

大規模複雑 Simulink モデルのためのテンプレートベース Monte-Carlo 法に基づいたテスト自動生成ツール

富田 堯^{1,a)} 石井 大輔^{2,b)} 村上 徹^{3,c)} 竹内 成樹^{3,d)} 青木 利晃^{1,e)}

概要：車載システム等の制御ソフトウェアの開発では、MATLAB/Simulink を用いたモデルベース開発が広く用いられている。本稿では、モデル及びソフトウェアの開発効率化及び高品質化を目的とした、Simulink モデルのためのテスト自動生成ツールを紹介する。本ツールは、(1) 高カバレッジを達成するテストスイートを自動生成し、(2) 判定・条件・MC/DC カバレッジを代表的既存ツールよりはるかに高速に計測することができる。このテストスイート自動生成では、各テストケース（入力時系列信号データ群）を予め用意したテンプレートを基にした Monte-Carlo 法を用いており、代表的既存ツールではテスト生成不能な大規模かつ複雑なモデルも扱うことが可能である。

キーワード：モデルベース開発，自動テスト生成，修正条件/判定網羅，信号テンプレート，Monte-Carlo 法，MATLAB/Simulink

Template-Based Monte-Carlo Test Generator for Large and Complex Simulink Models

TOMITA, TAKASHI^{1,a)} ISHII, DAISUKE^{2,b)} MURAKAMI, TORU^{3,c)} TAKEUCHI, SHIGEKI^{3,d)}
AOKI, TOSHIKI^{1,e)}

Abstract: MATLAB/Simulink is a de-facto standard tool in model-based development (MBD) for control software of automotive systems. In this paper, we introduce an automated test generation tool for developing a high-quality Simulink models efficiently. The tool provides functions for (1) generating automatically a high-coverage test-suite for practical models, which cannot be handled by SLDV, and (2) measuring decision, condition, MC/DC coverage much more efficiently than Simulink Coverage. This automatic test-suite generation is a Monte-Carlo method based on templates of test cases (i.e., groups of time-series signals for inputs).

Keywords: Model-based development, Automated test generation, Modified condition/decision coverage, Signal template, Monte-Carlo method, MATLAB/Simulink

1. はじめに

1.1 背景

電子制御技術の発達に伴い、自動車/航空/宇宙産業では、多くの部品の電子化が進められてきた。同時に、それらの

電子制御装置 (electronic control unit; ECU) の安全性と信頼性をいかに担保するかが大きな問題となっており、国際標準として、一般電子機器向けの基本的機能安全規格 IEC 61508^{*1}や、それを自動車向けに特化した機能安全規格 ISO 26262^{*2}が策定された。近年、多くの企業がハイブリッド/電気自動車及び自動運転技術の開発に注力し、部品の電子化及び電子制御部品の大規模化/複雑化はさらに加速すると予想されており、安全で信頼性のある電子制御ソフトウェアを効率的に開発するための技術が強く求められている。

¹ 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology

² 福井大学
University of Fukui

³ ガイオテクノロジー株式会社
GAIO Technology Co., Ltd.

a) tomita@jaist.ac.jp

b) dsksh@u-fukui.ac.jp

c) murakami.t@gaio.co.jp

d) takeuchi.s@gaio.co.jp

e) toshiaki@jaist.ac.jp

^{*1} <http://www.iec.ch/functionalsafety/standards/>

^{*2} <https://www.iso.org/obp/ui/#iso:std:iso:26262:-1:ed-1:v1:en>

高安全かつ高信頼な電子制御ソフトウェアの開発には、**モデルベース開発 (model-based development; MB)** が広く適用されている。通常、制御器 (コントローラ) は制御対象 (プラント) をフィードバック制御する。モデルベース開発では、制御器と制御対象が共に数学的・形式的にモデル化され、シミュレーションによりその振る舞いを確認できる。そして、仕様モデルである離散システムに対して、データの定義、コード生成の設定及び実装のために必要となる付加機能 (オーバーフロー対策など) を追加して実装モデルを作成する。上流工程の段階から詳細な妥当性検査と検証が可能であり、最終的に実装モデルをコードへ自動変換するという統合的開発工程により製品の品質を保証する。このようなモデルベース開発においては、数値計算、モデリング、シミュレーション環境を提供する **MATLAB^{*3}/Simulink^{*4}** が広く利用されている。

IEC 61508 や ISO 26262 ではソフトウェアの品質を保証するために形式手法を適用することが推奨されているものの、増々大規模化・複雑化する現実的なモデル/ソフトウェアの品質を保証する現実的な手段はテスト (testing) である。モデルに対するテストでは、各テストケースに対する振る舞いをシミュレーションし、上流/下流のモデルとの一致性検査等が実施されている。コードに対するホワイトボックステストの品質基準として、

