OpenModelicaサポートサービスについて

松澤　邦裕*　　佐藤　甫*　　小池　晋太朗*　　山下　貴志*

OpenModelica Support Services

Kunihiro Matsuzawa*, Hajime Sato*, Shintaro Koike*, and Takashi Yamashita*

OpenModelica は、1DCAE やモデルベース開発に適した Modelica 言語による、オープンソースのモデリング・シミュレーション環境である。OpenModelica のようなオープンソース・ソフトウェア（OSS）は、ソースコードが公開されており、無償で商用利用が可能なため、コストダウンが期待できる一方で、サポートがないことで業務が進まなくなる懸念があり、導入が避けられるという傾向がある。アドバンスソフトでは、この課題を解決するために、秘密保持契約を結んだ上で、お客様のサポート依頼に対して具体的な回答を行う「OpenModelica サポートサービス」を事業化している。

本稿では、「OpenModelica サポートサービス」の事業内容として、OpenModelica を利用したモデルの開発や改良、他ツールからのモデル移植、FMU 化やモデル連成で発生する課題やトラブルなど、モデルベース開発全般に対する技術的サポートについて紹介する。

Keywords: OSS, MBD, モデル流通, FMU/FMI, OSS, OpenModelica, 機械学習、ベイズ最適化、強化学習

1. はじめに

近年、国内外を問わず研究開発に利用するソフトウェアの価格は上昇傾向であり、研究開発予算に対して大きなインパクトを与え始めている。解決策の一つとして、オープンソース・ソフトウェアに置き換えることを検討している企業も多い。無償で商用利用が可能なため、コストダウンが期待できる一方で、サポートがないことで業務が進まなくなる懸念があり、導入が避けられるという傾向がある。例えば、モデルベース開発ツールでは、商用のものは充実した機能や高度なモデルが多数含まれているため、大変高価である。オープンソース・ソフトウェアの OpenModelica は、無償で利用できるが、同梱されたモデルはシンプルなものであり、パラメータ同定や最適化などの機能は含まれていない。そのような現状を踏まえ、OpenModelica を利用した実用的なモデル開発、他のソフトウェアと組み合わせることで、パラメータ同定や最適化など新しい機能を実現する使用法を紹介する。

*アドバンスソフト株式会社　第 5 事業部

5th Computational Science and Engineering Group, AdvanceSoft Corporation

2. OpenModelica サポートサービスの概要

OpenModelica サポートサービスは、OpenModelica の使い方だけではなく、他のツールとの連携や関連技術に対してもサポートの対象としている。

2.1. サポートサービスの対象例

サポート対象となる内容については、以下のような項目を挙げる。

- OpenModelica の使用法に関する説明
- Modelica 言語の文法に関する説明
- 計算実行時のエラーなどの問題改善に対する技術サポート
- モデル作成や計算結果の振る舞いに対する技術サポート

2.2. お問い合わせ

- お問合せの方法
  - E-mail による問合せに対応。
- お問合せの受付時間

- E-mail でのお問い合わせは 24 時間受け付け、回答は業務時間内に対応。

- 業務時間
  - 平日（月曜日～金曜日）の 9:00～17:00
  - ※ 土曜日、日曜日、国民の祝日、国民の休日、振替休日、年末年始（12 月 29 日～1 月 3 日）、および事前に通知する休業日を除く。
- 回答までの期間
  - 翌営業日までに回答にかかる期間を通知する。
  - 回答までに必要な期間はお問合せ内容によって変動する。

3. OpenModelica について

3.1. Modelica 言語

Modelica[1]言語は、数式に支配された構成要素の非因果的な接続をサポートし、第一原理からのモデリングを容易にするツールである。モデルの再利用を容易にするオブジェクト指向の構造を提供し、機械、電気、電子、磁気、油圧、熱、制御、電力、プロセス指向のサブコンポーネントなどを含む複雑なシステムのモデリングに便利に使用することができる。

