

Abstract

Model-Based Development (MBD) refers to a method where models that behave in the same way as the real world are created on a computer, allowing for desk-based verification before producing actual prototypes. By using this approach, the number of physical prototypes can be reduced, leading to lower development costs and shorter development periods.

At Yamaha Motor Engineering (hereinafter referred to as "the Company"), the acquisition of MBD technology is being advanced. This includes not only control and dynamics for mobility across all domains of land, sea, and sky, handled by the Yamaha Motor Group, but also the development of models capable of calculating disturbances like wind and tidal currents under environmental conditions. However, one challenge with applying MBD to mobility is that the simulation results are expressed in numerical form, making it difficult to understand the behavior as an actual phenomenon. As a unique initiative, the Company has developed a simulation system called the "1D Virtual Viewer," which visualizes MBD simulation results as animations of 3D-CAD data, making it easier to understand the behavior of mobility. This paper will first explain about the MBD technology and the 1D Virtual Viewer that was developed, using drones as a case study, and then introduce examples of how this technology has been applied to the marine and land domains.

1 はじめに

MBD (Model Based Development) とは、コンピュータ上で現実と同じ振る舞いとなるモデルを作成し、実機試作前に机上検証する手法である。この手法を用いることで、実機の試作回数を削減でき、開発コストの削減と開発期間の短縮につながる。

ヤマハモーターエンジニアリング(以下、当社)ではヤマハ発動機グループで扱う陸海空全ての領域におけるモビリティの制御や運動だけでなく、環境条件に関する風や潮流といった外乱の演算が可能なモデル開発を行いながら、MBDの技術獲得を進めている。

しかし、モビリティに適用した MBD では、シミュレーション結果が数値で表現されるため、その挙動を実際の現象として理解しづらいことが課題である。

そこで、当社独自の取り組みとして、MBD によるシミュレーション結果を3D-CAD データのアニメーションとして可視化することで、モビリティの挙動をより理解しやすくするシミュレーションシステム「1D Virtual Viewer」を開発した。

本稿では、はじめに当社の MBD 技術と開発した1D Virtual Viewer についてドローンを題材に説明し、次にマリンおよび陸域への本技術の展開事例について紹介する。

2 MBD 技術

本章では当社で取り組んだ内容についてマルチロータ型ドローンを事例に紹介する。システム概要図を図1に示す。ここではプロペラと制御について詳細を解説する。

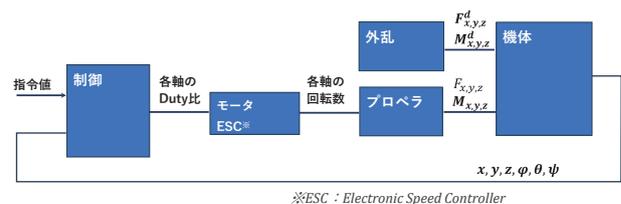


図1 ドローンシステム構成概要図

表1 図1の記号と定義

記号	定義	記号	定義
x	地上座標系 x 方向位置	$F_{x,y,z}^d$	外乱(風)による並進方向の力
y	地上座標系 y 方向位置	$M_{x,y,z}^d$	外乱(風)によるモーメント
z	地上座標系 z 方向位置	$F_{x,y,z}$	プロペラによる並進方向の力
φ	Roll 角	$M_{x,y,z}$	プロペラによるモーメント
θ	Pitch 角		
ψ	Yaw 角		

2-1. プロペラモデル

一般的に、プロペラの回転による推力は式(1)で算出される。

$$F = k\omega^2 \quad \dots(1)$$

表2 式(1)の記号と定義

記号	定義
k	変換係数
ω	プロペラ回転数

しかしプロペラを用いた回転翼機では大気速度、姿勢変化、外乱(風)の影響で式(1)では精度が不十分であり、プロペラが発生させる推力やモーメントの精度を高めるために、翼素理論をベースにした数式から力とモーメントを算出した^{[1][2]}。風洞にて様々な角度から風をプロペラに与えた際の3軸方向の力やモーメントを計測し、数式の一部パラメータを適合した上でモデルに織り込んだ。図2にパラメータ適合のための同定試験の様子を示す。