- 判定網羅 (decision coverage; DC) ,
- 条件網羅 (condition coverage; CC) ,
- 修正条件/判定網羅 (modified condition/decision coverage; MC/DC)

などがあるが、モデルテストにおいても同様の基準がある。典型的には、カバレッジの高いテストスイートでのテストに合格することが求められており、ISO 26262 でも MC/DC カバレッジテストの実施が要求されている。MATLAB/Simulink では、設計エラー検出やテストケース自動生成する機能を提供する Simulink Design Verifier^{*5} (SLDV) や、テストスイートのカバレッジを計測する機能等を提供する Simulink Coverage^{*6*7} (SLC) などの IEC 61508 及び ISO 26262 に準拠したツールボックスによって、効率的なテストを支援している。

1.2 課題

MATLAB/Simulink が提供するモデルベース開発環境にはいくつかの課題がある。1 つ目の課題は、高カバレ

ジテストスイートの作成が困難な点である。SLDV は、判定・条件・MC/DC カバレッジ等の各テスト項目に対するテストケースを自動生成する機能を有するものの、実際の規模と複雑さをもったモデルは解析不能である。これは SLDV が主に形式手法による解析を基にテストケース生成を試みるためである。非線形演算や特殊演算などは制約充足解析等の形式的解析を決定不能又は決定困難にするため、形式手法のみに頼らないテストケース自動生成手法が求められている。そこで、テストケース自動生成では、**Monte-Carlo 法**、即ち、テストケースを無作為に生成し、カバレッジに寄与するものをテストスイートに加えていくことでカバレッジを最大化する手法がしばしば採用されている [1], [2], [3]。ただし、制御モデルに対するテストケースは時系列データであるため、そのテストケース標本空間は非常に巨大である。従って、効果的に乱択探索するための工夫も必要である。

2 つ目の課題は、SLC が提供するカバレッジ計測関数 `cvsim` の実行速度が非常に遅い点である。カバレッジ計測は、(i) シミュレーションと (ii) 各テスト項目の合致/不合致な時刻数の数え上げの 2 工程から成るが、MATLAB/Simulink 標準のシミュレーション関数 `sim` と比較して、`cvsim` 関数は著しく低速である。つまり、高カバレッジテストスイートを作成することができたとしても、その合否判定に多大な時間を要する。そのため、効率的なモデル/ソフトウェア開発には、高速なカバレッジ計測が要求されている。特に、Monte-Carlo 法では、多くの候補テストケースを生成し、そのシミュレーションを実行した上でカバレッジ情報を収集することになるため、カバレッジ計測の高速化は必要不可欠である。

1.3 貢献

本研究では、前節の課題を解決するためのテスト自動生成ツールを開発した。本ツールは以下の特徴をもつ。

- 高カバレッジを達成するテストスイートを自動生成する。
- 判定・条件・MC/DC カバレッジを高速に計測する。

テストスイートの自動生成においては、[4] で提案されているテンプレート (template) ベースの Monte-Carlo 法を拡張して用いる。Monte-Carlo 法を用いることで、形式手法では解析が困難/不能な非線形算術等を多数含むモデルであっても、扱うことが可能となる。また、より現実的かつ理解可能な入力信号のみを考慮できるよう、テストケースを構成する各入力信号に対してテンプレートを用意し、その範囲でテストケースを探索する。

カバレッジ計測においては、高速シミュレーションモードと配列/行列演算の両方を活用できる手法を採用、実装する。それにより、Monte-Carlo 法によるテストケース生成効率の底上げが可能となる。

^{*3} <https://www.mathworks.com/products/matlab.html>

^{*4} <https://www.mathworks.com/products/simulink.html>

^{*5} <https://www.mathworks.com/products/sldesignverifier.html>

^{*6} <https://www.mathworks.com/products/simulink-coverage.html>

^{*7} 2017b 版から Simulink Verification and Validation (V&V) のカバレッジ計測機能が切り出され、1 つのツールボックスとして独立した。

2. 準備

2.1 MATLAB/Simulink

MATLAB は、数値計算環境であり、高速な配列/行列演算や画像/信号処理のための専用命令群などの様々なライブラリを標準としてもつ他、特定領域向けの機能をパッケージ化したツールボックスを多数もつ。

Simulink は、MATLAB と連携する、モデリング及びシミュレーション環境である。Simulink モデルは、一種のブロック図であり、様々な種類のブロックとラインから構成される動的システムをグラフィカルに表現する。ブロックは入力/出力、算術演算、比較演算、論理演算、(多入力)スイッチ、サブシステムなどの1処理単位であり、それらの間を連結するラインが論理、整数、浮動/固定小数点などの型のデータを各時刻毎に伝送する。また、サブシステムブロックにより、システムの階層的構造化が可能である。そして、シミュレーションにおいては、以下のようなモードを利用できる。