言語仕様やメンテナンスは非営利国際組織の Modelica 協会によって行われており、無償で公開されている。

3.2. OpenModelica

OpenModelica[2]は、複雑な動的システムのモデリング、シミュレーション、最適化、分析を行うための Modelica モデリング言語に基づいた、無料のオープンソース・ソフトウェア環境である（図 1)。このソフトウェアは、非営利、非政府組織である Open Source Modelica Consortium （OSMC）によって開発されている。 OSMC は、スウェーデンのリンショーピング大学と共同で RISE SICS East AB のプロジェクトとして運営されている。 OpenModelica は学術及び産業界で幅広く利用されている。

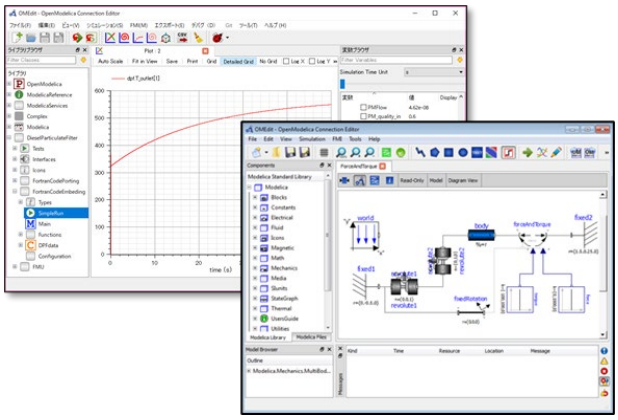

図 1　OpenModelica 画面

4. 外部制御インターフェイス・ライブラリ

4.1. OMPython

OMPython[3] は、 Python 言語で書かれた OpenModelica を外部制御するためのインターフェイス・ライブラリである。
OpenModelica 標準ライブラリに基づいており、モデリング、コンパイル、およびシミュレーション環境に対し、純正の拡張機能として、外部制御インターフェイスを提供している。

4.2. OpenModelicaCompilerForPython

アドバンスソフトでは、Modelica および OpenModelica を用いた設計・開発の高度化を目的として、OpenModelica をより安定的かつ自由度の高い制御を可能とするための Pyhton インターフェイス・ライブラリ OpenModelicaCompiler ForPython[4]を開発した。

開発した OpenModelicaCompilerForPython を利用することで、OpenModelica プロセスに対し、インターフェイス・ライブラリを通じたプロセスの呼び出し、シミュレーションの外部制御やシミュレーション結果の外部保存、モデルパラメータへの読み書きなどだけではなく、Python 環境で使用可能な多岐にわたるパッケージ機能を可能にする。

5. 事例

5.1. OpenModelica モデル

本事例で使用した OpenModelica のモデル定義図を図 2 に示す。本モデルは、シンプルなバネ・

マス・ダンパーモデルであり、外力として正弦波を与えている。このバネ・マス・ダンパーモデルを対象として、ベイズ最適化および強化学習を用いたモデル制御を行った。

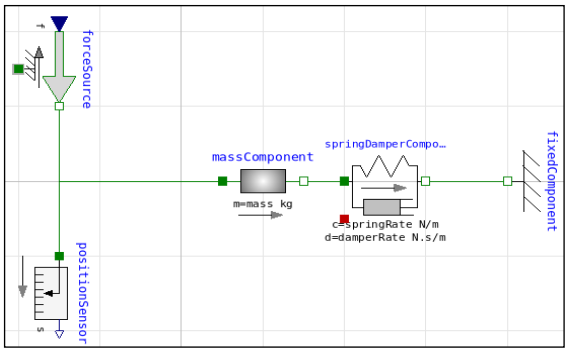

図 2　シンプルなバネ・マス・ダンパーモデル

5.2. 事例①：ベイズ最適化

本事例では、OpenModelicaCompilerForPython を介してオープンソースのベイズ最適化ライブラリである BayesianOptimization[5] を用いて OpenModeica モデルを外部制御し、ベイズ最適化によってモデルパラメータを探索する検証を行った。事例①で与えたパラメータの一覧を表 1 に示す。

