

MATLAB/Simulinkコード生成ツールとC言語コントローラーの連携事例

トータルソリューション事業所 技術第1部 開発課
藤原 永年、久保 英治郎

1. まえがき

近年、自動車業界の開発手法として定着しているモデルベース開発は、化学プラントや鉄鋼圧延制御の分野にも展開が進んできており、代表ソフトウェアとしてMATLAB/Simulink^(注1)が用いられている。

MATLAB/Simulinkでは、設計したモデルを実機レス環境で検証し、C言語コードを生成することができる。コーディングする必要がなくなるため、開発工数削減と品質向上を図る手段として注目されている。制御システムにおいても、MATLAB/Simulinkより自動生成したC言語やST言語(Structured Text)^(注2)をパソコンやPLC(Programmable Logic Controller)に実装する事例が増えてきており、特にPLCへの適用においては、C言語プログラムを実行できるMELSEC^(注3)C言語コントローラー^(注4)(以下C言語コントローラー)を用いた構成が主流となってきている。

しかし、MATLAB/Simulinkから自動生成したC言語プログラムをC言語コントローラー上で実行させるには、C言語プログラムを修正する必要があることから、その作業を排除するための改善が必要であった。

当課では、MATLAB/SimulinkのC言語コード生成技術をC言語コントローラーに効率よく展開するためのライブラリーを開発するとともに、化学工場の試験プラントで実証試験に適用し、効果を確認した。本稿では、そのライブラリー開発について紹介する。

2. 制御システムにおけるMATLAB活用例

2.1 シミュレーションによるシステム検証

鉄鋼プラントにおける熱間圧延工程では、圧延材の材質や温度による鋼片の伸び・形状張力の計算、電動機の振動、ネットワークの伝送揺らぎなど様々な外乱が発生し、工場内試験で予測できない事象が発生する。

一般的な開発プロセス(ウォーターフォール)



モデルベース開発プロセス

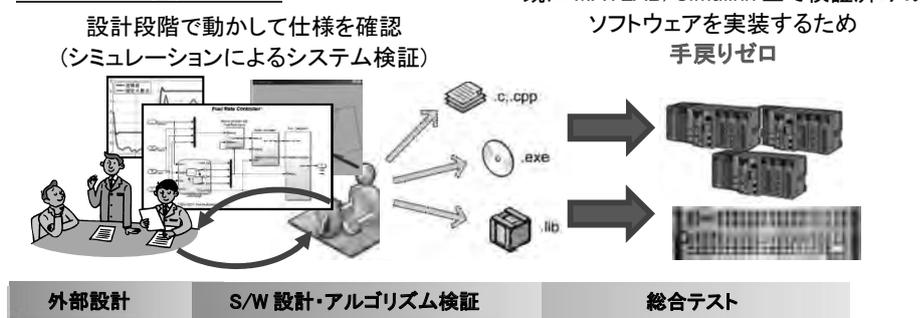


図1. モデルベース開発プロセス

(注1) MATHWORKS社が開発した、行列・ベクトル計算、グラフ化など様々なライブラリーを有したコンピューティング言語及び開発・シミュレーション環境である。

(注2) 数値演算や論理式の記述が容易な高級言語

(注3) 三菱電機(株)製のシーケンサー

(注4) C言語を実行できるMELSECのユニット

MATLAB/Simulinkでは、複雑な計算が必要な物理現象について、予め用意された回転・並進モデルや流体モデルのコンポーネントを用いて、システム検証をシミュレーションすることが可能である。

2.2 モデルベース開発による生産性向上

自動車業界では、ECU(Electronic Control Unit)の開発において、MATLAB/SimulinkのC言語コード生成機能を利用してモデルベース開発を行っている。モデルベース開発は、一般的なウォーターフォール型の開発プロセスとは異なり、図1のように設計段階でシミュレーション検証ができる利点があることから、後工程での手戻りが削減でき、生産性を向上することができる。