- **Normal:** MATLAB 環境上で直接シミュレーションを行う。
- **Accelerator:** メモリ内に実行エンジンを生成し、そこでシミュレーションを行う。
- **Rapid accelerator (Rapid):** スタンドアロンの実行ファイルを外部に生成し、そこでシミュレーションを行う (シミュレーション毎に前処理が必要)。

なお、Normal/Accelerator モードでは、初回のシミュレーション時に1度だけコンパイルを行い、次回以降の同設定のシミュレーションでは再コンパイルが不要な高速リスタート (**fast restart**) 機能が有効化できる。また、Accelerator/Rapid モードは、MATLAB 環境外でシミュレーションが実行されるため、API による通信が必要である。

2.2 テストケース/テストスイート

典型的には、モデルに対するテストは、ハーネスモデル (**harness model**) を用いて実施される。ハーネスモデルは、テスト対象を複製したサブシステムと、テスト対象に供給する入力信号を生成するシグナルビルダ (**Signal Builder**) ブロックから構成される。

ハーネスモデルにおいては、シグナルビルダブロックの1信号グループが1テストケースに対応する。即ち、このシグナルビルダブロックを編集し、グループを追加・修正することで、テストスイートが生成される。

2.3 テスト項目

ホワイトボックステストにおいて、網羅性 (**coverage**) とは、対象の構成部品がどれだけもれなく駆動されるかの指標である。Simulink モデルに関しては、判定、条件、MC/DC 網羅基準の測定対象は、モデル上のデータ流のパ

ターン、即ち、論理的特性を変化させるブロック*8である。
DC テスト項目: 判定網羅基準では、各ブロックのそれぞれの判定結果が出現することがあることを確かめる。

- 対象ブロック: 論理演算、(多入力) スイッチ、制御ポート付きサブシステムなど。
 - 判定結果: ブール値 (論理演算)、通過させる入力ポートの番号 (スイッチ)、有効/無効 (サブシステム) など。
- CC テスト項目:** 条件網羅基準では、各ブロックのそれぞれの条件が真にも偽にもなることを確かめる。
- 対象ブロック: 比較演算、論理演算など。
 - 条件: 比較述語 (比較演算)、原子条件 (論理演算) など。

MC/DC テスト項目: MC/DC 網羅基準では、各ブロックのそれぞれの条件が判定結果に独立に影響を与えることを確かめる。

- 対象ブロック: 多入力の論理演算など。
- 証拠対: 「あるときにすべての入力 (原子条件) が同時に真となる」と「あるときに i 番目の入力のみが偽でそれ以外がすべて真となる」 (論理積演算の i 番目原子条件の独立影響) など。

3. ツール概要

本ツールの概略を図1に示す。なお、本ツールは、MATLAB/Simulink の標準ライブラリのみを用いた MATLAB スクリプトとして実装されている。なお、解析対象モデルは、ハーネスモデルと仮定する。

3.1 構造解析器

構造解析器 (structure analyzer) では、カバレッジ計測やテストケース生成の前準備として、モデルの静的構造解析を行い、カバレッジ計測対象ブロック及びそれらに制御信号を発する上流ブロックを列挙する。

3.2 テストケース生成器

テストケース生成器 (test case generator) は、構造解析器で検出された最上位シグナルビルダブロックを編集し、与えられた入力信号テンプレート (4.1 節) を基にテストケースを生成する。テンプレートは少数のパラメータによって時系列データ全体を特徴付けるものであり、それらのパラメータの値を選択することでテストケースが定まる。

本ツールは、以下の2つのモードでランダムにテストケースを生成できる。なお、乱数生成には MATLAB 標準の乱数生成関数 **rand** (既定設定はメルセンヌツイスタ) を用いる。

一様無作為抽出モード: 与えられたテンプレートのパラメータ値を一様無作為に乱択することで、新たなテスト

*8 <https://www.mathworks.com/help/slcoverage/ug/model-objects-that-receive-coverage.html>

