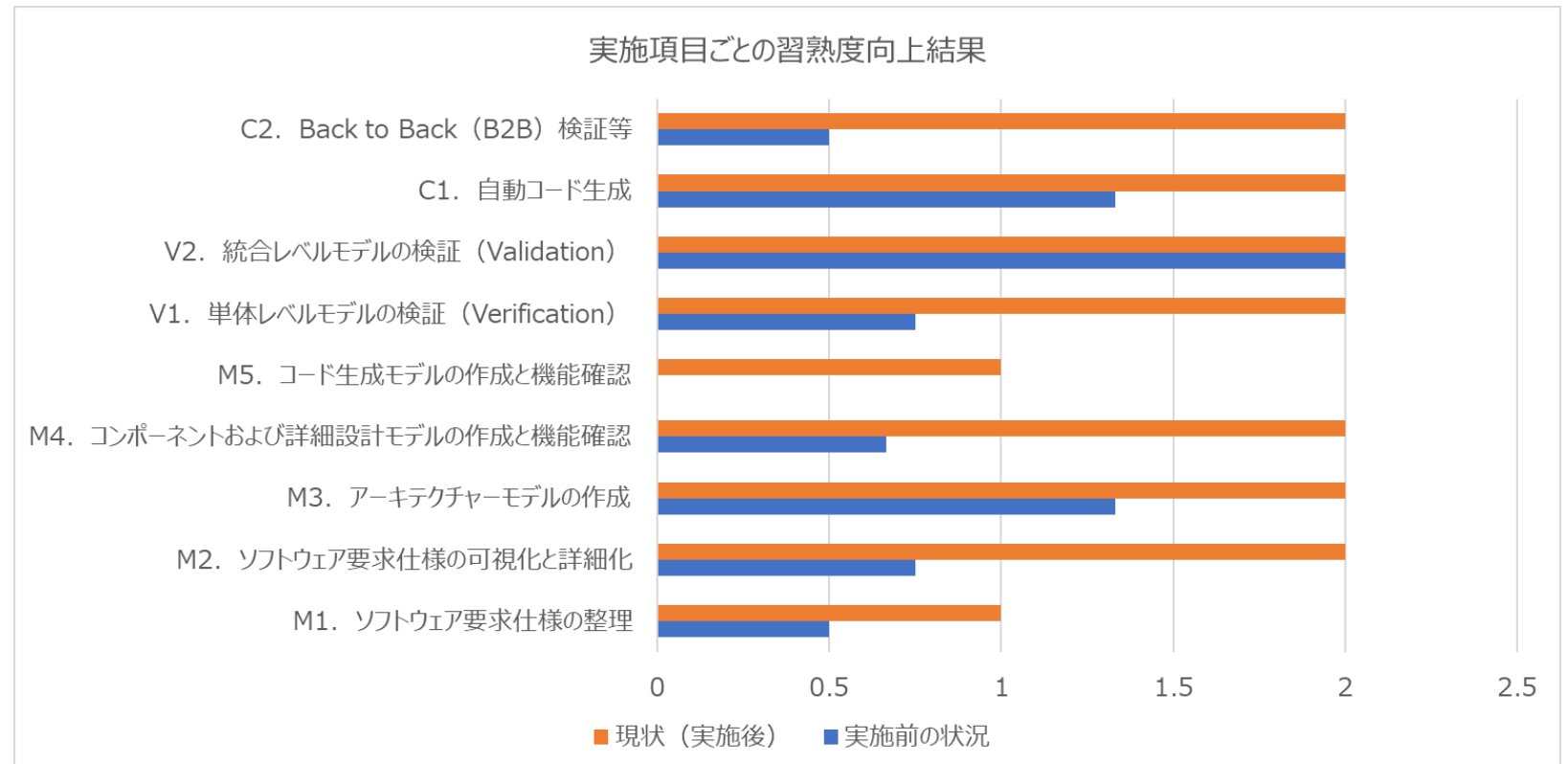
# プログラムの評価

- ・モデル作成、モデル検証、コード生成の各フェーズに関する一連の試行を行い、MATLAB/Simulink関連ツールの使用方法も含めて、その評価、課題の抽出を完了した
- ・当初目的であった、製品活用に向けた課題である、  
**モデル作成、モデル検証、コード生成に関する、社内標準規約の策定を完了させることができた**



