

MATLABを活用したHILS設備のプログラム開発

姫路事業所 開発部 開発第3課
廣瀬 昌樹、館 昌志

1. まえがき

近年、自動車業界や航空業界において、電子制御機器(Electronic Control Unit。以下、ECU)や試験機器等の周辺設備のプログラム開発にモデル記述言語であるMATLAB/Simulink(以下、MATLAB)の適用が広がっている。三菱電機(株)姫路製作所においても、取引先からの要望および機能安全等の搭載処理の複雑化から、MATLABを用いたプログラム開発が加速している。

MATLAB言語は、インタプリタ形式の数値演算・解析プログラム言語であり、Simulink言語は、ブロック線図モデルを用いてデータフローを記述するプログラム言語である。Simulink言語で記述するプログラミング手法をモデル開発と称し、プログラム言語全般をMATLABと称している。

車載ECUの試験設備として使用するHILS(Hardware In the Loop Simulation)システムは、図1に示すようにメインコンピュータ(PC)部と本体部(リアルタイムシステム)からなり、さらに本体部は、プロセッサ、アナログ/デジタル入出力とFPGA(Field Programable Gate Array)で構成される。

当所は、HILSシステムのPC部、プロセッサ、FPGAのプログラム開発にMATLABを適用し、駆動系モーター用HILS(以下、モーターHILS)の開発事例を通して、MATLABの開発手法を紹介する。

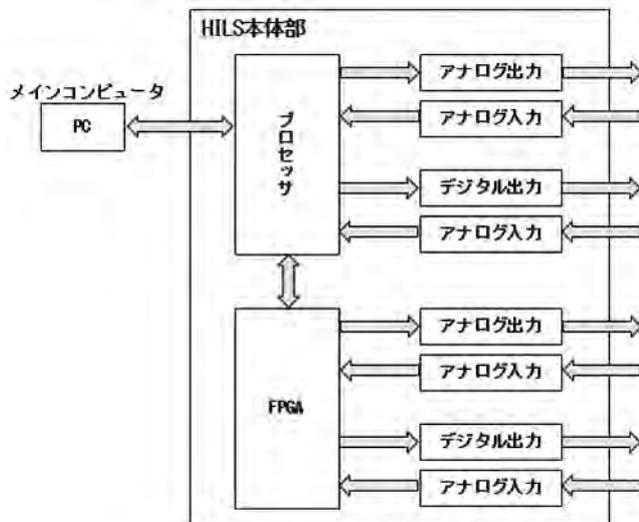


図1. HILSシステム構成図

2. モーターHILSの概要

モーターHILSは、モーター制御ECUを動的に試験するシステムである。モーター制御ECUでは、電気自動車やハイブリット自動車などに搭載される駆動系モーターシステムを制御する。

2.1 モーターHILSのシステム構成

モーターHILSは、以下の3つのモデルに分かれる。

- モーターモデル:モーター制御ECUが制御する駆動系モーターをモデル化したモデル。
- 周辺I/Oモデル:車両状態や他ECUを模擬したモデル
- GUIモデル:ユーザー操作などのデータ入力や自動テストを行うモデル。

3つのモデル構成イメージは、図2のとおり。

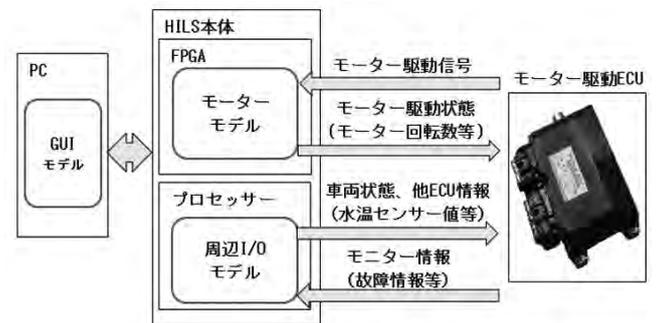


図2. モーターHILS構成図

2.2 モーターモデルの概要

モーター制御ECUは、制御目標トルクと実トルクの差異を減らすために、モーター駆動信号を出力して、モーターに供給される電流を制御する。モーター制御ECUが、モーターの実トルクを認識するには、モーターの回転速度および回転角とモーターに流れている電流値を認識する必要がある。モーターの回転速度および回転角の認識手段は、レゾルバセンサーが主流であり、モーターに流れている電流値を認識する手段は、3相(U、V、W)の電流センサーが主流である。

モーターHILSは、モーター制御ECUが回転角度を認識するために使用しているRDC(レゾルバ・デジタルコンバータ)の分解能(0.04deg)相当でレゾルバ信号を出力する必要があり、出力信号の更新周期は以下の式から求まる。