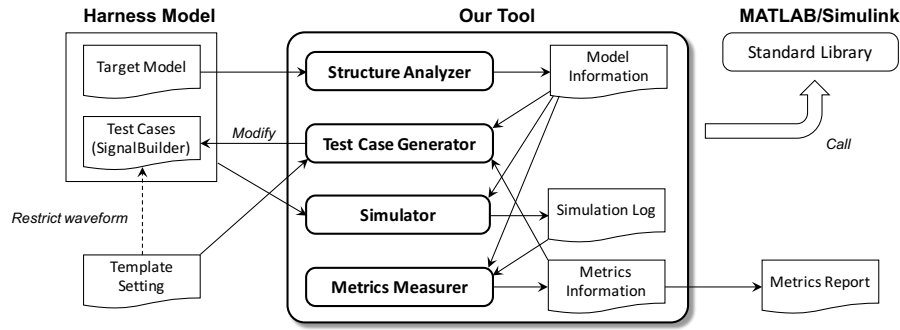


図 1 ツール概略

Fig. 1 Overview of our tool

ケースを生成する。

乱歩（ランダムウォーク）モード：与えられた生成済みテストケースのテンプレートパラメータ値に（切断）正規ノイズを付加することで，新たなテストケースとする。

3.3 シミュレータ

本ツールのシミュレータ（simulator）は，Simulink 標準のシミュレーション関数 `sim` のラッパとして機能する。

まず，構造解析器の解析結果を利用し，カバレッジ計測対象ブロックへの制御信号を発する上流ブロックを検出し，信号ログを設定する．そして，信号ロギングを有効化した上で `sim` 関数を実行し，各計測対象ブロックに流入する信号ログデータを採集する。

3.4 指標計測器

指標計測器（metrics measurer）は，対象モデルに対してテストケースを与えたときのシミュレーションログデータから各計測対象ブロックに流入する信号ログデータを抽出し，テストカバレッジや適合度を計算する。

カバレッジ計測：各計測対象ブロックに流入する信号ログデータから各テスト項目に合致/不合致する時刻数を数え上げる．また，複数のテストケースのカバレッジを比較・累積する機能ももつ。

適合度計測：テストケースに対するシミュレーションの信号ログデータから，注目する 1 テスト項目に合致する振る舞いと近さの度合，即ち，そのテスト項目に対する適合度（fitness value）を評価する [4]．どのテスト項目も比較演算と論理演算の組み合わせであるため，モデル内を流れる信号を観測することで各時刻における適合度を定めることができるため，その最大値をそのテストケースの適合度と解釈できる．ここでは，テストケースに対して，

- テスト項目に合致する場合には 1，
- そうでない場合，テスト項目から「遠い」ほど小さい値（ただし 1 未満かつ 0 より大きい）

を与える適合度関数を実装している．この実装は，[4] の方針に基づいているが，[4] では言及されていない種類のブ

ロックもサポートしている。

例 1（論理演算適合度）：テストケース（入力信号グループ） $\langle x_1, \dots, x_n \rangle$ に対する「2 入力論理演算ブロック b_1 の判定が真」という DC テスト項目に対する時刻 t の適合度は，「その上流ブロック（ b_2 と b_3 ）の判定が真」というサブ DC テスト項目に対する適合度に基づき，以下のように再帰的に与えることができる。

$$F_{DC}^{b_1, true}(x_1, \dots, x_n)(t) = \max\{F_{DC}^{b_2, true}(x_1, \dots, x_n)(t), F_{DC}^{b_3, true}(x_1, \dots, x_n)(t)\}. \quad (1)$$

例 2（比較演算適合度）：「比較演算ブロック b_4 の判定/条件 “流入する第 1 入力信号 y_1 が第 2 入力信号 y_2 より大きい” が真」という DC/CC テスト項目に対する時刻 t の適合度は，判定/条件の成否と信号差異 $|y_1(t) - y_2(t)|$ に基づき，以下の形で与えることができる。

$$F_{DC}^{b_4, true}(x_1, \dots, x_n)(t) = F_{CC}^{b_4, true}(x_1, \dots, x_n)(t) = \begin{cases} 1 & \text{if } y_1(t) > y_2(t), \\ e^{-\alpha \cdot |y_1(t) - y_2(t)| + \beta} & \text{otherwise.} \end{cases} \quad (2)$$

ただし， α と β は， b_4 に流入する信号の型及び取り得る値の範囲によって定まる係数である。

4. テストケース自動生成

本ツールは，前章各節のモジュールを適宜利用し，テストスイートの自動生成を行う．本章では，本ツールが採用したテストケース自動生成手法 [4] について簡単に述べる。

4.1 信号テンプレート

厳密には，時系列信号は任意の形状をもちうる．そのため，形式的解析に基づく SLDV 等では，その意図/意味が不明である信号や現実的でない信号をテストケースとして生成することがある．しかしながら，モデルベース開発の開発対象は基本的にフィードバック制御器であり，その入力信号は，制御された他電子的モジュールや物理法則に支

配された物理モジュール（のセンサー）から供給される。そのため、モデル/ソフトウェアの品質確認のためのテストでは、技術者がその意図/意味を理解しやすい信号に注目することが重要である。経験的には、いくつかの単純な形状をもつ信号だけを考慮した場合でも、しばしば高カバレッジを達成することができる。そうした形状は例えば定数、線形、正弦波などであり、これらの信号は少数のパラメータで特徴付けることが可能である。

