

フレームワーク手法に基づく制御系シミュレーションの効率化*

片岡 正俊[†] 小林 正幸[‡] 池田 英俊[§]

三菱電機株式会社

1. はじめに

制御系シミュレーションソフトウェアがパソコン上で動くようになり、ユーザにとって益々使いやすいものとなってきたが、シミュレーションを含む設計作業全体をみわたすと、データの再利用、過去の設計ノウハウの活用等、まだまだ十分ではない。

筆者らは、設計プロセス全体を効率化する視点にたち、フレームワーク手法に基づく制御系シミュレーションシステムを構築した。

本稿では、このシステム概要を紹介すると共に、フレームワーク手法を活用することで、設計作業の効率化が図れるだけでなく、設計ノウハウがシステムに蓄積し、経験の浅い技術者も高度な設計が可能になる事を指摘する。

2. 制御系設計作業とその課題

制御系設計作業とは、図1に示すように、制御対象 $P(s)$ が目標値に追従する制御器 $C(s)$ を設計することである。

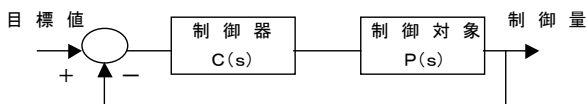


図1. 制御系の構成

一般に、この設計においては、動的システムである制御対象 $P(s)$ ならびに制御器 $C(s)$ をブロック線図で表現し、その制御器のパラメータを所定の制御性能が得られるまで、試行錯誤的に変化させる形で設計が進められる。

この際の問題点として指摘されるのは、ベテラン設計者であれば、その経験から簡単に初期解を得て、その後徐々にチューニングしていくことができるが、経験の乏しい設計者の場合、初期解に至るまでに多くの時間を要する事である。

筆者らは、この対策として、ある程度望ましい初期解が得られるまで、対話型で解の探索をくりかえすユーザインタフェースの提案を行っ

た。[1][2] この探索は通常、特定の制御指標を見ながら行うものであるが、今回この部分をフレームワーク化し、初期解が満たすべき条件入力を行うと、そのためのシミュレーションを繰り返し実行するプログラム（以下、シナリオと呼ぶ）を自動作成し、初期解を自動計算する方式を開発した。

3. フレームワーク手法の適用

フレームワーク手法とは、J A V A プログラミングの効率化手法として、最近注目されている。[3] その考え方は図2に示すように処理の順序をあらかじめパターン化し、そのパターンに必要な固有ロジックを選択、組み合わせてソフトウェアを構成する。

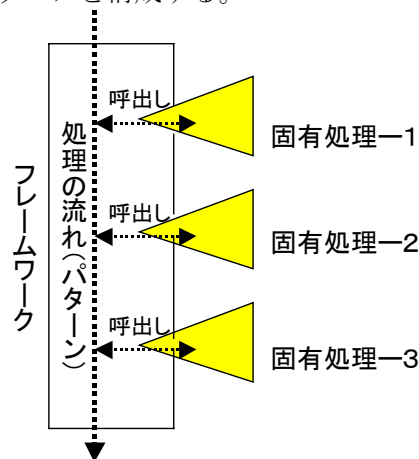


図2. フレームワークの処理概念

筆者らは、この考え方を制御系設計の初期解を求めるシナリオ作成に適用した。一般に制御系のシミュレーションは、以下の5つの部分から構成され、パラメータを変更しては、この処理を繰り返す。

- (1) シミュレーションモジュール選択
- (2) 評価指標モジュール選択
- (3) データセット選択
- (4) シミュレーション実行

* Making control system simulation efficient by using a framework method

[†] Masatoshi Kataoka, Mitsubishi Electric Corporation Information Technology Center

[‡] Masayuki Kobayashi, Mitsubishi Electric Corporation Design Systems Engineering Center

[§] Hidetoshi Ikeda, Mitsubishi Electric Corporation Advanced Technology R&D Center

(5) シミュレーション結果保存
すなわち、設計者はみずからの対象とする制御対象と制御器を指定すると、それに対応したシミュレーションモジュールが選択され、初期解として満たすべき制御指標を指定すると評価指標計算モジュールが選択される。その後、パラメータのデータセットを選択、その中のどのパラメータを変更するのかを指示すれば、自動的に、図3のような処理手順をおこなうシナリオが自動生成される。