2.3 制御システムでの導入状況

制御システムを持つ企業では、自社固有の制御理論を活用して製造製品の品質向上を図っており、それらの制御理論はC言語で実装されているケースが多い。

しかし、制御システムでは制御機器にPLCが多用されることから、自社固有のC言語による制御ソフトウェアを汎用パソコン上で実行し、PLCと連携させている構成が多い。

そこでPLCと汎用パソコンの組み合わせを意識しないソフトウェア開発と、自社固有の制御ソフトウェアの活用を両立させる手段として、MATLAB/Simulinkが注目されている。また、MELSECでは、C言語コントローラーを実装できることから、MATLAB/SimulinkとC言語コントローラーを組み合わせたシステムを検討する企業が増えている。

3. MATLAB-C言語コントローラーの連携機能

3.1 C言語コントローラー連携の検討

(1)C言語コントローラーへの実装課題

MATLAB/SimulinkとC言語コントローラーの連携における一番の課題は、MELSEC上の入出力デバイスへのアクセス処理の実装である。MATLAB/SimulinkにはC言語コントローラーの入出力デバイスにアクセスするライブラリー部品がないため、C言語ソースコードに対し手作業で入出力デバイスのアクセス処理を付け加える必要がある。また、MATLAB/Simulinkより生成されたC言語ソースコードは機械的なソースコードであるため解説が難しく、入出力処理の追加や単体試験に時間を要することが多い。さらに、MATLAB/Simulink上で検証を終えているプログラムに変更が加わるため、再試験する必要があるなどの課題がある。

(2)C言語コントローラー連携の問題解決策

MATLAB/Simulink-C言語コントローラーの連携におけるこれらの課題を解決するため、MATLAB/Simulink上でアルゴリズムを開発する際に入出力処理を埋め込むことができないかを、次の3種類の方法で検討した。

- (a) Simulinkからライブラリーを呼び出す。
- (b) Simulink上のブロックにC言語コードを埋め込む。
- (c) 作成したC言語よりMATLABを呼び出す。

これら3種類について検討した結果、(a)では、一部ブロックボックス化されるためC言語コントローラー上でデバッグができない、また(c)ではC言語コントローラー上にMATLABのエンジンを実装できないことから、(b)での実現を検討した。

(b)を実現するにあたり、C言語コード生成後に修正が不要であるS-Function^(注5)を採用し、C言語コントローラー入出力API(Application Programming Interface)をS-Functionライブラリーに埋め込んで部品化した。

3.2 S-Functionライブラリーの開発

C言語コントローラーが提供しているデバイス入出力APIには、図2のように“C言語コントローラーと外部入出力デバイスのインターフェース”と、“C言語コントローラーとシーケンサーCPU^(注6)内部デバイスのインターフェース”の2種類が存在する。これらのAPIをMATLAB/Simulink上で利用するために、外部入出力デバイスにアクセスするS-Functionライブラリーとして、“①デジタル・アナログユニットインターフェースライブラリー”と、“②MELSEC間インターフェースライブラリー”を開発した(図2、表1)。