本ツールでは、そうした形状毎に予め用意されたテンプレートのパラメータに引数を割り当てることでテストケースを生成する。引数の値域は、基本的には入力信号のデータ型の値域である。ただし、モデルベース開発においては、考慮すべき値域はしばしば限定されていることもある。そのため、本ツールの入力信号の探索空間は、有効なテンプレートとその各パラメータ引数の上限値・下限値で規定する。そして、各テストケースは、各信号にテンプレート及びその引数を割り当てることによって生成される。

例 3 (階段信号テンプレート): 階段信号 $\text{Step}(x, y, z)$ は、立ち上げ/立ち下げ時刻 x 、初期値 y 、最終値 z の 3 パラメータで特徴付けられる。

$$\text{Step}(x, y, z)(t) = \begin{cases} y & \text{if } t < x, \\ z & \text{otherwise.} \end{cases} \quad (3)$$

例 4 (正弦波信号テンプレート): 正弦波信号 $\text{Sine}(w, x, y, z)$ は、周波数 w 、振幅 x 、位相 y 、偏り z の 4 パラメータで特徴付けられる。

$$\text{Sine}(w, x, y, z)(t) = x \cdot \sin(w \cdot t + y) + z. \quad (4)$$

本ツールでは、現在、定数 (Constant) 階段 (Step) 線形 (Linear) 正弦波 (Sine) 矩形波 (Square) 三角波 (Triangle) の信号テンプレートをサポートしている。

4.2 標的なし自動生成法

標的なし自動生成では、標的とするテスト項目を予め決めることなく、テストケース候補を無作為に生成し、カバレッジに寄与する候補を採用することを単純に繰り返す。

- (1) 対象モデルの構造を解析し、カバレッジ計測対象を識別する (構造解析器)。
- (2) テンプレートからテストケース候補を無作為に生成する (テストケース生成器の同様無作為生成モード)。
 - (a) 各信号について、有効なテンプレートの中から同様無作為に 1 つ選出する。
 - (b) そのテンプレートの各パラメータに当てはめる引数を、その値域から同様無作為に選出する。
- (3) そのテストケースに対するモデルの振る舞いをシミュレーションする (シミュレータ)。
- (4) テストケース候補のカバレッジを計測する (カバレッジ計測器)。

- (5) テストケース候補のカバレッジを累積カバレッジと比較する (カバレッジ計測器)。

- 累積カバレッジに寄与する場合は正式にテストスイートに加える。
- 寄与しない場合はテストケース候補を破棄する。

- (6) 累積カバレッジが 100% である又は候補数の上限に達したのであればテストスイートを出力して終了する。

- そうでなければステップ 2 へ戻る。

なお、各入力信号の有効範囲、有効なテンプレート、及びそのパラメータ値の上限及び下限を利用者が絞り込むことが可能で、より探索の効率的を図ることができる。

4.3 標的あり自動生成法

標的あり自動生成では、標的とするテスト項目を予め決め、これまでに生成したテストケースの適合度を基に、より高い適合度をもつテストケースの生成を試みる。標的なし自動生成ではカバーし難いテスト項目を効率的に探索するために利用することを想定した、補助的な機能である。

本ツールでは、以下の単純な乱歩法を採用している。

- (1) 対象モデルの構造を解析し、カバレッジ計測対象を識別する (構造解析器)。
 - (2) テンプレートからテストケース候補を無作為に生成し (テストケース生成器の同様無作為生成モード)、現候補とする。
 - (3) 現候補に対するモデルの振る舞いをシミュレーションする (シミュレータ)。
 - (4) 現候補のテスト項目に対する適合度を計測する (適合度計測器)。
 - 適合度が 1 である場合、現候補を出力し終了する。
 - (5) 現候補に正規ノイズを付加し、新候補とする。 (テストケース生成器の乱歩モード)。
 - (6) 新候補に対するモデルの振る舞いをシミュレーションする (シミュレータ)。
 - (7) 新候補のテスト項目に対する適合度を計測する (適合度計測器)。
 - 適合度が 1 である場合、現候補を出力し終了する。
 - (8) 現候補と新候補の適合度を比較して、必要であれば現候補を更新し、ステップ 5 へ戻る。
 - 新候補の適合度 f_n が現候補の適合度 f_c より大きい場合、新候補を現候補とする。
 - そうでない場合、確率 f_n/f_c で新候補を現候補とする (確率 $1 - f_n/f_c$ で現候補を維持する)。
- 標的あり自動生成法では、Markov-Chain Monte-Carlo (MCMC) 法と同様の手順をとっている。MCMC 法本来の用途は特定の分布に従う無作為系列の生成であり、本自動生成法の目的である適合度の最大化とは異なる。しかし、本自動生成法では、現候補と新候補の適合度比に基づいて候補を確率に更新していくことで、(i) 適合度の高い

候補が残り易く、また、(ii) 候補系列が局所最適に陥ることをある程度抑制できるという特徴を得ることができ、適合度が高い候補を貪欲に選択する場合よりも安定的な適合候補探索が可能である [4].