[レゾルバ信号の出力更新周期]

$$\begin{aligned}
 &= \text{モーター最大回転数} \div 60s \times \\
 &\quad \times \text{モータ1回転の角度} \div \text{RDCの分解能} \\
 &= 20000r/m \div 60s \times 360deg \div 0.04deg \\
 &= 3MHz (0.333 \mu s)
 \end{aligned}$$

レゾルバ信号を3MHzで更新するために、モーターモデルでは高速演算処理(10MHz)が可能なFPGAに組込んでいる。

3. HILSシステムのプログラム開発

HILSシステムのプロセッサのプログラムは、MATLABを用いて作成した。また、HILSシステムのFPGAのプログラムは、MATLABとXilinx社製品であるVivadoに付属のSystem Generator for DSP(以下、SystemGenerator)を用いて作成した。

MATLABを用いたプロセッサ、FPGAのプログラム開発は、モデル開発(プログラム開発)、モデルシミュレーション(デバッグ)、実機シミュレーション(実機確認)の流れで進められる。

MATLABを用いたプログラム開発の流れと、SystemGeneratorを用いたFPGAのプログラム開発での留意点について記述する。

3.1 MATLABのプログラム開発の流れ

(1) モデル開発(プログラム開発)

機能のアルゴリズム設計後、MATLABを用いてモデル開発を行う。MATLABでは、基本的なモデルブロック(加減算、乗除算、論理演算等)のほか、機能的なプログラム処理を組み込んだモデルブロック(MAP演算、行列演算等)が提供され、ユーザーは、図3に示すようにモデルブロックの一覧から、使用するモデルブロックを選択して目的のプログラムを記述する。

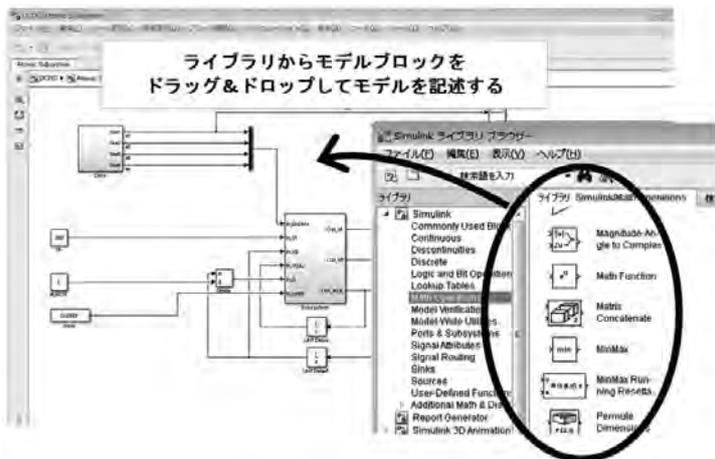


図3. MATLAB開発環境

機能ブロック毎にグループ化したモデルをサブシステムと呼び、サブシステムを外部ファイルから参照するモデルをリファレンスモデルと呼ぶ。リファレンスモデルを使用すると、同様の処理をコピーして冗長なプログラムを作成することなく、1つのモデルを参照してプログラムを作成することができる。サブシステムやリファレンスモデルを活用することで、モデルの可読性が向上したり、機能ブロック毎の単体確認が容易になる。

(2) モデルシミュレーション(デバッグ)

MATLABで作成したモデルをデバッグする場合、図4に示すように、まず、MATLAB上でシミュレーションを実行する。シミュレーションは、サブシステム毎に行い、サブシステムの入力端子に模擬信号を与え、デバッグ対象モデルの信号線の状態をモニタしてモデルのロジック検証を行う。

サブシステム単体のシミュレーションが困難な場合は、サブシステム内でモデルを分割してシミュレーションを実施し、ロジック検証を行う。

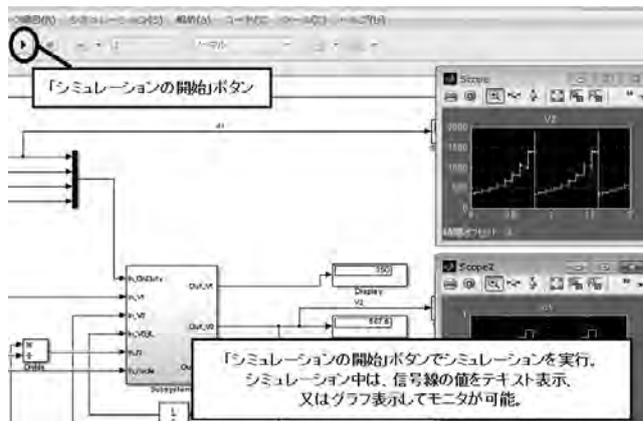


図4. シミュレーションイメージ

サブシステム毎にシミュレーションを実施し、機能毎の単体確認を行った後に実機シミュレーション(実機確認)を行う。

(3) 実機シミュレーション(実機確認)