本事例の目的はシミュレーション条件と一致するバネ定数とダンパー定数の決定であるが、本事例のために実測データを模擬したダミー実測データを作成した。ダミー実測データは、バネ定数およびダンパー定数のそれぞれの正解値を 2.0 [N]として、外力の振幅にランダムノイズを加えたシミュレーションを実行することで、ランダムノイズによって異なる “マスの位置の時間変化” の波形データを 100 個生成した。ベイズ最適化による探索の各ステップでは、ベイズ最適化によって提案されたバネ定数およびダンパー定数を与えたシミュレーションを実行し、その際の “マスの位置の時間変化” の波形データを取得し、ダミー実測データ（波形×100）との残差二乗和を目的関数とすることで、その値が最小となるバネ定数およびダンパー定数を求めた。また、獲得関数には信頼性上限関数 UCB（Upper Confidence Bound）を用いた。

表 1　Parameter list of Bayesian optimization

parameters	values	unit	target
amplitude of external force	1.0 (sine wave)	[N]	
Cycle of external force	1.0 + noise	[Hz]	
Mass of the object	1.0 + noise	[kg]	
Spring rate	0.001 - 9.999	[N/m]	✓
Dunbar rate	0.001 - 9.999	[N/m]	✓

表 2　Parameter list for reinforcement learning

parameters	values	unit	target
amplitude of external force	1.0 (sine wave) + noise	[N]	
Cycle of external force	1.0	[Hz]	
Mass of the object	1.0	[kg]	
Spring rate	2.0	[N/m]	✓
Dunbar rate	2.0	[N/m]	✓

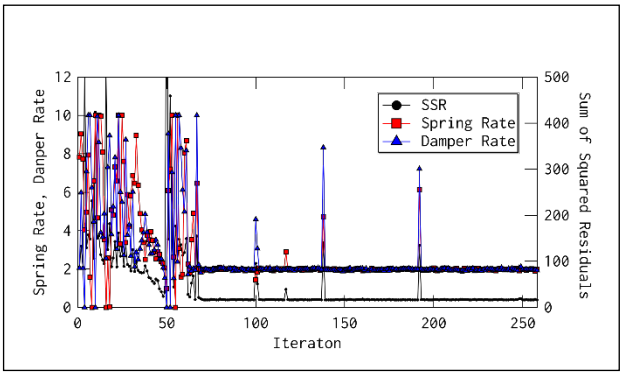

図 3　Result of Bayesian optimization

ベイズ最適化による探索結果を図 3 に示す。探索回数の増加に伴って探索範囲が正解値である 2.0 の付近に集中していき、探索回数が約 70 回程度以降ではバネ定数とダンパー定数ともに正解値に近い付近の微小な差異を探索している様子が見られた。探索回数 258 回までに、バネ定数 1.98593（正解値 2.0）[N/m]，ダンパー定数 1.97402（正解値 2.0）[N/m]を得た。

5.3. 事例②：強化学習

本事例では、オープンソースの強化学習フレームワークである OpenAI Baselines[6]および環境定義 OpenAI Gym[7]を利用して、Python プログラム

から OpenModelica モデルを外部制御する検証を行った。事例②で与えたパラメータの一覧を表 2 に示す。

本事例の目的は、図 2 で説明したバネ・マス・ダンパーモデルに対し、バネ定数とダンパー定数を未知の変数と仮定して、特定のシミュレーション結果を再現するバネ定数とダンパー定数を探索することである。ここで言う探索とは、熟練した技術者が複数回の試行によってモデルパラメータを調整するのと同様の行為であり、機械学習アルゴリズムのひとつである強化学習を用いて代替することで自動調整が可能かどうか検証した。

強化学習モデルの訓練のため、バネ・マス・ダンパーモデルを強化学習の「環境」、バネ定数とダンパー定数を強化学習の「打ち手」として、初期のバネ定数とダンパー定数をランダムに与えてシミュレーションを実行する。このとき、測定誤差の存在を考慮して、バネ・マス・ダンパーモデルには外力の強度と質量にノイズ（ばらつき）を加える。シミュレーションの実行結果のうち、“マスの位置の時間変化” の波形を強化学習の「観測」として取得し、その結果から強化学習の「報酬」を求める。報酬は、図 4 の報酬の獲得条件をクリアしているかを確認し、クリアしている場合は報酬が計算によって与えられ、クリアしていない場合はペナルティとなる低い報酬が与えられる。また、報酬の計算では、バネ定数とダンパー定数が大きいと報酬の獲得条件がクリアしやすいことが予想されるため、どちらの定数もより小さい値で探索条件を満たす方が、報酬が大きくなるように調整している。強化学習モデルは、初期のシミュレーション結果から得られる「観測」から、次のバネ定数とダンパー定数の値を「打ち手」として考え、それらの値を次回のシミュレーション条件として設定してシミュレーションを実行し、その結果を「観測」する。上述のプロセスを繰り返すことで、より少ない試行回数で様々な初期条件から適切なバネ定数とダンパー定数を決定するように強化学習モデルを訓練した。訓練の際には、1 回あたりの学習でこの試行を最大 10 回を限度として繰り返し行うこととした。報酬の獲得条件に合致し、報酬が得られたタイミングで終了となる。よって、試行回数が少なければ少ないほど、報酬が高くなるようになる。報酬の最大は 1.0、最小は-1.0 である。学習を 100 ステップ行うごとに、10 ケースの検証を行った。これを 150 回繰り返した。

