

ブロック線図を用いたダイナミックシステムの モデル化・シミュレーションツール

Modeling and Simulation Tool for Dynamic System based on Block Diagram

古賀 雅伸[†]
Masanobu Koga

筒井 勇介[†]
Yusuke Tsutsui

1. はじめに

システムを構成する一つの要素を一つのブロックで表現し、サブシステムの関係性をブロックのつながりという概念でとらえたものがブロック線図である。制御系の開発プロセスでは複雑なシステムをモデリング (表現) するためにブロック線図が用いられる。

ブロック線図で表現されたシステムの解析やシミュレーションを行う場合に必要となるブロック線図から数値計算プログラムのコードを書き起こす作業は、制御器設計の知識の他に数値計算言語 MATLAB[1] や Matlab[2] などの専門知識を必要とし、制御系設計者に大きな負担となる。

制御系の開発では、図 1 に示す制御対象のモデリング、解析、制御器の設計、シミュレーションというプロセスを繰り返し行う [3] ので制御系設計者への負担は更に大きなものとなる。

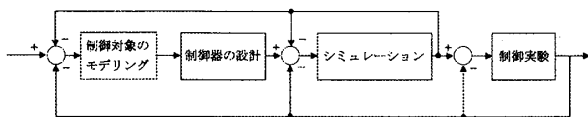


図 1: 制御系開発プロセス

ブロック線図を操作することで、その制御系のボード線図や σ プロットを描画でき、さらにシミュレーション応答を得ることができれば制御系開発プロセスを効率化できる。

そこで本研究では、ブロック線図を用いてダイナミックシステムのモデル化・シミュレーションを実行できるツールの開発を目的とする。なお、本研究で開発するツールを Jamox[†] と呼ぶ。

Jamox の開発言語は Java であり、数値計算ライブラリには NFC(Numerical Foundation Classes)[4] を用いる。

Simulink[5] や LabVIEW[6] などの制御系モデリングを行う商用製品が市販されているが、それらは非常に高額である。また、ソースコードが公開されていないので、演算の並列化・分散化による高速化などの改良や、他の有用なツールとの関係が困難である。

2. Jamox

Jamox は簡単なマウス操作でブロック線図を画面上に作成し、任意の点から任意の点までの伝達関数を求めることが可能である。また、その伝達関数のボード線図、

σ プロット、ステップ応答などのシミュレーション応答を得ることができる。

2.1 GUI

ブロックを集めたブロックライブラリからマウスのドラッグ&ドロップを行うことで各要素の配置を行う。ブロックの端にあるノードから別のブロックのノードにドラッグすることでブロックの結合を行うことが可能である (図 2)。

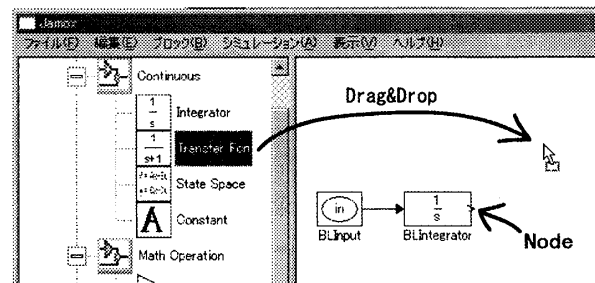


図 2: ドラッグ&ドロップによるブロックの配置

GUI の作成に使用する SWT(Standard Widget Toolkit) では各プラットフォームのネイティブインターフェースを利用することになるので、ユーザーが使用しているプラットフォームのインターフェースと相違ない使用感が実現されている。

2.2 伝達関数の結合計算

ブロックの結合を行うことは、システムの伝達関数を四則演算することと等価である。伝達関数の乗算を繰り返すとシステムの次数が増加し、演算に時間がかかるなどの問題点が出てくる[§]。結合演算をするうえで重要なことは演算の回数をもっとも少ない形にすることである。そこで本研究では、記号処理を行うことで縮約された結合計算を行うことができる結合計算アルゴリズム [7] の実装を行った。

2.3 シミュレーション結果の表示

一般のツールではシミュレーション応答をツールに付属するプロットツールを用いて表示する。別のツールを利用したい場合には、シミュレーションの結果を数値としてファイルに書き出し、それを編集する。

Jamox にはプロットツールは付属していない。結果の表示は Outputter インターフェースを実装したクラス (例えば OutputImpl) が行う。OutputImpl クラスはシ

[†]九州工業大学

[†]Jamotcs(Java Agile MOdeling Tool for Control System) から

[§]数値的に約分も可能だが誤差を含むことがある

ミュレーション結果を時系列と値という行列で受け取り、目的の処理を行う実装を行う。

実装した Java クラスは一つのユーザー出力ブロックで指定することが可能である。Jamox では標準実装としてフリーのグラフ描画ツール gnuplot へのラッパークラスである Gnuplot クラスを提供している。

