

プロセス制御への モデル・ベースド・デザインの 適用と実装(後編)

村田 豊/江本 諭

前編で、モデル・ベースド・デザインをプロセス制御に適用し実装する具体的方法を紹介しました。後編では、そこで取り上げられたコントローラ AD-4820 のハードウェアとソフトウェア開発環境について詳細に解説します。

プロセス制御コントローラの発展

初めに電子化されたコントローラはどのように発展してきたか、簡単に振り返ってみましょう。

● コントローラ設計の変遷

▶ 昔の制御コントローラの設計は力技

制御コントローラの回路設計ではロジック IC を中心に設計をしていましたが、仕様が変わると回路そのものを変更しなければならぬので、設計者は回路図を書き直し、プリント基板の変更や配線を変えて仕様変更に対処していました[図1(a)]。

この時代はハードウェアとコントローラの仕様が一体となっていたといえます。

▶ マイコンの出現によりコントローラの設計も様変わり

これが大きく変わるのはマイコンの出現以降です。1970 年代後半にはコントローラもマイコン化され始めます。マイコンの登場によって新たにソフトウェアの設計が必要になってきました。このことは同じハードウェアでもソフトウェアの変更で異なった仕様のものができるようになるということです。回路図を書き直したり、配線を変更しなくても、プログラムの変更により仕様の変更に対応できるようになったわけです[図1(b)]。

ソフトウェア開発のツールや技術もマイクロコンピュータの発展と同様、時とともに進化していますが、製品仕様とソフトウェア実装との距離感も大きくなってきていると思われます。これは前号で指摘されているように、仕様書からソフトウェアの実装までに多くの専門技術者が入っていることと無関係ではありません。

▶ C 言語による開発ではネックになることが多かった

従来のコントローラでは、C 言語などによってソフトウェアの開発が行われていましたが、肥大化するソフトウェアが新しい製品開発の大きなネックになってきました。それは、仕様をソフトウェアで具体化する設計や実装だけでなく、有効で効率

的な検証、メンテナンスの各段階にわたって担当者間の意思疎通やドキュメントの整備などの問題を生じていたからです。

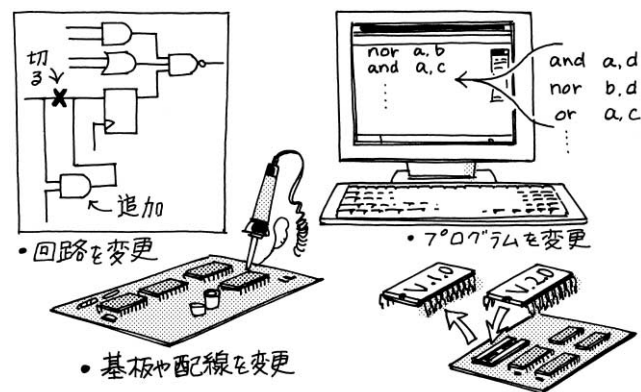
● モデル・ベースド・デザインを生かせるコントローラの開発

このような状況に至った現在、モデル・ベースド・デザインによって開発されたソフトウェアを実装して運転できるコントローラを作ることによって、ソフトウェア開発の効率を大幅に上げることが可能となります。

▶ MATLAB/Simulink の結果を生かしたい

モデル・ベースド・デザインに適した MATLAB/Simulink (The MathWorks 社) は、通信、自動車、医療、教育業界で広く使われているソフトウェアです。その応用範囲は広く工業計測制御の世界でも活用されるようになってきています。これは、このソフトウェア群が計測制御システム開発の上流から実装段階までを一貫して開発できる環境を有しているからだと考えられます。

MATLAB/Simulink による設計、シミュレーションなどの開発の上流側の仕事は、PC の Windows などの OS 上で行うことができます。とくに、設計時におけるシミュレーションは検証方法としてはたいへん有効で、さまざまな入出力の条件や環境条件を仮想的に作り出して効率よく実験検証が行えます。



(a) ハードウェアによる変更 (b) ソフトウェアによる変更

図1 ハードウェアによる変更とソフトウェアによる変更

▶ シミュレーションから実装まで行えるコントローラの条件

シミュレーションなどでのアプリケーション・ソフトウェアの検証が終了すれば、次に実行用ハードウェアへの実装となります。しかし、現場での制御機器に要求される信頼性や長期にわたる製品サポートを考慮すると、シミュレーションで使用したパソコンをそのまま現場で使うことはできません。さらに、現場に適した機器には、MATLAB/Simulinkで開発したソフトウェアを容易に実装できるものがなく、ソース・コードの移植やその動作確認に多大な手間をかけざるを得なくなっていました。