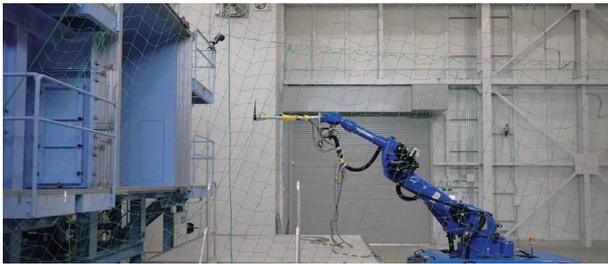


図2 プロペラモデル同定試験の様子

3軸の力は式(2)で実装する。

$$F_{x,y,z} = \rho S R^2 \Omega^2 \times C_{Tx}, C_{Ty}, C_{Tz} \quad \dots(2)$$

3軸のモーメントは式(3)で実装する。

$$M_{x,y,z} = \rho S R^3 \Omega^2 \times C_{Mx}, C_{My}, C_{Mz} \quad \dots(3)$$

表3 式(2)、(3)の記号と定義

記号	定義
ρ	空気密度
S	プロペラ回転面積
R	プロペラ半径
Ω	プロペラ回転数
C_{Tx}	x 軸推力係数(進行率、流入比等を用いて算出される)
C_{Ty}	y 軸推力係数(進行率、流入比等を用いて算出される)
C_{Tz}	z 軸推力係数(進行率、流入比等を用いて算出される)
C_{Mx}	x 軸モーメント係数(進行率、流入比等を用いて算出される)
C_{My}	y 軸モーメント係数(進行率、流入比等を用いて算出される)
C_{Mz}	x 軸モーメント係数(進行率、流入比等を用いて算出される)

比較結果の代表例を図3、4に示す。

回転数に対するz方向推力 実験とシミュレーション結果の比較

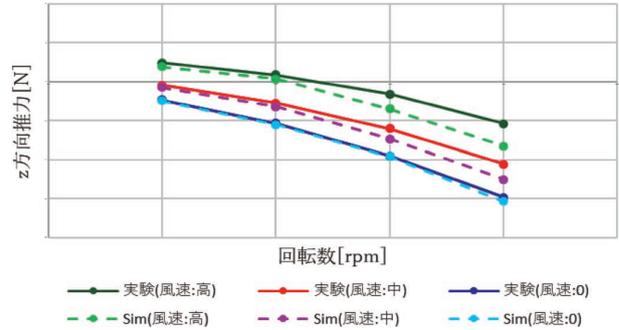


図3 回転数に対するz方向推力 比較結果

回転数に対するx方向推力 実験とシミュレーション結果の比較

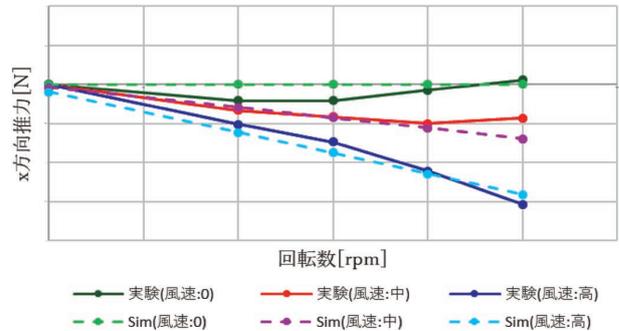


図4 回転数に対するx方向推力 比較結果

実機で計測した結果と同定結果がおおむね一致していることが確認できる。

このプロペラモデルを織り込むことで、様々な外乱(風)のある環境下での飛行試験が可能となり、実機試験のリソースを低減できると考えられる。

2-2. 機体モデル

また、外乱(風)はプロペラだけでなく機体への影響も大きい。そこで外乱(風)による機体への影響を織り込むために図5に示す風洞前のロボットアームに機体を固定し、様々な角度から風を与えた際の力とモーメントを計測する機体抵抗試験を実施した。



図5 機体抵抗試験の様子

プロペラや機体などのプラントモデルは Modelica[®]ベースの非因果モデルで作成した。因果モデルでの作成も可能であるが、機体全体を数式化する必要がないこと、可読性や今後のカスタム、拡張性を鑑みて非因果モデルを採用した(図6)。