## 評価指標

- 0 : 出来ていない
- 1 : 一部出来ている
- 2 : 出来ている



# 規格準拠に向けた課題

- 一方、DO-178C/331に準拠したMBDプロセスの構築にあたって、以下の課題もある
- ✓ DO-178C/331は、プロセスや成果物は定義されているが、それを**実現する具体的な手法は記載がなく**、詳細は各社で定める必要がある
  - ✓ PILS (Processor In the Loop Simulation)による、**実プロセッサ上でのカバレッジ取得が必要**なため、プロジェクト毎にPILS環境の構築が必要である
  - ✓ モデルとコードが等価であることを証明するために、**使用可能なSimulinkブロックの制限がある**ため、モデリング方法の新たな標準化が必要である

# Agenda

1. SUBARUの事業紹介
2. 無人航空機の開発概況とMBD適用事例
3. 標準化に向けた活動
4. **適用範囲の拡大に向けて**



# 適用範囲の拡大に向けて

システム開発として、更なる高品質化・効率化のため、  
MBDプロセスの適用範囲を拡大し、開発工程全体で高品質化・効率化を目指している

それには、以下のような組織の壁を越えた、適用範囲の拡大、標準化が必要

- ✓ 開発工程間の壁
- ✓ 専門分野間の壁
- ✓ メーカー間の壁

# 開発工程間の壁

現状の主なMBD適用範囲は、設計～製造～試験に関わる開発工程の一部のみ  
上流／下流へ適用範囲を拡大し、トレーサビリティの確保、モデルの活用による効率化を最大にする

