Vol.2016-SE-192 No.16
Vol.2016-EMB-41 No.16
2016/6/3

Simulinkモデルから状態方程式・出力方程式の抽出

池田 良裕[1,a)]　枝廣 正人[1,b)]

概要：車載制御向けの設計において、モデルベース開発を用いることが多くなってきている。一方で制御システム・プラントシステム共に複雑化が著しく、モデルのテスト・修正を繰り返していくうちにシステムの現状把握が難しいという状況が起きている。そこで本研究では、モデルベース開発ソフトであるMATLAB/Simulink を対象として、モデルから状態方程式・出力方程式の抽出手法について提案した。実際に提案手法を用いてフィボナッチ数を出力するモデルから状態方程式・出力方程式を抽出できることを確認した。

1. はじめに

車載制御向けの設計において，モデルベース開発が採用されることが増えてきている．モデルベース開発ではMATLAB/Simulink[1] が広く用いられており，制御システムをモデルとして作成することで視認性を向上させ，システムの大規模化，複雑化に対しても簡潔かつ正確に表現することが可能である．また，実際のハードウェアをモデル化することにより実機を用いることなくシミュレーションを行うことができ，制御システム開発における手戻りによるロスを減らすことにも貢献している．

モデルベース開発において制御システム設計する際に，制御対象の仕様が変更されることがある．制御設計者はモデルを変更することで仕様変更に対応していくが，モデルの変更が繰り返されていくと制御対象システムのモデルは存在しても数式によるシステムの特性を把握することができない．システムの特性を把握する上でモデルからの数式化は必須である．

2. 状態方程式・出力方程式の抽出

2.1 各方程式該当部の探索

Simulink モデル内の状態方程式・出力方程式の該当部の探索には BLXML(Block-Level XML)?を用いた．MATLAB/Simulink の API を用いることでブロック接続関係をたどることも可能ではあるが, 本研究では,Simulink モデル BLXML 及び BLXML パーサである SimulinkXSD を使用し，BLXML からブロックの接続関係を取得，及び状態方程式・出力方程式の探索を行った．

現代制御理論では，システムは内部状態を持つものと捉え，それを状態方程式で記述する．時刻 n における状態方程式には，時刻 n − 1 やそれ以上前の時刻における状態変数が必要になる．よって，状態方程式の探索では，まずSimulink モデル内の前状態を保持しているブロックを探した．MATLAB/Simulink において前状態を保持できるブロックは遅延ブロック，積分ブロックが挙げられる．中でも，Unit Delay(遅延) ブロック，Integrator(積分) ブロックはモデルベース開発での制御の分野において，広く使われているため，状態方程式の探索での基準とした．以降，これらを状態方程式の**探索開始ブロック**と呼ぶ．

状態方程式の探索は，探索開始ブロックから始め，**探索終了ブロック**まで Simulink モデルを遡りながら状態方程式該当ブロックを追加していった．探索終了ブロックとは Unit Delay ブロック，Integrator ブロックのような内部状態を保持しているブロック，または Inport(入力) ブロックのようなこれ以上遡ることが出来ないブロックである．

出力方程式はシステムの出力を式として表したものである．出力方程式の探索開始ブロックは Outport(出力) ブロックであり，探索終了ブロックは状態方程式と同様である．出力方程式の探索の流れも状態方程式と同様に探索開始ブロックから始め，探索終了ブロックまで Simulink モデルを遡りながら出力方程式該当ブロックに追加していった．

また両方程式とも複数の探索開始ブロックが存在する可能性があるため，各探索開始ブロックに対してそれぞれ該当ブロックのリストを作成した．

2.2 Maple/BlockImporter を用いた数式化

Maple/BlockImporter は Simulink モデルを入力として，

[1] 名古屋大学 大学院情報科学研究科
〒 464-8603 愛知県名古屋市千種区不老町
[a)] yoshi12@ertl.jp
[b)] eda@ertl.jp

数式に変換するツールである．このツールでは入力の Simulink モデルは連続モデルでなければならず，一つの Simulink モデルに対して一つの数式しか出力できない．そのため，2.1 章で探索した各方程式該当ブロックを元のモデルから抜き出した．元のモデルから該当ブロックを抜き出すと探索開始・探索終了ブロックに含まれる Unit Delay ブロック，Integrator ブロックの入出力どちらかの信号線が接続されていない状況となり，数式への変換が行えない．この問題を解決するため，それらのブロックを探索開始ブロックなら Outport(出力) ブロックへ，探索終了ブロックなら Inport(入力) ブロックへ置き換えた．