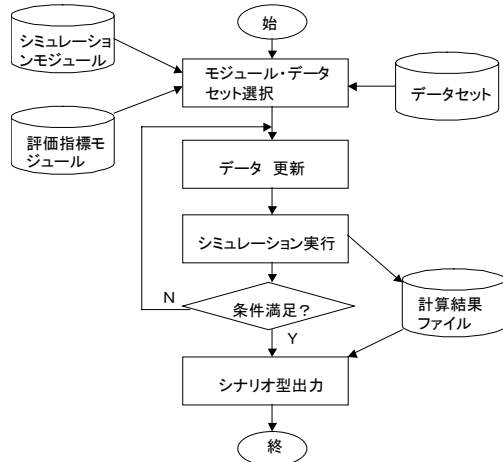


図3. 自動生成されるシナリオ

4. 例題による検証

制御系設計の例題として、無駄時間つき1次遅れをPIコントローラで制御する問題を考える。図4参照。

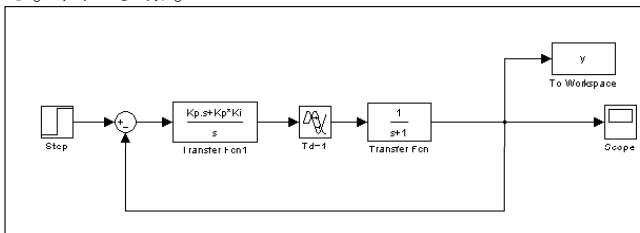


図4. 制御系設計の例題

ここで、ステップ応答の目標値が整定幅-10%ないし+15%の範囲に4秒以内に整定する初期解を求める事が課題である。

この課題に対し、以下のように記述することで、シナリオが自動生成され、シミュレーションを繰り返す。

- (1) 制御対象： $1/(s+1)*e^{-s}$
- (2) 制御器： $K_p(1+K_i/s)$
- (3) 制御目標：整定時間=4, -10%+15%
- (4) パラメータ： $0.1 < K_p < 1.0$
- (5) パラメータ： $0.1 < K_i < 1.0$

その結果であるシミュレーション時間を、こうしたシナリオを使わないで、全探索した場合と

の比較で以下に示すと、

(A) 全探索： 130.2 秒

(B) シナリオ探索： 86.3 秒

となり約2/3の時間に短縮できた事がわかる。また、この際の探索初期値(①)、探索途中解(②)、制御目標を満たす初期解(③)、整定時間最小解(④)を図5に示す。

こうした自動生成されたシナリオを用いる事により、従来より早く、初期解を見つけ、その後のパラメータのチューニングがよりやりやすくなることがわかった。

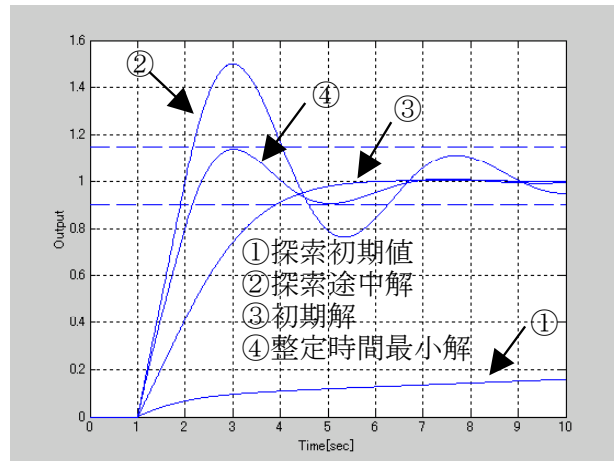


図5. 計算結果

5. 考察

ベテランでない設計者が初期解を見つけるのに、シナリオを活用したシミュレーションが有効である事を検証した。このシナリオ作成も、フレームワーク手法を援用することで簡単に実現できる事もあわせて確かめる事ができた。

このことは、単にシミュレーションを効率化するだけでなく、制御対象、制御器、制御目標、パラメータが1セットとなったケーススタディであり、これを蓄積することにより同種の問題に対する知識ベースを形成する事ができる。

この知識ベースを活用することにより、未熟な技術者でもベテランに近い設計が可能になると筆者らは考えている。

参考文献

- [1] 片岡、小林：ワークステーション上のシミュレーション言語におけるユーザインタフェースについて、第7回シミュレーションテクノロジーコンファレンス、1988年6月
- [2] 小林、小池、川村：フレームワークの概念に基づくシミュレーションユーザインタフェースの開発、第12回シミュレーションテクノロジーコンファレンス、1993年6月
- [3] 特集：フレームワークで生産性をあげる、日経オープンシステム、2002年2月号