要求分析、システム設計  
(MBSE) への活用

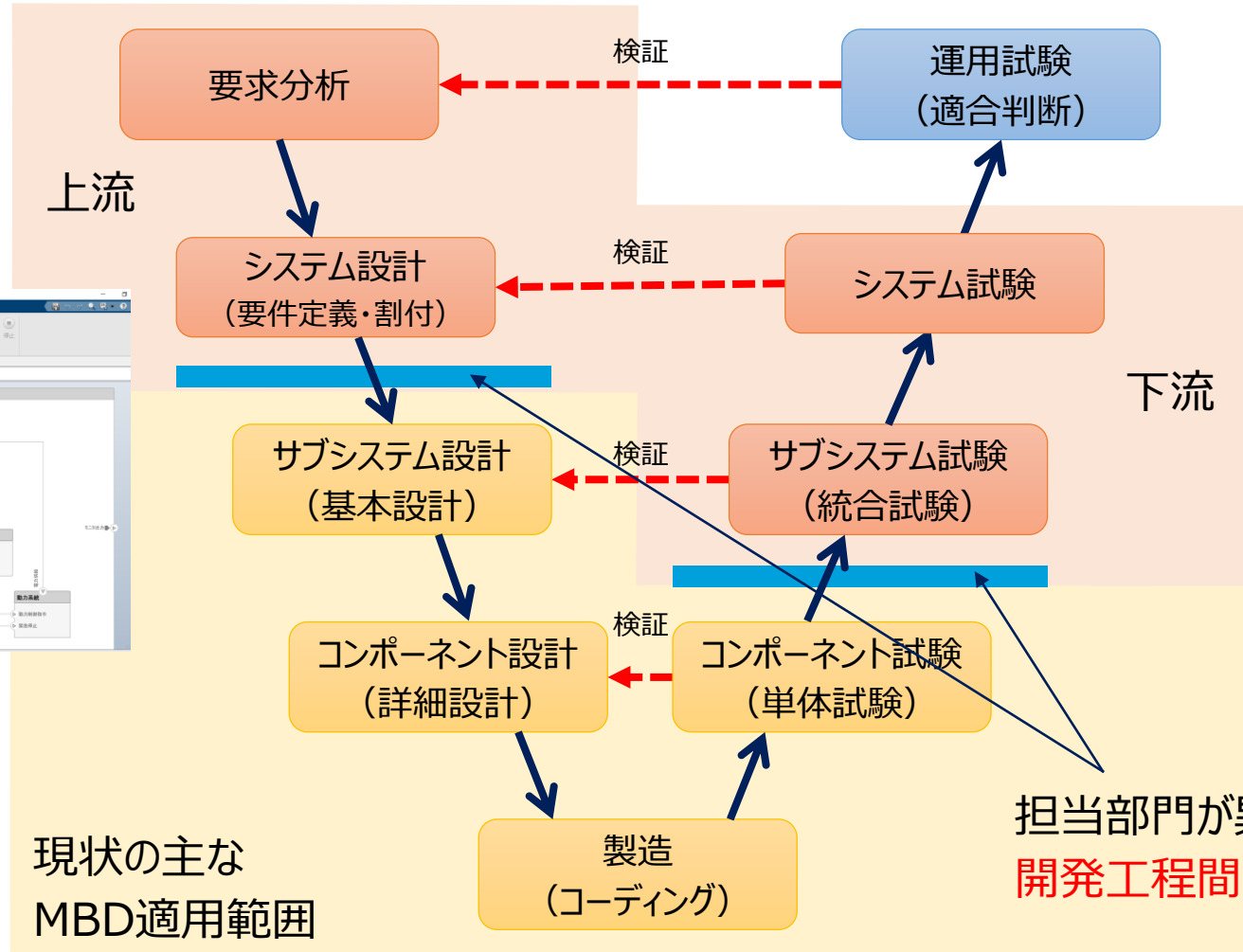
上流

下流

PILS、HILSの効率化

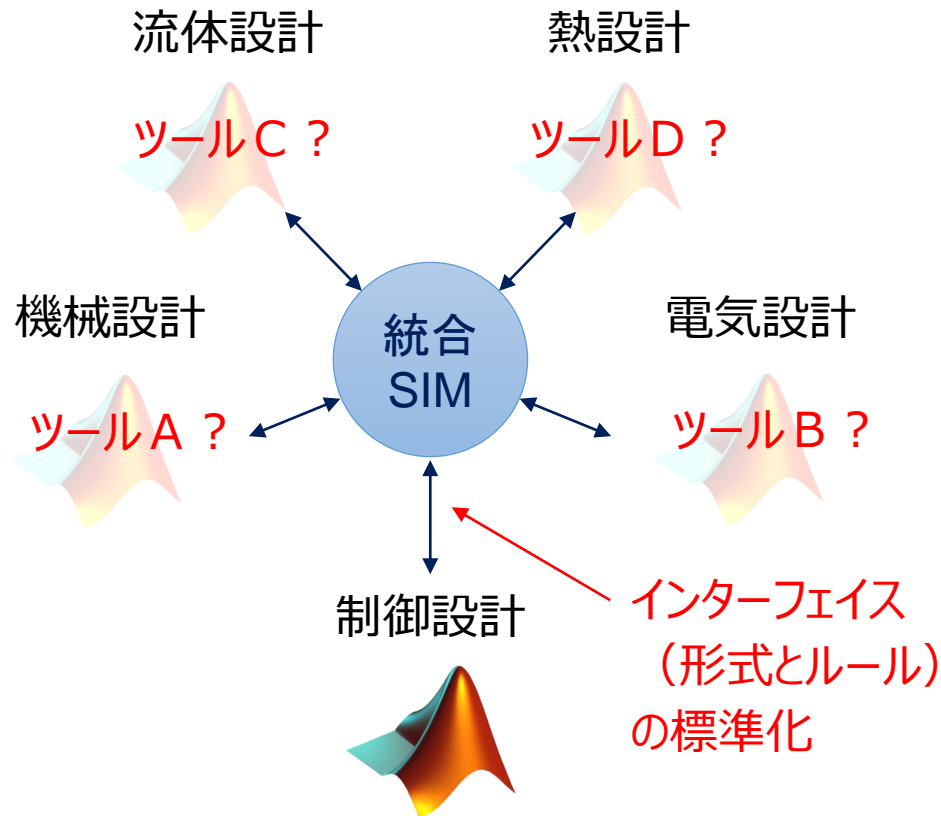
現状の主な  
MBD適用範囲

担当部門が異なることによる  
開発工程間の壁が有り (ボトムアップ)