各方程式のみの Simulink モデルへ抜き出した後，Maple/BlockImporter により数式化を行った．しかし，このツールは Simulink モデルから数式へ変換すると入力に対応する変数には $u_{1,1,1}$ から，出力に対応する変数に $y_{1,1,1}$ から順に変数を割り当てていく．よって，別々に抜き出した Simulink モデルではたとえ同じブロックでも別々の変数が割り当てられてしまい，状態方程式・出力方程式を数式化してもシステムの把握が難しいままとなってしまう．そこで，各方程式間で変数を共有する必要がある．

変数の共有には 2 つのテーブルを用いた．一つはシステム全体で共有する変数と抜き出す前の Simulink モデル内の Inport(入力)，Outport(出力) ブロックの名前の組，もう一つは抜き出された Simulink モデル内の Inport(入力)，Outport(出力) ブロックの名前と Maple/BlockImporter により割り当てられた変数の組である．前者は MATLAB の API により入出力ブロックの一覧を取得し，機械的に変数を割り当てることで作成した．後者は MATLAB の API により取得できる入出力ブロックのリストと Maple/BlockImporter により変数が割り振られる変数のリストが対応しているため，それらを元にテーブルを作成した．

2 つのテーブルを使い，抜き出したモデルから変換した数式内の変数からブロック名を取得し，ブロック名からシステム全体で共有している変数を取得した．

3. 適用結果

フィボナッチ数を計算するモデル (図 1) を対象として状態方程式・出力方程式の抜き出し，数式化を行った．フィボナッチモデルは Unit Delay ブロックを 2 つ，Outport ブロックを 3 つ含むモデルのため状態方程式を 2 つ，出力方程式を 3 つ持つが，出力方程式の内 2 つはフィボナッチ数を計算することと無関係のため省略した．

状態方程式では，ブロック名 Fn から探索すると式 (1)，ブロック名 Fn_1 から探索すると式 (2) が生成された．出力方程式では，ブロック名 Out1 から探索すると式 (3) が生成された．

$$x_1(n) = x_1(n-1) + x_2(n-1) \tag{1}$$

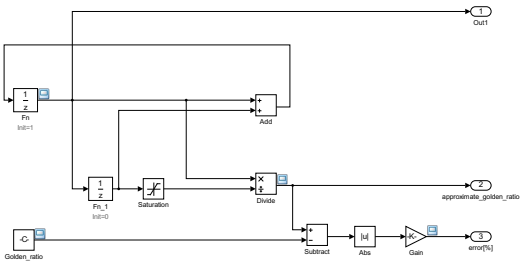

図 1 フィボナッチモデル

$$x_2(n) = x_1(n-1) \tag{2}$$

$$y_1(n) = x_1(n-1) \tag{3}$$

式 (1) に式 (2) を代入することで式 (4) を得ることが出来，正しく Simulink モデルから数式へ変換できていることが確認できた．

$$x_1(n) = x_1(n-1) + x_1(n-2) \tag{4}$$

4. おわりに

4.1 まとめ

本研究では制御モデルを対象とした，Simulink モデルから状態方程式・出力方程式該当部の探索手法及び抽出後の Simulink モデルから数式へ変換した際に起こる問題の解決方法を述べた．制御設計者は常に制御モデルの特性を把握しておくべきであるが，Simulink モデルから数式レベルの特性を把握することは時間がかかる．そこで数式で表すことで容易に特性を把握することができる．これは図 1 のフィボナッチモデル (ブロック数:12) でも式 4 の方がシステムの全体は把握しやすいことからも分かる．

4.2 今後の課題

本研究で提案した手法では複数の状態方程式・出力方程式によりシステムを表現している．そこで，フィボナッチモデルにおいて式 (1) に式 (2) を代入し式 (4) を得たように，最小数の変数・式でシステムを表現することが挙げられる．これはシステムが大規模になるにつれて，制御設計者がシステムを把握するために必要不可欠である．

参考文献

[1] MathWorks: MATLAB/Simulink, `http://www.mathworks.co.jp`
[2] Maplesoft: Maple/BlockImporter, `http://www.maplesoft.com/`
[3] 山口, 池田, 枝廣他: Simulink モデルからのブロックレベル並列化, 組込みシステムシンポジウム 2015(ESS2015),2015,pp.123-124.