強化学習による結果を図 5 に示す。左図では、学習回数が 3000 回くらいまでは、試行回数も多く、獲得報酬や到達回数も少ないが、学習回数が進むに連れ、試行回数が 1 回か 2 回に収まるようになり、報酬も多く得られるようになることが示された。また、右図では、学習が 1000 回、5000 回、15000 回のときの検証で選ばれたバネ定数とダンパー定数の分布を示している。学習回数の少ない方は、失敗の割合が高いため、試行回数が多くなっている分、プロットの点が多い。また、学習が多くなるに連れて、正解に近い値を早くから得られるようになるため、分布の広がりが狭くなっていることがわかる。

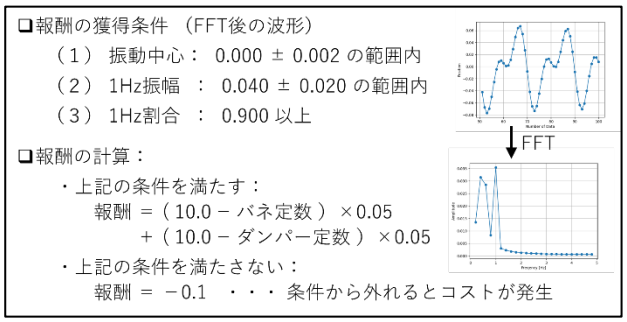

図 4 Conditions for earning rewards and Reward calculation

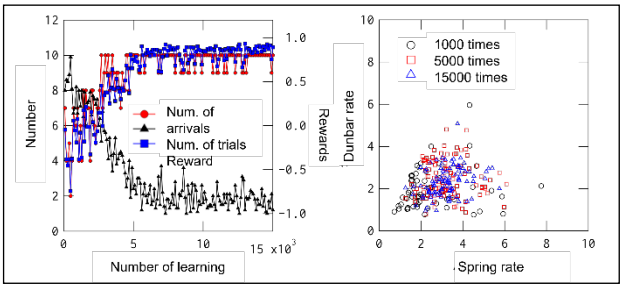

図 5 Result of reinforcement learning

6. 単体シミュレータの FMU 化

実用的な MBD モデルを新しく開発することは難しく、完成までに長い期間や多大な人的リソー

スが必要なことが予想される。一方で、長年開発し、活用してきた単体のシミュレーションソフトが MBD モデルへ移植できれば、「モデル流通」への取り組みの後押しとなるため、C/C++言語および Fortran 言語で開発した単体シミュレータプログラムを FMI 規格に準拠した fmu ファイルへビルドするための C++ライブラリである fmi2++を開発した。 この fmi2++に従って既存のプログラム資産を fmu ファイルへ移植し、ビルドすることで、FMI 規格に従った MBD ツール上で利用することが可能となる。

汎用プログラミング言語で記述された単体シミュレータプログラムを fmu ファイルへとビルドする方法として、 MBD ツールで動作する中間モデルを経由する間接的な方法と、中間モデルを経由しない直接的な方法の二通りを考えることができる。 間接的な方法では、 汎用プログラミング言語のプログラムをモデルとして取り込む機能と、 モデルを fmu ファイルへとエクスポートする機能の両方を備えた MBD ツールを用いるが、MBD ツールによってはこれらの二つの機能の間の相性が問題になることがある。相性が悪い場合、各変換に一見問題がないのにも関わらず、fmu ファイルの動作が単体シミュレーションプログラムの動作と異なったり、そもそも動作しなかったりという事例があった。 この場合、fmu ファイルを動作させるために中間モデルや MBD ツールの動作を検証する必要があり、本来の目的から遠い技術的な課題を解決しなければならない。一方、直接的な方法では MBD ツールの相性に左右されないメリットがあるが、MBD ツールを利用する場合とくらべて手間がかかる。