なお、標的なし自動生成法と同様に、探索範囲を利用者が絞り込むことが可能で、過去のテストケース（のテンプレート及びそのパラメータ）と適合度の組みを参考に、利用技術者の知識や技術、経験に基づいて有望な範囲に絞り込むことができる。

5. カバレッジ計測の高速化

本章では、既存カバレッジ計測手法（SLC）の概略を説明し、その後、それと対比して本テスト自動生成ツールのカバレッジ計測手法の特徴を述べる。ただし、SLC の計測手法の詳細は公開されていないため、SLC の動作を観察して推論した特徴に基づいて議論を行う。

5.1 既存カバレッジ計測手法の特徴

SLC が提供するカバレッジ計測関数 `cvsim` は、Normal モードでしかシミュレーションできない。ただし、シミュレーションを一時停止した場合、それまでのカバレッジ計測の途中結果を即座に表示できる。以上のことから、`cvsim` 関数では、以下を繰り返していると考えられる。

- (1) シミュレーションを 1 ステップ進める。
- (2) 条件及び判定が真/偽となった時刻点の数を加算する。即ち、`cvsim` 関数では、MATLAB がもつ高速な配列/行列演算命令群は利用されていないと考えられる。そのため、一時停止をしない利用の場合には効率的ではない。

5.2 本ツールのカバレッジ計測手法の特徴

一方、本ツールでは、以下の計測アプローチをとる。

- (1) モデルの構造解析を行い、各計測対象ブロックに制御信号を供給する信号のロギングを有効化する。
- (2) 終了時間まで無停止でシミュレーションを行う。
- (3) その信号ログを基に、条件及び判定が真/偽となった時刻点の数を数え上げる。

効率的にカバレッジを計測するだけであれば、シミュレーションが一時停止可能である必要はない。そこで、無停止とする代わりに Accelerator/Rapid モードを利用可能とした。また、時刻点数の数え上げでは、配列/行列演算命令群を用いて高速化を図る。なお、適合度計測においても同様のアプローチを採用している。

6. 性能評価

本章では、本ツールのテストケース生成性能及びカバレッジ計測性能を評価するため、商用開発で実際に作成された実用的モデルに対して行なった実験の結果を報告する。ここで、現実的な規模と複雑さをもつモデルでは、標的

なし自動生成法でカバーできないテスト項目に合致するテストケースを探索範囲の絞り込むことなしに標的あり自動生成法で発見することは非常に困難である。探索範囲の絞り込みによる探索効率化は利用技術者の技量に依存し、現在のところ公正公平な評価実験を実施できていないため、本稿では標的なし自動生成法に関してのみ評価する。

なお、4 章で紹介した手法の評価については [4] で実施されている。本稿の性能評価は、5 章のカバレッジ計測高速化の実装も含めた、ツールとしての性能評価である。

6.1 実験環境

実験環境は以下の通りである。

- MacBook Pro (15-inch, 2017)
 - ー プロセッサ: 2.8 GHz Intel Core i7
 - ー メモリ: 16GB 2133 MHz LPDDR3
- OS: macOS High Sierra (10.13.4)
- MATLAB/Simulink/SLDV/SLC R2017b

また、タイムアウト時間を 600 秒として実験を行なった。

6.2 対象モデル

本評価実験では、中規模制御器ハーネスモデル M_1 と大規模制御器ハーネスモデル M_2 を対象として用いた (表 1)。なお、どちらのモデルも商用開発で実際に作成された実用的なものであり、詳細は非公開できない。

6.3 テストケース生成性能の評価

カバレッジ計測性能を評価するため、6.1 節の実験環境上で 6.2 節の 2 つの実用モデルに対して、以下の 2 ツールを用いてフルカバレッジテストスイート生成を試みた。

- 本ツール
 - ー 試行回数: 20 回
 - ー シミュレーションモード: Accelerator (Accel)
 - ー 高速リスタート: 有効
 - ー テンプレート設定
 - * 中規模モデル M_1 :
 - ・ `int16` 型の入力: Step テンプレート、値の上下限は仕様から既知のものを設定。立上/立下時刻の下限は 0、上限はシミュレーション時間 (StopTime)。
 - ・ `double` 型の入力: Constant テンプレート, Linear テンプレート, Sine テンプレートのいずれかをランダム。値の上下限は設定なし (データ型の上下限)。Sine テンプレートの位相及び偏りは 0 で固定。
 - * 大規模モデル M_2
 - ・ 各入力: Step テンプレート又は Linear テンプレートのいずれかをランダム。値の上下限は設定なし (データ型の上下限)。