モデルシミュレーションで単体確認が完了したモデルをHILSで動作させるために、ロードモジュールを作成する。ロードモジュールは、MATLAB CoderやSimulink Coderを用いてモデルをコード化し、HILS用のコンパイラを用いて編集される。

MATLABのモデルブロックには、一定のサンプリング周期(モデルの設定)で動作する離散系モデルブロックと、連続的に動作する連続系モデルブロックがある。

ロードモジュール作成の注意点として、コード化するモデルは、連続系モデルブロックを使用せず、離散系モ

デルブロックを使用して作成する必要がある。

実機シミュレーションでは、図5に示すように、HILS操作PCのGUIモデルから、周辺I/Oモデルのパラメータを変更して、ECUの動的検証を行う。



図5. 実機シミュレーションイメージ

3.2 FPGAのプログラム開発

一般的にFPGA開発環境では、VHDL^(注1)などのハードウェア記述言語(Hardware Description Language: HDL)を基本とした開発環境を用いる。

今回、HILSシステムのFPGAのプログラム開発では、SystemGeneratorを使用した。SystemGeneratorでは、MATLAB用のモデルブロックが提供され、プロセッサのプログラム開発と同様の手法で、FPGAのプログラムが記述できる。図6に示すように、FPGA用ブロックには「Σ」のイメージが表示され、プロセッサ用ブロックと識別できるようになっている。

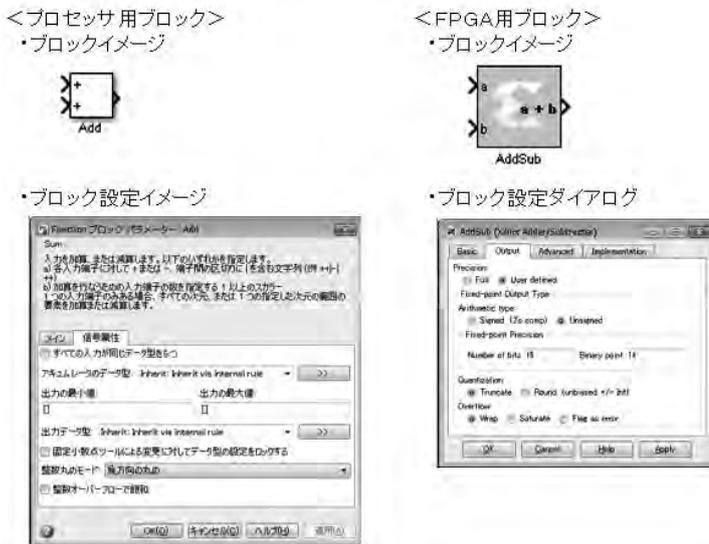


図6. モデルブロックイメージ

FPGAのプログラムを記述する際に、下記2つの留意点がある。

(1) 同期回路設計

SystemGeneratorのモデルブロックは、全て連続系モデルブロックで用意され、ブロックの動作は、FPGA内部で生成されるクロック信号(CLK)のエッジのタイミン

グで動作する。ブロックが動作するクロック信号の周波数は、SystemGeneratorで周波数の変更が可能である。

ブロック毎に処理時間(クロック数)が決まっている。それぞれが並列処理となり、ブロック毎に処理結果の出力タイミングが異なると意図しない結果が得られる。このために、ブロック毎の処理時間の遅れを考慮した同期設計が必要となる。

