B-005

高信頼性ロケット飛行制御ソフトウエアの検証手法の提案

A Proposal for Verification Method of High Reliability Flight Software

高橋 正和†　　**水越 紀良**††　　**津田 和彦**†††
Masakazu Takahashi　　Noriyoshi Mizukoshi　　Kazuhiko Tsuda

１．まえがき

本論文では，ロケットの飛行制御ソフトウエア（Flight Software: FSW）を効率的に検証する手法を提案する．ロケットのミッションの目的は，搭載される衛星の軌道投入であるが，飛行中の環境条件変化や機器不調等により，衛星の軌道投入が困難となる事態が発生する．その場合，FSW は適切なリカバリ（飛行経路再計算，故障部分分離）を行い，ミッションを継続する．リカバリを適切に実施するには十分な FSW の検証が必要となるが，そのためには FSW と実際のロケットを組み合わせた検証（以下，結合検証）が必要である．しかし，ロケットの完成は開発工程の後半以降であり，十分な検証期間を確保することが困難となる．そこで，ロケットの完成に先立って，結合検証に相当する検証を実施する手法が必要となる．

２．ロケット飛行制御ソフトウエア検証手法

現状の FSW 開発工程を図 1(a)に示す．FSW の結合検証はロケット完成まで開始することができず，待ち時間が生じる．このため，結合検証工程が圧迫される．加えて，結合検証期間中に要求仕様の誤りに起因する不適合が発見され，大幅な修正作業が必要となるケースが発生する．これらの追加・修正作業のために，FSW の検証期間の長期化と信頼性の低下が生じている．

表１　FSW の追加・修正の原因と内訳

分類	原因	内訳
1	要求仕様自体の誤り	約 40%
2	ソフトウエア制作上の誤り	約 30%
3	ハードウエアとソフトウエアの インターフェースの誤り	約 20%
4	その他	約 10%

追加・修正作業の原因内訳を表 1 に示す．分類 3 については，ハードウエアに依存するが，分類 1・2 は FSW のみでの検証（要求仕様や処理手順の妥当性確認）が可能である．従って，FSW 開発時に生ずる不適合の約 70%は結合試験実施前に解決が可能である．これらの検証を行うためには，機体を模擬する機体シミュレータが必要となる．さらに，発生した不適合に対して，FSW 中の故障個所の識別・修正を支援する検証支援ツールが必要となる．以降，機体シミュレータと検証支援ツールを高度信頼性飛行制御検証ツール（FCVT: Flight Control Verification Tool）と呼ぶ．

†　株式会社　ギャラクシー・エクスプレス
††　石川島播磨重工業　株式会社
†††　筑波大学大学院ビジネス科学研究科企業科学専攻

図 1 (b)に示す高度信頼性飛行制御検証支援ツールを用いた FSW 開発工程を実現し，事前検証を行うことで，検証期間を短縮し，開発期間全体の 20%削減を実現する．

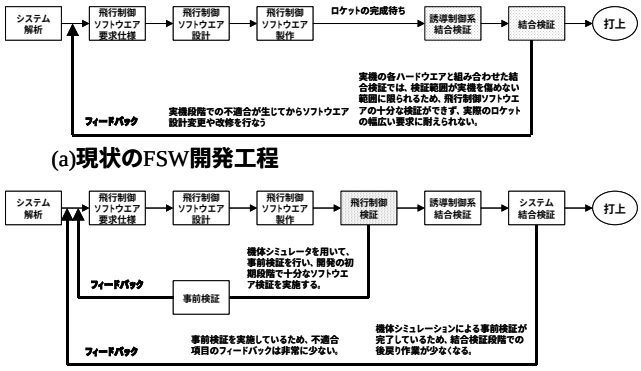

図１　FSW 開発工程の比較―現状と提案手法―

３．高度信頼性飛行制御検証ツールの達成目標と構成

本章では，はじめに FCVT を構成する機体シミュレータと検証支援ツールの達成目標と構成について述べる．

3.1　機体シミュレータ(Vehicle Simulator)

VS の達成目標は，異なる種類・構成のロケットの運動モデルを短期間で作成できることである．

VS の構成要素である(a)機体・(b)環境・(c)射場設備の各モデルについて，以下に説明する[1],[2],[3]（図 3 の機体シミュレータ部分）．

(a)　機体モデルは，汎用部分と特化部分から構成される．汎用部分は全ての機体で共通な機能であり，特化部分は機体毎に異なる機能である．汎用部分は，入出力モデル・運動モデル・機体構成管理機能からなる．さらに，入出力モデルは，機体を構成する機器を模擬するモデルであり，バルブ・モータ・アクチュエータ等がある．運動モデルは，機体の制御式を記述したものであり，比例制御・微分制御・積分制御・遅れ・飽和等がある．機体構成管理機能は，実際の機体の構成に従って入出力モデルと運動モデルを組み合わせ，機体モデルを実現するものである．

(b)　環境モデルは，飛行中の環境状態を模擬するものであり，気象・大気・重力のモデルから構成される．

(c)　射場設備モデルは，飛行前の機体準備作業を模擬するものであり，射点信号・推進薬供給・加圧設備・発射管制システムのモデルから構成される．

3.2　検証支援ツール(Verification Tool)

VT の達成目標は，不適合個所を効率的に同定できるこ

とである．VT では，二分岐木とプログラムスライス[4]を用いたデバック手法を採用した．検証手法の概要を以下に示す．①不適合が発生したソフトウエアの分岐構造に着目して二分岐木を作成し，②全ての分岐の前後で不適合の有無を確認して不適合を含む範囲を識別し，③その範囲に対して着目変数のプログラムスライスを実施して不適合部分の絞り込みを実施する（図２）．