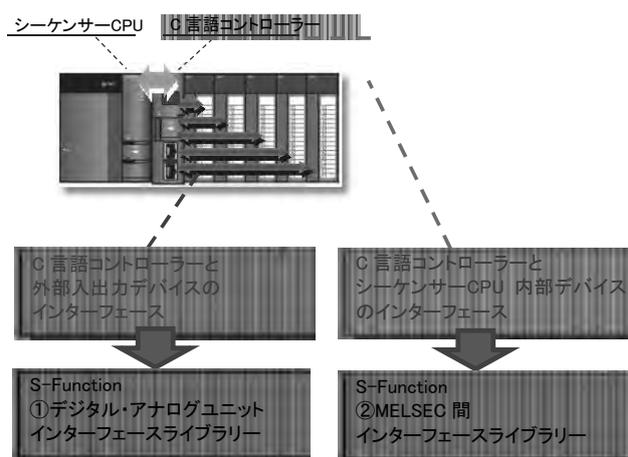


図2. 2種類のインターフェース

(注5) C言語を記述できるSimulinkライブラリー

(注6) 機器のシーケンスを制御するCPU。ラダープログラムやFBD言語を実行する。

表1. S-Functionライブラリー

S-Function	説明
①デジタル・アナログユニットインターフェースライブラリー	
LibBusReadB	デジタル入出力デバイスを読み出す。
LibBusReadW	インテリジェント機能ユニットを読み出す。
LibBusWriteB	デジタル出力デバイスに出力する。
LibBusWriteW	インテリジェント機能ユニットに出力する。
②MELSEC間インターフェースライブラリー	
LibMdReadB	MELSECのBitを読み出す。
LibMdReadW	MELSECのWordデバイスを読み出す。
LibMdWriteB	MELSECのBitデバイスに書き込む。
LibMdWriteW	MELSECのWordデバイスに書き込む。

3.3 S-Functionを用いたC言語コード生成の効率化

MATLAB/Simulinkで作成したプログラムからC言語コードを生成する際、通常は、MATLAB/SimulinkのEmbedded Coder^(注7)でC言語コードを出力し、そのコードに対してMELSECとのI/F処理を手修正でコーディングしていた(図3①)。

しかし、今回開発したS-Functionライブラリーの採用により、C言語のソースコードを修正することなくC言語コントローラー上に実装することができた。また、この作業プロセスでは、コーディング作業の大幅削減、ソースコード変更による再試験を削減することができた(図3②)。

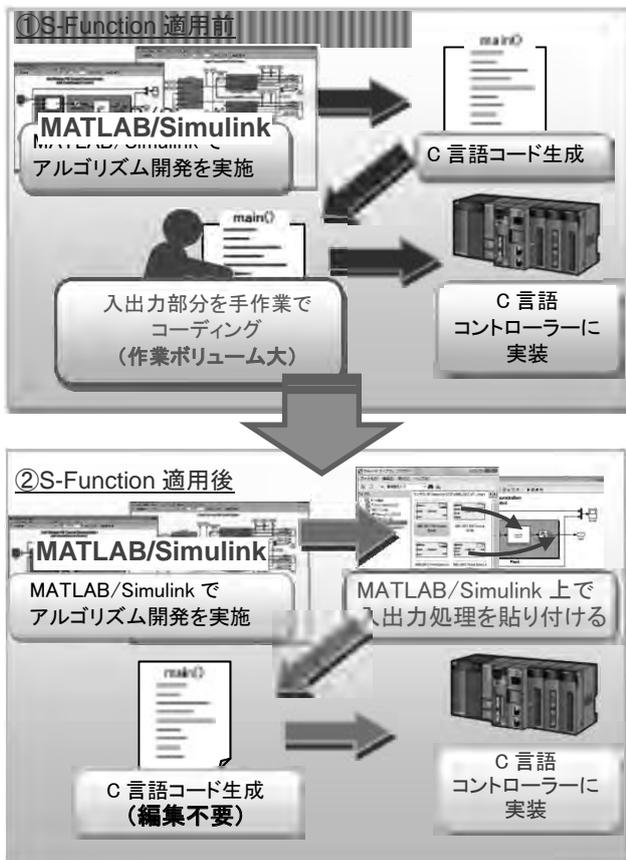


図3. S-Functionによるプロセス改善

4. 適用の効果

4.1 生産性や品質からみた効果

開発したS-Functionライブラリーを、化学工場の試験プラントに実証試験として適用した(図5)。客先で作成されたPID制御のSimulinkモデルにC言語コントローラー入出力用のS-Functionライブラリーを追加し、試験プラントで実証実験を行った。結果、MATLAB/Simulink上と同じ出力ができることを確認した。