MBD ツールの相性問題がないメリットがある直接的な手法を、より手軽に利用できるようにするため、 fmu ファイルを生成するための fmi2++ライブラリを実装することとした。

fmi2++というライブラリ名は、C/C++プログラムを FMI 2.0 規格に従った fmu ファイルへと直接ビルドすることを意図している。 FMI 規格を読み解き、fmu ファイルとして動作するために必要な要件を抽出し、再利用性が高くなるように実装した。

7. まとめ

オープンソース・ソフトウェアの活用は、「モデル流通」の発展的な取り組みへの後押しとなると考えている[8][9]。OpenModelica は、無償で利用可能なオープンソースの MBD ツールであり、またシンプルな 1DCAE や MBD に適した Modelica 言語のモデリングおよびシミュレーション環境であるが、商用 MBD ツールのような高度な機能は備えていない。そこでアドバンスソフトは、OpenModelica を外部制御できる Python インターフェイス・ライブラリ OpenModelicaCompiler ForPython を新規に開発し、他のオープンソース・ソフトウェアと組み合わせることによる高度な機能拡張の実現可能性についての検証を行った。また、その事例として、OpenModelica モデルに対してベイズ最適化と強化学習による外部制御の 2 つの事例を紹介した。

我々が開発した Python 外部制御インターフェイス OpenModelicaCompilerForPython を利用することで、Python 環境で使用できる豊富なパッケージ群を OpenModelica と組み合わせることが可能となる。これは、高度な機能拡張が獲得できる汎用性の高いオープンソースによる MBD の枠組みであり、設計・解析の効率化など、今後の活用が期待される。

C/C++および Fortran などの汎用プログラミング言語における、単体シミュレーションプログラムを fmu ファイルへと変換するための方法として、MBD ツールを用いて変換した中間モデルを経由した場合の課題について議論した。この課題を回避するために、単体シミュレーションプログラムを直接 fmu ファイルへとビルドする方法が求められる。

単体シミュレーションプログラムを直接 Model Exchange の FMU へと変換するために FMI 2.0.3 規格を調査した。fmu ファイルは規格に定められたディレクトリ構成とファイル内容を ZIP 形式で保存したものである。fmu ファイルの構成要素のうち必須なものは、モデル記述ファイルとモデル

実装ライブラリの二点である。それぞれの構成要素の仕様について規格書の内容に基づいて網羅し、規格に準拠した fmi2++ライブラリを実装した。fmi2++は C++で記述されており、プラットフォームやコンパイラに依存しないため、様々なビルド環境で FMU を作成することができる。fmi2++ライブラリには規格で定められたモデル記述ファイルおよびモデル実装ライブラリのテンプレートが含まれているため、はじめから FMI 規格に準拠した空の fmu ファイルをビルドすることができる。

OpenModelica サポートサービスに関するお問い合わせを心よりお待ちしております。

参考文献

[1] https://modelica.org/
[2] https://www.openmodelica.org/
[3] https://github.com/OpenModelica/OMPython
[4] https://github.com/ijknabla/OpenModelicaCompilerForPython
[5] https://github.com/fmfn/BayesianOptimization
[6] https://github.com/openai/baselines
[7] https://gym.openai.com/
[8] 加藤廣ら: MBD を起点としたモデル流通事業への展開，アドバンスシミュレーション，第 28 号，P141 177 (2020)
[9] 松澤邦裕ら: AI 活用によるモデルベース開発への取り組み，アドバンスシミュレーション，第 28 号，P60 63 (2020)

※ 技術情報誌アドバンスシミュレーションは、アドバンスソフト株式会社 ホームページのシミュレーション図書館から、PDF ファイル（カラー版）がダウンロードできます。（ダウンロードしていただくには、アドバンス/シミュレーションフォーラム会員登録が必要です。）