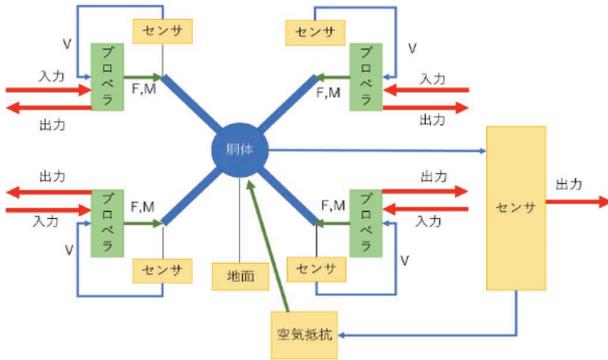


図6 プラントモデル全体概要図

※Modelica は Modelica Association の商標

2-3. 制御モデル

外乱(風)のある環境下で、ホバリングが比較的容易な位置制御を開発した。制御構成(代表例として x 軸、Pitch 軸のみ)を図7に示す。

制御は古典制御のフィードバック(以下 F.B.)制御を採用し、姿勢制御(角度 F.B.+ 角速度 F.B.)の上流に速度 F.B.を設置する構成とした。詳細は式(4)~(10)に示す。

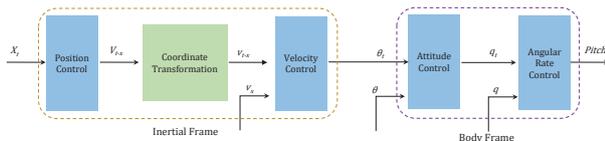


図7 位置制御構成図

$$V_{t-x} = K_{p1}(X_t - X) \quad \dots (4)$$

$$v_{t-x} = f(V_{t-x}, \varphi, \theta, \psi) \quad \dots (5)$$

※関数 f : 地上座標系速度から機体座標系速度への変換関数

$$e_{vx} = v_{t-x} - v_x \quad \dots (6)$$

$$\theta_t = K_{p2}e_{vx} + K_{I2} \frac{1}{s} e_{vx} - K_{D2} \frac{s}{\tau s + 1} v_x \quad \dots (7)$$

$$Q_t = K_{p3}(\theta_t - \theta) \quad \dots (8)$$

$$e_q = Q_t - q \quad \dots (9)$$

$$Pitch_c = K_{p4}e_q + K_{I4} \frac{1}{s} e_q - K_{D4} \frac{s}{\tau s + 1} q \quad \dots (10)$$

表4 式(4)~(10)の記号と定義

記号	定義	記号	定義
K_{p1}	位置制御の P ゲイン	X_t	x 軸目標位置
K_{p2}	速度制御の P ゲイン	X	x 軸現在位置
K_{p3}	角度制御の P ゲイン	V_{t-x}	x 軸目標速度(グローバル座標系)
K_{p4}	角速度制御の P ゲイン	v_{t-x}	x 軸目標速度(機体座標系)
K_{I2}	速度制御の I ゲイン	v_x	x 軸実速度
K_{I4}	角速度制御の I ゲイン	e_{vx}	x 軸速度の偏差
K_{D2}	速度制御の D ゲイン	θ_{tgt}	Pitch 角目標値
K_{D4}	角速度制御の D ゲイン	θ	実 Pitch 角
τ	微分フィルタ時定数	Q_t	Pitch 角速度目標値
		q	実 Pitch 角速度
		e_q	Pitch 角速度の偏差
		$Pitch_c$	Pitch 軸の制御量