この問題の解決には、システムの開発、シミュレーションから実装までをシームレスに、かつ実際の設置環境で使うことができるコントローラが必要となります。そのためには次のような性能、機能が要求されます。

- ① 上位の開発環境で設計された MATLAB/Simulink のソフトウェアが簡単に実行できること
- ② 設計時のシミュレーションと実際の値との誤差等が容易に比較ができるように、開発用パソコンとの間でリアルタイムでシグナル/パラメータ^{注1}のやり取りができること
- ③ 操作者(人間)に対してわかりやすい操作インターフェースが実現できること
- ④ 実際の設置環境で使用できる、耐環境性能をもっていること
- ⑤ 長期間安定動作が確保されること
- ⑥ 長期にわたるハードウェアやソフトウェアのメンテナンスができること
- ⑦ 現場での急なトラブルに対応できるように、メンテナンスが簡単に行える構造になっていること

▶ プロセス制御コントローラ AD-4820 を例としたモデル・ベースド・デザインを考えてみる

コントローラ AD-4820 はこのような要求を満たすために開発された機器です(写真1)。

この機器には、自動車業界向けに MATLAB/Simulink によるエンジン・テスト・ベンチなどの制御、モニタ・ツールのハードウェアおよびソフトウェアの開発生産で使われた技術を



写真1 プロセス制御用コントローラ AD-4820
(エー・アンド・デイ)

注1: GUI 画面に表示するデータで、更新不可のものをシグナル、更新可能なものをパラメータという。

取り込んでいます。これは、前節で述べたようなソフトウェアの設計、検証、実装、メンテナンスの問題の解決手段を、MATLAB/Simulink によるモデル・ベースド・デザインにより実践してきたからです。この方法を計測制御用のコントローラにも使うことによって、上記の要求を満たすプロセス制御向けコントローラができました。このコントローラは、マイコン・ベースから発展した産業用コントローラと、パソコンで構築されていた自動車業界向けテスト・システムとを統合した形です。

MATLAB/Simulink をベースとすることにより、多様化するニーズに迅速に 대응することができ、各種コントローラ・システム開発の有効な手段になると考えられます。

※ モデル・ベースド・デザインを実現するハードウェアとソフトウェア

● プロセス制御用コントローラによる開発の流れ

モデル・ベースド・デザインを実際実現するために、プロセス制御用コントローラを適用した開発の流れを図2に示します。

モデル・ベースド・デザインでは、設計仕様を MATLAB/Simulink シミュレーション・モデルで表現します。MATLAB/Simulink によるシミュレーションにおいて仕様を満足したシミュレーション・モデルは、C コード自動生成機能を経由し、実行プログラムとして出力されます。実行プログラムは、クロ

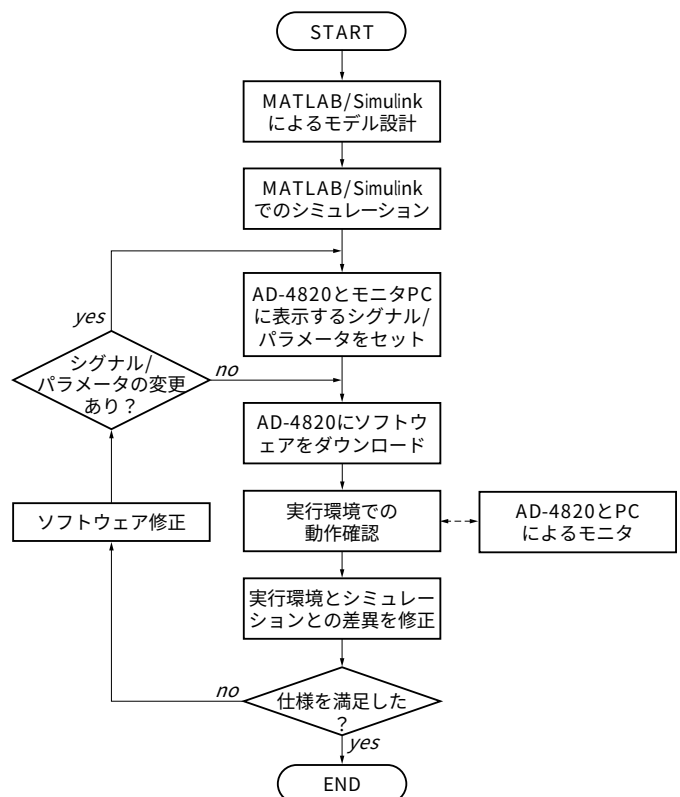


図2 コントローラ・ソフトウェア開発の流れ