表 1 テスト対象モデルの概略
Table 1 Summary of experimental models

Attributes	M_1	M_2
#Inputs	2	51
#uint8-typed signals	0	7
#int16-typed signals	1	3
#uint16-typed signals	0	1
#single-typed signals	0	40
#double-typed signals	1	0
#Blocks	476	2374
#RelationalOperator blocks	16	120
#Logic blocks	7	7
#Switch blocks	1	152
#MultiPortSwitch blocks	33	42
#Abs blocks	0	1
#MinMax blocks	0	1
#If blocks	0	11
#Subsystems	29	145
#Enable subsystems	2	4
#Trigger subsystems	5	0
#IfAction subsystems	0	24
#Objectives	198	710
#DC objectives	121	428
#CC objectives	64	268
#MC/DC objectives	13	14
StopTime	3.0	7.0
SampleTime (base)	1/20000	1/1000

● SLDV

ー 試行回数：1 回（乱択アルゴリズムでないため）

その結果を表 2 に示す。なお、本ツールの実行時間には構造解析器の実行時間も含む。また、中規模モデル M_1 のテスト項目が SLDV の方が 13 個多いのは、MC/DC テスト項目を重複してカウントしているためである。そして、大規模モデル M_2 のテスト項目が SLDV の方がかなり少ないのは、SLDV がテスト項目を数え漏らしている又は扱えないブロックを無視しているためである。これは、SLDV が示すテスト項目数が比較演算ブロック、スイッチ類の個数の和の 2 倍に満たないことから明らかである。

本評価実験では、既存ツールではおよそ 7 割のカバレッジしか達成できなかった中規模モデルに対しても、本ツールは安定的かつ高速にフルカバレッジを達成できた。また、大規模モデルに関しては、本ツールはフルカバレッジを達成できなかったものの、平均して SLDV よりおよそ 1 割高いカバレッジを達成した。なお、SLDV が数えていないテスト項目を考慮すれば、平均しておよそ 2 割高いカバレッジである。

6.4 カバレッジ計測性能の評価

カバレッジ計測性能を評価するため、6.1 節の実験環境上で 6.2 節の 2 つの実用モデルに対して、本ツールのテ

スケース生成器の一律無作為抽出モードで作成した 20 個のテストケースのカバレッジを以下の 2 ツールの有効化可能なシミュレーションモードを用いて計測した。

- 本ツール（カバレッジ計測器）
- SLC (cvsim 関数)

各シミュレーションモード毎に、20 個のテストケースに対するカバレッジ計測に要した平均時間の比較を表 3 に示す。なお、Normal/Accelerator モードにおいては、高速リスタート機能を有効化しており、初回コンパイル時間は実行時間に含めていない。Rapid モードにおいては、高速リスタート機能を有効化できずシミュレーション毎にスタンドアロン実行ファイル生成が必須なため、その処理時間も実行時間に含めている。

本評価実験では、本ツールは既存ツールと比較して 5～27 倍高速にカバレッジを計測できた。中規模モデル M_1 では高速シミュレーションモードの方が通常シミュレーションモードよりむしろ低速であったが、これはシミュレーション毎に必要な前処理（スタンドアロン実行ファイル生成）や MATLAB 環境外との API 通信のオーバーヘッドの比重が大きかったためと考えられる。

7. 関連ツール

Simulink モデルに基づいた電子制御部品のモデルベース開発においては、MathWorks 社が提供する純正ツール SLDV/SLC の他、いくつかの第三者ツールが提供されている。

MathWorks 社の SLDV は、主に形式手法に基づいた解析によりテストケース生成を試みる。そのため、現実的な規模・複雑性をもつモデルを扱うことができない。また、T-VEC 社の T-VEC Tester^{*9} for Simulink/Stateflow^{*10} (sl2tvec) [5], [6] も、形式手法ベースでテストケース自動生成を試みる第三者ツールである。一方、本ツールでは、Monte-Carlo 法によって高カバレッジテストスイート生成を試みる。単純なモデルでも高カバレッジテストスイートを得ることができない可能性は 0 ではないが、MATLAB/Simulink が許容するどのような規模・複雑性をもつモデルも扱うことが可能である。

Reactive Systems 社^{*11}の Reactis は、シミュレータ、テスト、検証器からなる第三者ツールである。Reactis のテストは、Monte-Carlo 法とガイド付きシミュレーション (guided simulation) という発見的解決技術を組み合わせ、未達成テスト項目に合致するテストケース生成を試みる [1]。ガイド付きシミュレーションでは、前のテストケースのシミュレーション結果に対する後向きデータフロー解析を基にある時刻以降の入力信号を変化させることで未達成テ