2-4. 実機とシミュレータの比較

制御モデルとプラントモデル(プロペラモデルや機体モデルなど)を統合した環境を構築した。モデルの全体図を図8に示す。

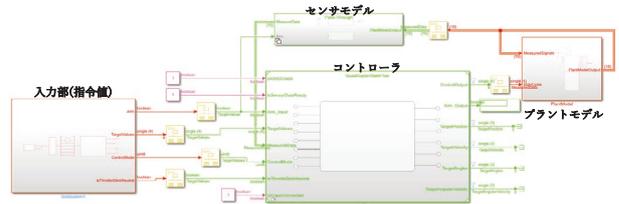


図8 ドローンのモデル全体像

この環境を用いて実機飛行試験と同条件でシミュレーションを実施した。比較結果を図9に示す。

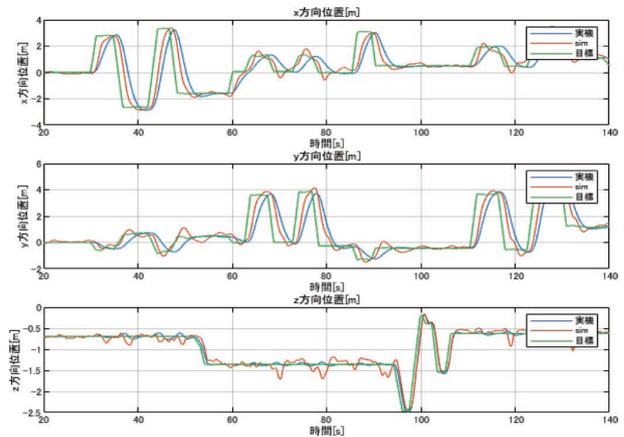


図9 実機とシミュレーションの位置制御の追従性比較

実機と同様に、シミュレータも目標位置に対し追従する傾向にあり、本シミュレータは実機と同様な振る舞いを再現できている。

シミュレーションの姿勢角のグラフを図10に示す。

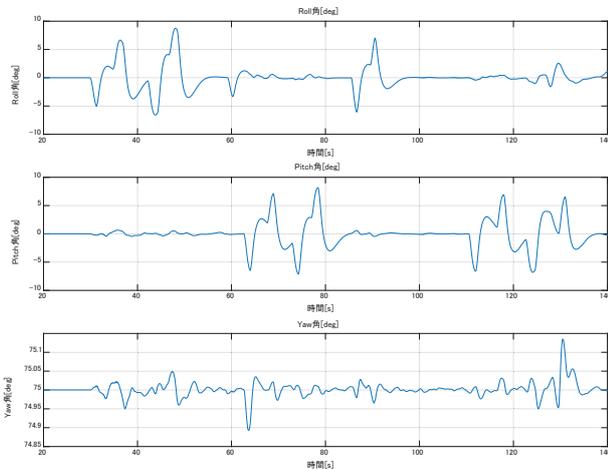


図10 位置制御で飛行させた際の姿勢角

図9、10のグラフだけでは、機体の挙動を想定することが難しい。飛行中の機体の挙動によらず目標が不変な場合は影響ないが、機体の挙動に応じて目標を補正操作する場合にはオペレータモデル(認知・判断・操作に相当するモデル)を織り込む必要があり、その開発に多くのリソースが必要となる。

これらの問題を解決するために、当社独自の取り組みとして機体挙動のシミュレーション結果をアニメーションとして視覚的に可視化するシミュレータ1D Virtual Viewerを開発した(図11)。



図11 改善前後の結果表示および操作方法イメージ図

3 1D Virtual Viewer

当社開発の1D Virtual Viewerは、制御モデルやプラントモデルなどの各要素モデルとゲームエンジンを連成させながら、シミュレーション結果をモビリティの挙動として3Dアニメーションで可視化する。これによりオペレータはモビリティの挙動を直感的に理解しながら、補正操作が可能となる。

1D Virtual Viewerのアーキテクチャを図12に示す。

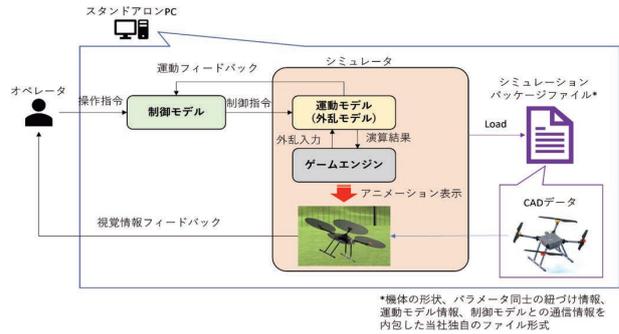


図12 1D Virtual Viewerのアーキテクチャ

次項で技術的な特徴を紹介する。

3-1. 制御、運動、外乱、ゲームエンジンのリアルタイム連携

各要素モデルは、FMU(Functional Mockup Unit)などの多様な形式で生成され、それぞれが異なる時間スケールで動作するため、これを調整する必要がある。

1D Virtual Viewerでは、それらを統合したシミュレーションを行うため、システムを構成する要素モデル全てを同じ時間スケールで動作するよう同期を取りながら実行可能とする時間同期メカニズムを織り込んだ。