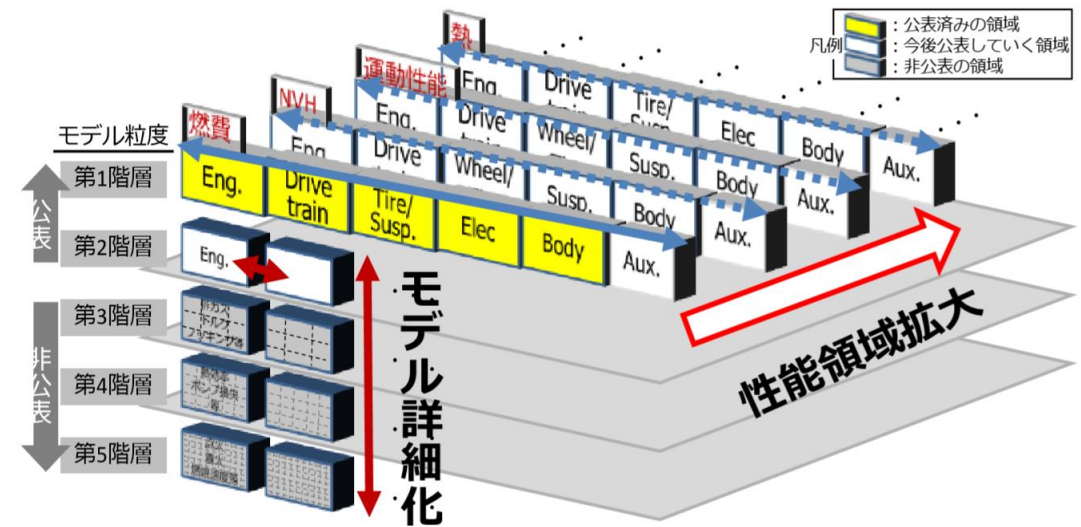


# 専門分野間の壁

- ・ 航空機をまるごと模擬したシステムレベルでのシミュレーションを行いたいが、各ドメイン(制御、機械、電気など)で適したツールと実績が異なるため、統一するのは非常に困難
- ・ 各ドメインのモデルが、専門分野の壁を越えてシームレスに活用できるよう、**モデル間のインターフェイス(形式とルール)を標準化**し、統合シミュレーションできるようにする



ジェネリックモデル(I/Fガイドライン準拠モデル)