並列処理のタイミングを合わせるには、図7に示すDelayブロックを挿入したり、各ブロックの出力遅延設定(レイテンシ)を使用してモデルを作成する必要がある。

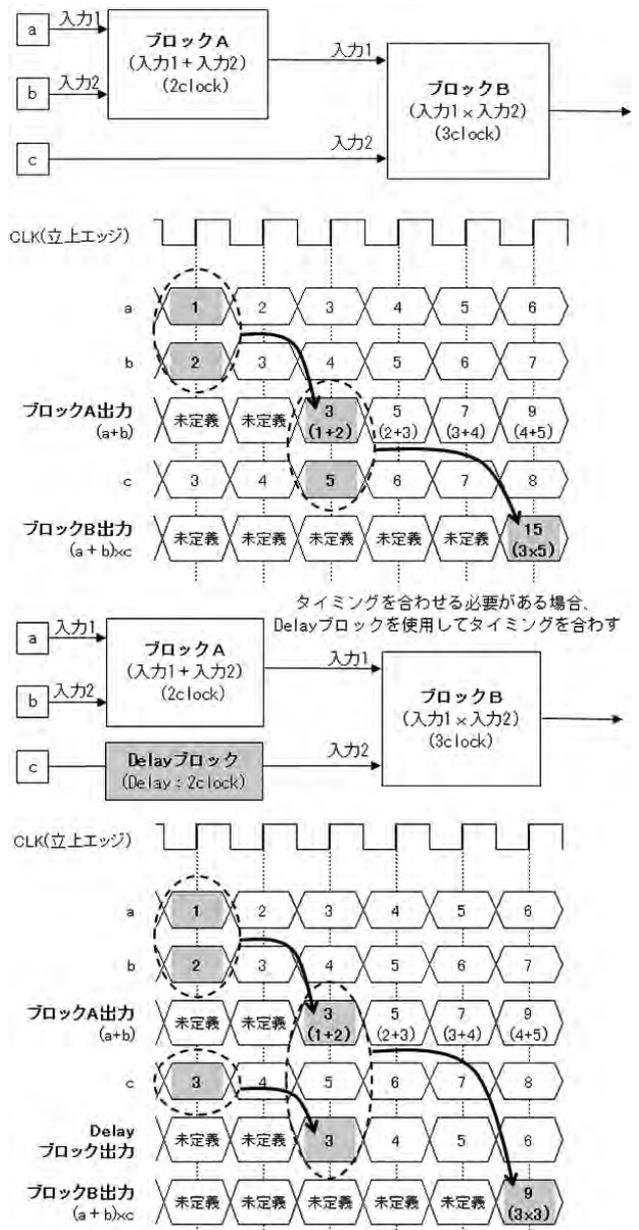


図7. FPGAブロックタイミング

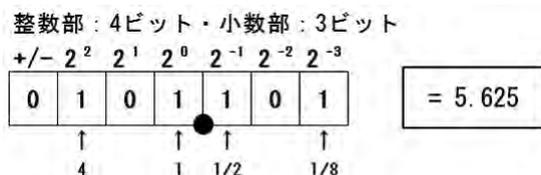
(注1) VHDL: VHSIC HDLの略で、デジタル回路設計用の、ハードウェア記述言語の一種である。VHSICとは、Very High Speed Integrated Circuits (超高速集積回路)を意味する。

(2) 基本データ型

FPGAで扱えるデータ型は、整数、固定小数点と浮動小数点の3種類がある。整数は、各種フラグやカウンターなどに使用し、固定小数点は、各種演算に使用する。

FPGAを用いて、高い周波数で積分演算や積算演算を行う場合、単位時間当たりの変化量がナノ単位の微小データとなるが、演算結果は数千となるため、浮動小数点で表現できない桁数のデータを取り扱う場合がある。

ただし、図8に示すように固定小数点はデータ毎に桁数(整数/小数ビット数)を設定する必要がある。また、System Generatorで用意されている演算ブロックでも、入力値と出力値の桁数設定が必要である。設定桁数が不足した場合は、オーバーフローやアンダーフローが発生するため、使用する全ての演算ブロックの入力値、出力値の有効範囲を把握して、桁数設定を行う必要がある。



※) 小数部3ビットの場合の最小分解能
0.125 (1/8)

図8. 固定小数点

4. 今後の展開

当所で製作しているHILSシステムは、試験対象である車載ECU(モーター制御、インバーター制御、パワーステアリング制御等)に特化した固有のシステムとしている。

今後、ECUの高機能化や制御周期の高速化などから、HILSの処理能力の高速化がさらに求められる。このためFPGAの適用の拡大やプロセッサのマルチコア化(処理分散)など、HILSの高速処理にMATLABを用いたモデル開発を展開していく。

5. むすび

FPGAのプログラム開発にMATLABによる開発手法が適用できたことは大きな成果である。

これにより、プロセッサとFPGAのプログラム開発が同じ手法となり、開発手法の統一化によりHILSシステムのプログラム開発の効率改善が期待できる。

最後に、本システムの開発に際し御指導、御協力いただいた関係者各位に深く感謝申し上げます。

執筆者紹介



廣瀬 昌樹 ヒロセ マサキ
1998年入社。2014年からHILSシステムのソフトウェア開発に従事。現在、姫路事業所開発部開発第3課。



舘 昌志 タチ マサシ
1990年入社。主にHILSシステムのソフトウェア開発に従事。現在、姫路事業所開発部開発第3課。