これにより、異なる時間スケールで動作する各モデルの実行タイミングを調整しながら、同じペースで計算できるため、足並みが揃ったシミュレーション結果をゲームエンジン上で確認できる。

3-2. 連成シミュレーション設定

複数の要素モデルを連成する場合、その組み合わせパターンに応じて、相互でやり取りするパラメータを都度プログラミングしながら設定していく必要があり、多くの時間を要していた。

これを解決するため、1D Virtual Viewerでは制御や運動等の各要素モデルが持つパラメータ情報を読み込み、関連付けするパラメータ同士を容易に設定することが可能なユーザーインターフェース(以下UI)を導入した(図13)。

このUIを活用することによって、様々な要素モデルの連成パターンをプログラミングせずに素早く設定できるため、モデルの仕様に変更があった場合でも新たな仕様でのシミュレーションが早期に実施可能となる。

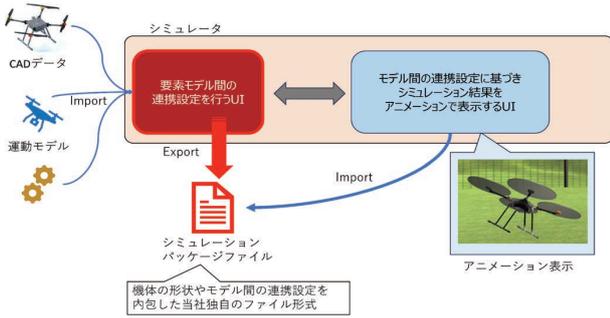


図13 連携設定 UI の活用イメージ

みでは、実艇ではなく全長約1.5mの模型船を対象とした。図15にシステム概要図を示す。ここでは、マリンシステムを構成する要素の代表例として「船外機」と「外乱」モデルについて記載する。

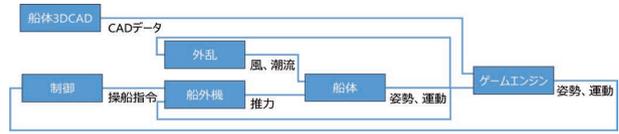


図15 マリンシステム構成概要図

船外機モデルの出力をプロペラ推力とし、式(11)で実装した。

$$T = \rho n^2 D^4 K_T \quad \dots(11)$$

表5 式(11)の記号と定義

記号	定義
ρ	密度
n	プロペラ回転数
D	プロペラ直径
K_T	推力係数

推力係数 K_T は図16のプロペラ単独性能試験により取得した。プロペラ模型を一定回転数で回転させながら一定速度の流速を与え、プロペラの推力とトルクを計測する。



図16 プロペラ単独性能試験の様子

外乱モデルは、風や潮流の向きと大きさを考慮し、船体に与える力を式(12)で実装した。

$$F = \frac{1}{2} \rho V^2 SC \quad \dots(12)$$

表6 式(12)の記号と定義

記号	定義
ρ	密度
V	外乱の速度
S	代表面積
C	抵抗係数

抵抗係数 C は図17の抵抗試験により取得した。模型船に一定速度の流速を与え、模型船に発生する抵抗を計測する。

4 1D Virtual Viewer を適用したマルチロータシミュレータ環境

ドローンの MBD 環境と1D Virtual Viewer 環境を統合した環境のメイン画面を図14に示す。



図14 ドローンシミュレータのメイン画面

メイン画面には姿勢角、速度、各軸の回転数、位置等の機体情報が表示され、操縦者目線や機体後方からの視点といったカメラ視点の変更も可能である。

このシミュレータ環境ではゲームエンジン側に風向、風量の設定機能を持たせており、周辺環境の構築と合わせた設定が容易となる。

また、外乱(風)要素の織り込みにより、実機では評価が難しい突風や、風が時々刻々と変化する環境下でもシミュレーション評価が可能になる。最適な機体・モータ・プロペラの諸元、制御ロジックや定数の提案が実機試作前に可能になり、開発期間短縮に貢献できる。