VT の構成要素である(ｱ)重要度分析，(ｲ)検証条件自律設定，(ｳ)検証結果予測，(ｴ)検証結果自律評価の各ツールについて以下に説明する（図３の検証支援ツール部分）．

(ｱ)　重要度分析ツールは，機体の飛行に影響を与える変数を抽出するツールである．検証に先立って，重点的に検証を行う変数を識別するために使用する．

(ｲ)　検証条件自律設定ツールは，各変数の変動範囲から検証データを自動生成するツールである．

(ｳ)　検証結果予測ツールは，検証データから予測結果を生成するツールである．

(ｴ)　検証結果自律評価ツールは，予測結果と実際の結果を比較し，検証結果の判定を行うツールである．

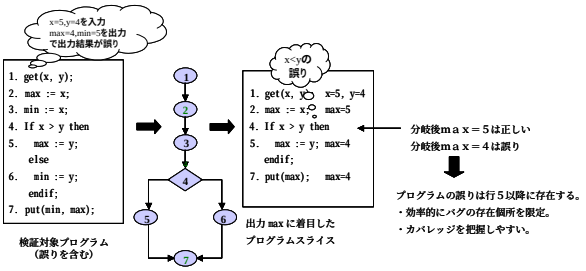

図２　提案する不適合箇所同定手法の概要

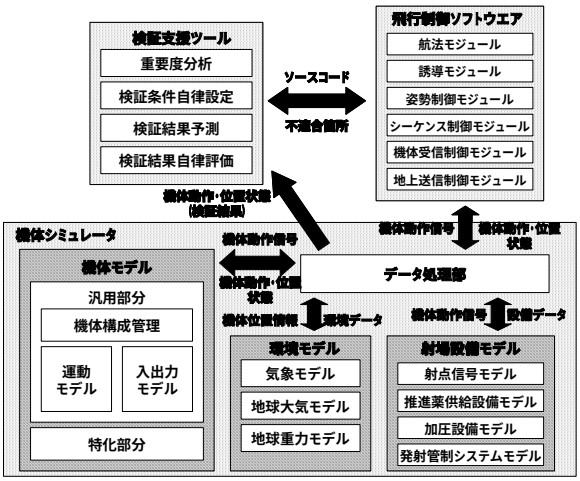

図３　FCVT の構成―VS と VT―

４．高度信頼性飛行制御検証ツールの評価

FCVT を実現するために必要な項目について評価した．

はじめに，VS の汎用モデルを使用して，既存の機体の運動モデルを記述できるか思考実験をした結果，入出力モデルの 100%をカバーできた．

次に，機体の動作制御式が既決の場合，運動モデルを記述することは VS を用いて機械的に実施できる．従って，機体の機器構成に応じて，これらの入出力モデルと運動モデルを組み合わせることで，機体シミュレータを機械的に作製できる．以上より，VS を用いて検証対象機体に適合したシミュレータを容易に実現できるものと考えられる．

表２　従来・提案の検証手法の効率比較

プログラム	従来手法			提案手法		
	発見率	発見時間	検証行数	発見率	発見時間	検証行数
A	100%	24.3 分	38 行	100%	10.0 分	4 行
B	100%	22.3 分	46 行	100%	18.0 分	3 行
C	100%	16.7 分	17 行	100%	13.0 分	3 行

最後に，不適合個所の検出時間に関する評価を行う．不適合を含むソフトウエアに対し，提案ツール使用の有無による不適合の発見率と発見時間を調査した．結果を表２に示す．発見率に変化はないが，発見時間が平均で約 65%となることが確認できた[5]．これは，二分岐木とプログラムスライスを用いて不適合範囲を大幅に絞り込んだため，プログラマがチェックする部分が減少したためと考えられる．また，二分岐木を用いて検証条件自律設定を実施することで，分岐を網羅した検証条件を，ほぼ自動で作成することができ，検証データ作成時間は，ほぼゼロになった．

５．終わりに

標準的な FSW 開発期間は約 60 ヶ月であり，そのうち，設計・制作期間は約 38 ヶ月，検証期間は約 22 ヶ月である．検証期間の内，約 7 ヶ月は検証ケースの作成，約 15 ヶ月は検証・修正作業である．本研究の結果，FCVT により検証ケース作成の時間は，ほぼゼロに，検証・修正の時間は約 2/3 になることが確認できた．従って，検証期間は約 10 ヶ月となり，開発工程全体では 48 ヶ月となり，開発時間の 2 割短縮が可能となると考えられる．今後は，試作した部位以外の開発を行う予定である．

謝辞

本研究は，新エネルギー・産業技術総合開発機構（NEDO）からの平成 14 年度業務委託「次世代輸送系システム設計基盤技術開発プロジェクト」により実施しました．この場を借りてお礼を述べます．

参考文献

[1] 高橋正和，津田和彦：効率的なプラントソフトウエアの要求定義手法，情報処理学会論文誌，Vol.42,No.3,pp518-528(2001.03)

[2] TAKAHASHI M, and TSUDA K.: An Efficient Integrated Development Environment of Plant Control Software, IEEJ Trans (2003.9.掲載予定)

[3] TAKAHASHI M., TERANO T.and TSUDA K.：" A study of estimation of both execution time and LOC of embedded system", 3rd International Workshop on Emergent Synthesis – IWES'01,pp187-192(2001.03).

[4]Weiser M．: Program Slicing, IEEE Trans． Software Engineering, Vol.10 No．4, pp．352-357(1984)

[5]平成 14 年度「次世代輸送系システム設計基盤技術開発 高度信頼性飛行制御検証技術」成果報告書