今回の実証試験では、開発したS-Functionに起因した障害は発生しなかった。また、C言語コード生成後にコード修正などの手戻り作業もなかったことから、生産効率の改善だけでなく、品質面でも効果を確認できた。今後は、数箇所の試験プラントで検証後、実プラントに展開していく予定である。

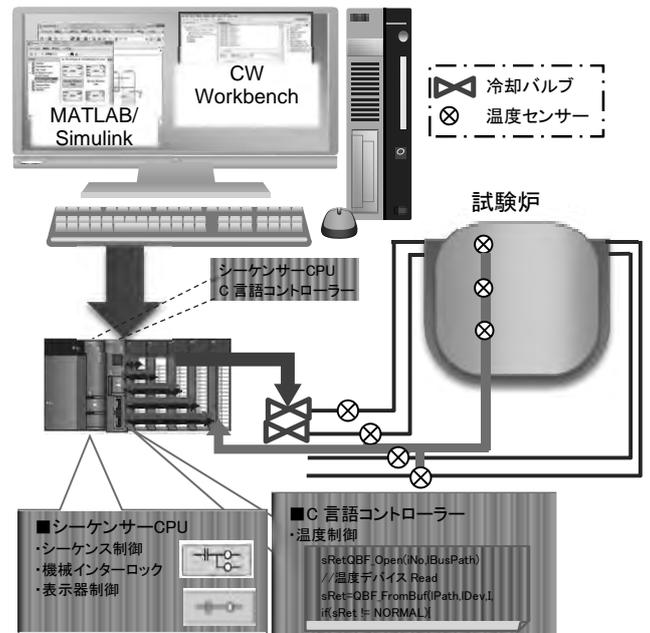


図5. 温度制御プラントへの適用事例

4.2 メンテナンス性からみた効果

2種類のS-FunctionをMATLAB/Simulink上で利用し、生成されたC言語プログラムから外部入出力デバイスおよび、シーケンサーCPU内部デバイスの2種類のデバイスに入出力できるようになり、C言語コントローラー、シーケンサーCPU、外部デバイス間の通信が容易となった。これにより、複雑な計算をC言語コントローラーに実装し、機器動作に関わるインターロック回路をシーケンサーCPUに実装するといったマルチCPU構成^(注8)が構築できた。また、メ

(注7) Simulink上のモデルをC言語に変換するツール

(注8) 複数のCPUユニットを用いた構成。シーケンサーCPUと加工機用CPUなど複数のCPUユニットを同じ筐体上に柔軟に構築することが可能。

メンテナンス頻度が高いインターロック回路の改修や改造は、シーケンサーCPUのみで対応可能となり、メンテナンス性の向上にも貢献できた。

5. 今後の展開

化学工場の試験プラントにおいて、MATLAB/SimulinkのS-Functionライブラリーを用いてC言語コントローラーと連携させた事例を紹介した。鉄鋼メーカーやボイラーメーカーでも、本開発手法を用いたシステム開発の要望が増えていることから、今後も着実に適用分野を広げていきたい。

6. むすび

本開発では、MATLAB/SimulinkをC言語コントローラーに適用するためのS-Functionライブラリーを開発した。今後もモデルベース開発を推進するお客様の要望を取り込み、さらなる改善要望をフィードバックすることで、生産管理と連携するなどのシステム拡張につなげていきたい。

最後に、MATLAB/Simulinkの開発で技術支援いただいた社内外の関係各位に深く感謝申し上げます。

執筆者紹介



藤原 永年 フジワラ ナガトシ
1998年入社。鉄鋼システムのソフトウェア開発、パッケージ製品開発に従事。現在、トータルソリューション事業所技術第1部 開発課課長



久保 英治郎 クボ エイジロウ
1997年入社。主にMELSEC C言語コントローラーを用いたシステム開発に従事。現在、トータルソリューション事業所技術第1部 開発課