5 他フィールドへの展開

5-1. マリン領域

マリン領域にも同様の効果を適用するために、船外機を1基搭載した船舶を題材にシミュレータ環境を構築した。本取組

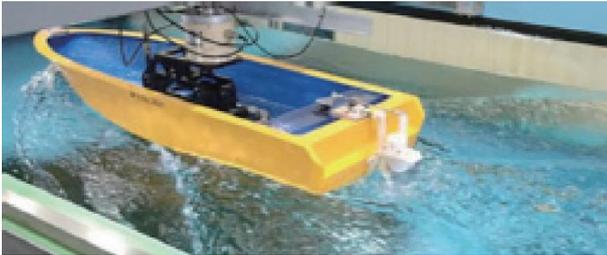


図17 抵抗試験の様子

モデルの計算精度を確認するため、模型船とモデルそれぞれで直進や定常円旋回などの基本的な運動の振る舞いを比較した。図18から、模型船と同等の振る舞いを再現できていると考えられる。

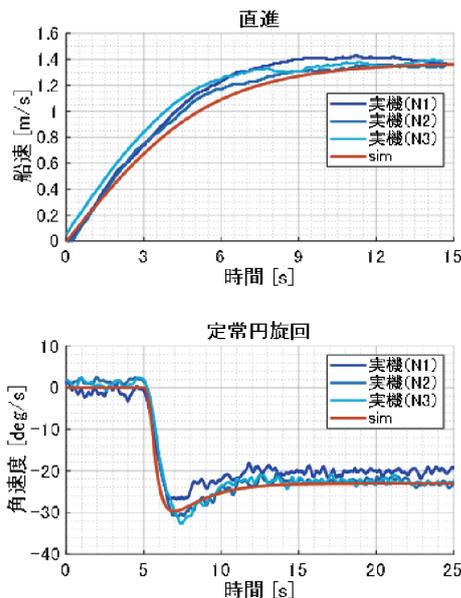


図18 実機とシミュレーションの模型船の挙動比較

1D Virtual Viewer 環境とドローンと同様に上記モデルを統合した環境のメイン画面を図19に示す。メイン画面には、船外機の推力や回転数、船体姿勢、操船情報などが表示されている。

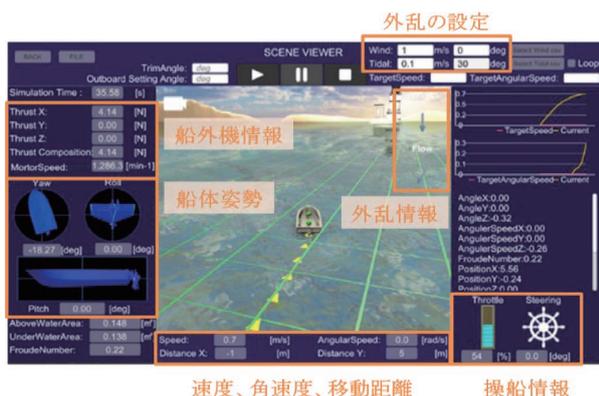


図19 マリンシミュレータのメイン画面

5-2. ランド(二輪)領域

ランド領域にも同様の効果を適用するために、パワートレイン(モータ)をプラントモデルとした1D Virtual Viewer 環境を構築した。そのメイン画面を図20に示す。



図20 ランド(二輪)シミュレータのメイン画面

本環境を用いて段差乗り越し時の走破性や、登坂時の必要駆動力の検証に活用した。

6 おわりに

当社で取り組んでいる MBD、それに可視化を連成させた技術について紹介した。これは陸海空のモビリティ全てに適用できる技術である。1D Virtual Viewer の活用により、実機では評価が難しい様々な条件のシミュレーションや、対象となるモビリティの挙動を確認しながらの補正操作が可能である。これらにより開発期間の短縮につながる事が想定される。

今後はユーザビリティの向上や拡張性などに取り組み、ヤマハ発動機グループ内に広げていき、開発効率化につなげていきたい。

■参考文献

- [1] 航空工学(1) 東昭 裳華房 1989
- [2] ヘリコプタ入門 加藤寛一郎、今永勇生 東京大学出版会 1985

■著者



堀川 雅弘
Masahiro Horikawa
ヤマハモーターエンジニアリング(株)
先行技術開発部



太田 博康
Hiroyasu Ota
ヤマハモーターエンジニアリング(株)
先行技術開発部



松清 一樹
Kazuki Matsukiyo
ヤマハモーターエンジニアリング(株)
先行技術開発部