^{*9} <https://www.t-vec.com/solutions/simulink.php>

^{*10} <https://mathworks.com/products/stateflow.html>

^{*11} <http://www.reactive-systems.com/>

表 2 テストケース生成性能の比較

Table 2 Comparison on performance for test suite generation

Model	Tool		Coverage	TestSuite	#Candidates	Processing Time
M_1	Our tool	Best	198/198	5	9	29.9 sec
		Average	198/198	6.6	69.5	209.3 sec
		Worst	198/198	10	151	445.5 sec
	SLDV		144/211	8	-	>600 sec
M_2	Our tool	Best	395/710	6	16	>600 sec
		Average	367.9/710	8.5	16	>600 sec
		Worst	335/710	11	16	>600 sec
	SLDV		232/549	8	-	>600 sec

表 3 カバレッジ計測性能の比較

Table 3 Comparison on performance for coverage measuring

Model	Tool	Simulation mode		
		Normal	Accel	Rapid
M_1	Our tool	1.8 sec	2.9 sec	10.2 sec
	SLC	50.6 sec	-	-
M_2	Our tool	54.6 sec	38.5 sec	30.0 sec
	SLC	258.4 sec	-	-

スト項目をカバーする新テストケース候補を生成する。本ツールとは、形式手法に頼らずランダムにテストケース候補を生成する点では同じだが、その生成アプローチが大きく異なる。Reactis は前候補の時系列データの後半部分を変化させることで新候補を生成するが、本ツールでは信号全体の形状を決めるテンプレートに流し込むパラメータ値を変化させることで新候補を生成する。

Simulink モデルに対する Monte-Carlo テストケース生成手法としては、乱択有向テスト (randomized directed testing; REDIRECT) [2] や出力多様性 (output diversity) に基づいた乱択生成法 [3] など提案されている。REDIRECT 法は、Stateflow によって記述された状態遷移系を含む Simulink モデルを対象として、乱択テストと有向テストと逆戻り (バックトラック) の技術の組み合わせによって状態網羅・遷移網羅となるテストスイートの生成を試みる。テストケースの生成アプローチとしては Reactis と同様に徐々に時系列データを伸長しており、非線形演算に対応するためにパターンに基づいた発見的手法ライブラリを用いる制約ソルバーによって伸長部分の時系列データを生成する。即ち、取り扱う網羅基準とテストケース生成アプローチが本ツールとは異なる。[3] の手法は、進化的アルゴリズムによって出力信号群が高い多様性をもつようにテストスイート (テストケース群) を生成するもので、そのようなテストスイートを利用することで高品質な機能テスト (ブラックボックステスト) が可能としている。一方、本ツールでは制御テスト (ホワイトボックステスト) に注目し、高い判定、条件、MC/DC カバレッジをもつテストスイートを生成する。なお、標的あり自動生成法に進化的/遺

伝的アルゴリズムを適用することも可能である。ただし、[3] の手法が全体として多様になるようにテストスイート群を生成しようとするのに対して、標的あり自動生成法では標的テスト項目に合致する 1 つのテストケースを生成しようとしている。

8. まとめ

本稿では、実用的な Simulink モデルを対象としたテスト自動生成ツールを紹介した。本ツールは、既存ツールよりはるかに高速な判定・条件・MC/DC カバレッジ計測が可能である。また、Monte-Carlo 法により、既存ツールでは扱うことができない規模の実際のモデルに対しても高カバレッジテストスイートを自動生成できる。さらに、テンプレートを利用することにより、各テストケースを構成する入力信号の波形は、技術者がその意図/意味を理解しやすい信号に限定しており、モデルの改良・洗練に活かしやすいテストの実施を支援することができる。

今後は、ツールを実際の開発現場で利用してもらい、そのフィードバックを基にツールを洗練する予定である。

参考文献

- [1] Sims, S. and DuVarney, D. C.: Experience Report: The Reactis Validation Tool, *Proc. of the 12th ACM SIGPLAN Int. Conf. on Functional Programming*, pp. 137–140 (2007).
- [2] Satpathy, M., et al.: Randomized Directed Testing (REDIRECT) for Simulink/Stateflow Models, *Proc. of the 8th ACM Int. Conf. on Embedded Software*, pp. 217–226 (2008).
- [3] Matinnejad, R., et al.: Automated Test Suite Generation for Time-continuous Simulink Models, *Proc. of the 38th Int. Conf. on Software Engineering*, pp. 595–606 (2016).
- [4] Tomita, T., et al.: Template-Based Monte-Carlo Test Generation for Simulink Models, *7th Workshop on Design, Modeling and Evaluation of Cyber Physical Systems* (2017). (To be published)
- [5] Blackburn, M. R. and Busser, R. D.: T-VEC: a tool for developing critical systems, *Proc. of the 11th Annu. Conf. on Computer Assurance*, pp. 237–249 (1996).
- [6] Blackburn, M., et al.: Automatic generation of test vectors for SCR-style specifications, *Proc. of the 12th Annu. Conf. on Computer Assurance*, pp. 54–67 (1997).