3. パラメータのチューニング

我々のグループではシステムの伝達関数のグラフを表示するだけでなく、表示したグラフを数式の構造を保ったままパラメータの変更が可能なツール Jpit[8] を開発中である。Jpit も開発言語は Java であるから容易に連係が可能である。(図 3)。

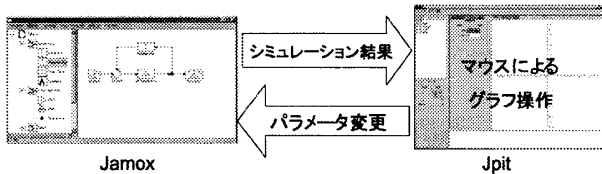


図 3: Jamox と Jpit の連係

Jpit にシステムの伝達関数を渡し、ユーザーが Jpit で表示されたグラフの変更を行うと、ブロック線図のパラメータも変更できるようになる予定である。一般に制御系のチューニングを行う場合、ユーザーはグラフ表示で性能を確認した後、パラメータを変更し再度シミュレーションを行うという動作を繰り返す。上述の機能を利用することで目的の特性にパラメータチューニングを行うことが容易になる。これは設計者にとっての負担を大きく軽減する。

4. Jamox を用いたモデリング

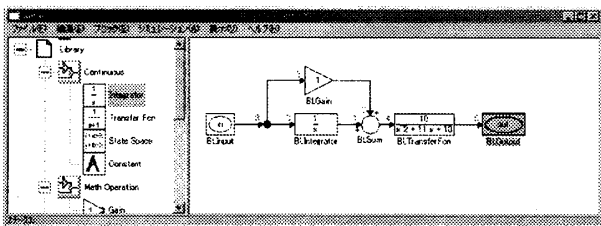


図 4: Jamox によるモデリング

例題 制御対象が $P(s) = \frac{10}{(s+1)(s+10)}$ で与えられたとき、PI 補償を行う [3, pp151] コントローラを設計する。

本ツールを用いて $P = 1$ で PI 補償を行った場合の画面を図 4 に示す。左側の Tree から右側の Canvas にブロックをドラッグ&ドロップで配置し、ブロック同士を結合するだけで作成することができる。

ブロックを配置し終わったら、メニューから「ボード線図」というメニューをクリックするだけで入力ブロック

から出力ブロックまでの伝達関数を導出し、ボード線図を描くことができる。ボード線図の表示にここでは Jpit を利用している。

図 5 に PI 制御をする前のゲイン線図と PI 制御を行ったあとのゲイン線図を示す。PI 制御を行った方が開ループゲインが高いことなどがわかる。

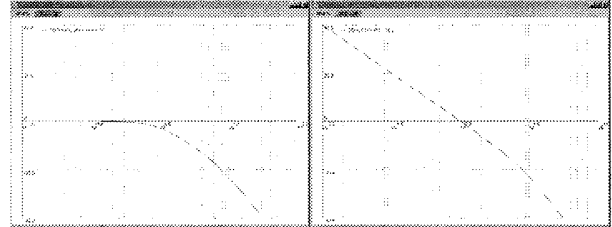


図 5: コントローラを用いるゲイン設計結果

5. まとめ

ブロック線図を与えることで、連続時間線形システムのモデル化・シミュレーションを行うことができるツールを開発した。

現在は線形連続時間システムにしか対応していないが、今後は非線形システム、離散時間システムにも対応する予定である。また、用意されたブロック同士を組み合わせることでユーザー定義ブロックとして取り扱えるようにする。並列化や分散化による高速化の他の有用なツールとの連係を進める予定である。

参考文献

- [1] 古賀雅伸. Linux・Windows でできる MATIIX による数値計算. 東京電機大学出版, 2000.
- [2] *The MathWorks - Matlab*. <http://www.mathworks.com/products/matlab/>.
- [3] 杉江俊治, 藤田政之. フィードバック制御入門. コロナ社, 1999.
- [4] 松木毅. Java 数値計算パッケージの開発と制御系設計への応用, Feb. 2003. 2002 年度卒業論文.
- [5] *The MathWorks - Simulink*. <http://www.mathworks.com/products/simulink/>.
- [6] *National Instruments - LabView*. <http://www.ni.com/labview/>.
- [7] 原辰次, 小林史典. 制御系 cad のための結合計算アルゴリズム. Vol. 29, No. 2, pp. 105-114, 1985.
- [8] 森智宏. グラフで設計仕様を与えることができる制御系設計支援ツールの開発, Feb. 2003. 2003 年度卒業論文.