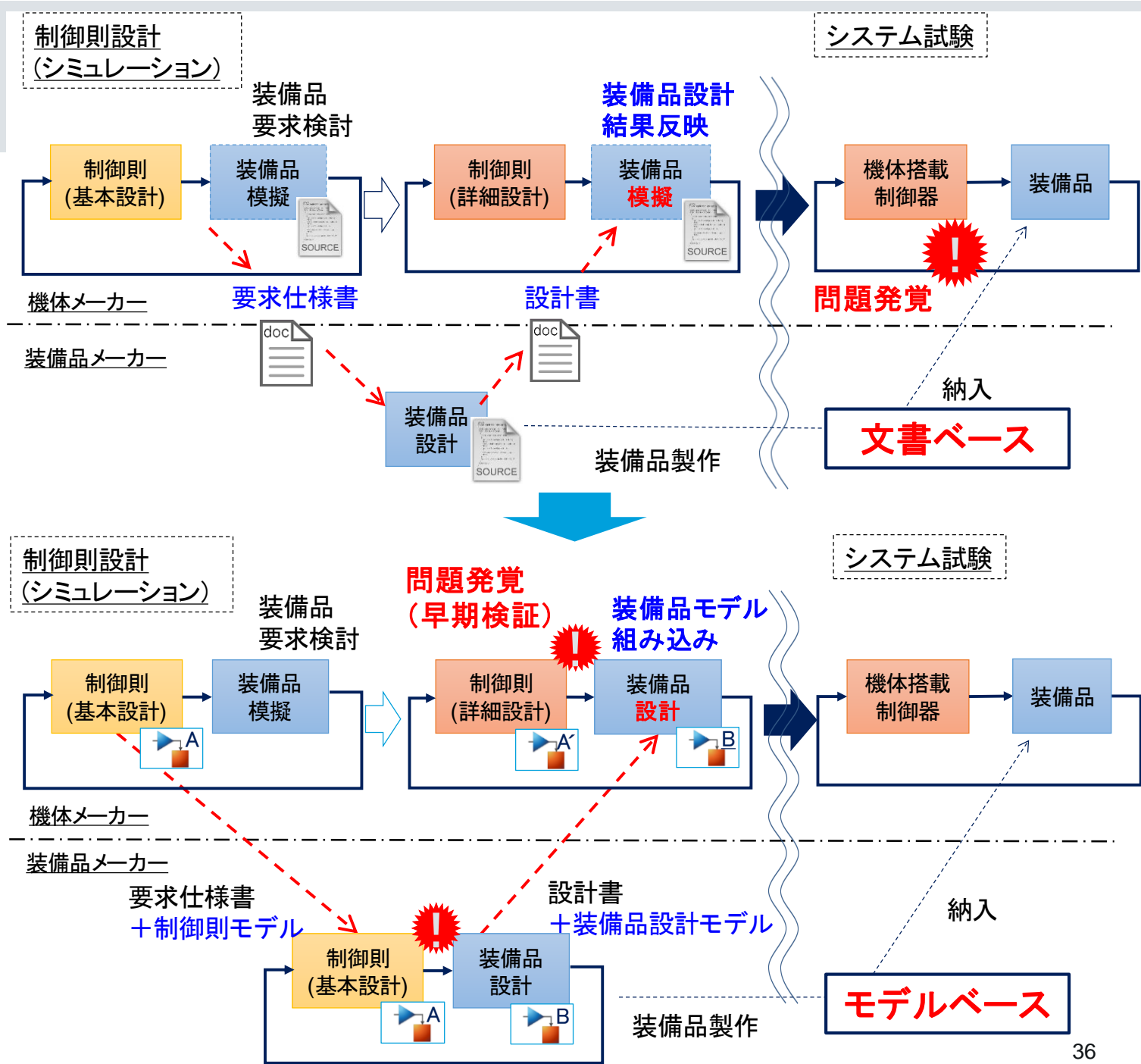


出典: MBD推進センター(JAMBE)  
<https://www.jambe.jp/uploads/20211008.pdf>  
 発足記念オンラインフォーラム資料(モデル流通推進委員会)より

# メーカー間の壁

- メーカー間の壁を越えて、モデルを授受できるようにする
- 仕様書や設計書等の設計情報を、文書からモデルに変えることで、メーカー間で同じモデルを使用し、認識齟齬の削減、設計段階での不具合洗い出しを図る

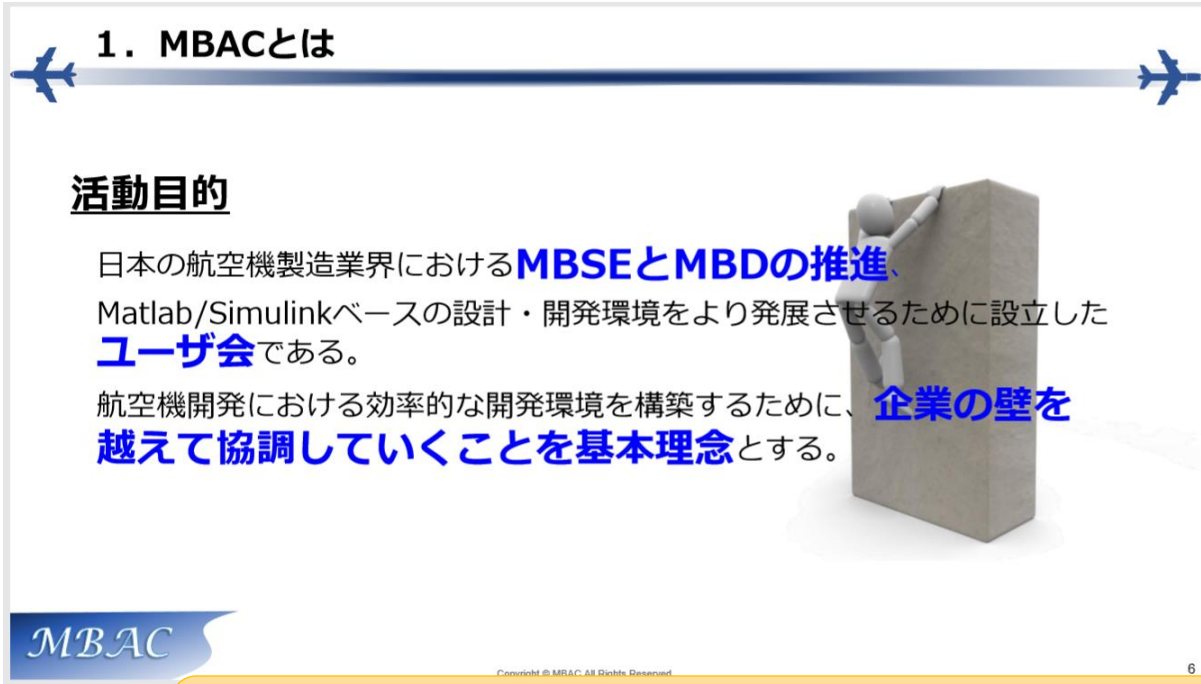
⇒ メーカー間でルールづくりが必要



# MBAC活動紹介

MBAC(Model Based Aviation Consortium)に  
コアメンバーとして参画

: 航空機開発におけるMBD技術情報交換会  
メーカー間の壁を越えた課題をWGとして解決していく



**ONE TEAMで世界に勝てる技術力を！ 航空機を！！**

# Thank You!



**SUBARU**