

Scilabによる制御工学演習について

海津 宏

A Study on Control Engineering Exercise using Scilab Software

Hiroshi KAIZU

1. はじめに

現在、制御系の教育、研究機関および実設計、解析のために多くの工学技術系ソフトウェアが用いられている。特に、MathematicaやMatlab等の数値処理システムは、かなり著名であり、これらの特徴を生かして工学上の問題解決の研究や工学教育への導入が進んできている。このMatlab/Simulinkは、商用ソフトとして古くから有名でその評価も高い。Matlabには、多くの関数が提供されており、高度な数値計算も比較的簡単な構文を用いることで容易に行うことができ、GUIによるシミュレーションや多くのToolbox、デモソフトの利用が可能である。但し、student版、アカデミック版、一括パック版など比較的安価なライセンス料で購入できる場合もあるが、費用対学習効果、毎年の更新料、初学者への使いにくさなど改善を求めべき点も少なくない。安価または無料の数値ソフトウェアを有効利用できることは、実用上とても望ましい。例えば、学生が自宅学習・課題に利用したい、もっと簡便なツールで教育効果を高めたい、エンジニアの再教育や講習会等で自由に利用したいなど目的、利便性に応じてソフトを使い分けたいとの考え方もできる。

本稿では、フリーのソフトウェアであるScilabを活用した制御科目への授業展開における例題演習と運用に関連する考察についての報告を行う。

2. Scilabについて

制御系では、Basic、c、FORTRANなどによる教育用ソフトがかつては広く活用されていた。古くにScilabを利用した段階では、実用面で満足できるものでは無かったが、最近、利用環境、操作性などが改善され見直しが進んでいると思われる。Scilab (A Free Scientific Software Package) は、フランスのINRIA (国立情報処理自動化研究所) とENPC (国立ポンゼシヨセ大学) によって開発されている数値計算用のソフトウェアパッケージ

である。日本でも今年、普及促進を目的としたToolbox Contestが開催されている。

Scilabの主な特徴は、次の通りである。

- ・フリーソフトのため広く一般に導入が可能である。
- ・Unix/Linux, Windows, Macなどの広いOSに対応可能。
- ・オープンソースファイルのためユーザーによる関数設定等が容易に可能である。
- ・GUIによるコマンド操作が可能である。
- ・多項式・行列・数値計算や特定分野 (制御、信号、通信ネット、最適化、非線形系、生物等) への関数機能が豊富に用意されている。
- ・Matlab やFortran, c, Excel, Maple, Labviewなどへのリンクが可能であり、拡張性も十分ある。
- ・比較的コンパクトな実行形式である。
- ・help機能、マニュアル等は、英語、仏語のみしかない。

Scicos (Scilab Connected Object Simulator) は、連続系および離散系のシステムモデリングとシミュレーション用のScilabパッケージである。MatlabにおけるSimulinkに相当しており、ブロックを相互結合することによりモデルが構築できるグラフィカルエディターを備えている。Scicosにおいて各信号は起動時間の配列に関連づけられており、一連の時間間隔とイベントと呼ばれる独立した点の組から構成されている。これらにより信号処理プロセスを視覚的に把握することができ、プログラムの作成を簡単化できるという特徴を有する。

制御分野に関連するMatlab以外の商用ソフトの例は、次の通りである。

MATRIX, Mathematica, Mathcad, MuPAD, Maple,

EASY5, Labview, Working Model

一方、フリーソフトとしては、数値解析ソフトのOctaveなどがあり広く使われているようだが、Matlabとの互換性や図形表示機能では Scilabの方が少し上回っている。その他のフリーソフトの例は、次の通りである。

MaTX (行列計算), guplot (グラフ作成),

Pro/DesktopExpress (3次元CAD), Circuit Maker

平成18年12月20日受理

(電子回路シミュレータStudent版), Pspice(電子回路シミュレータ評価版)

制御系科目では、学習効果を考えるとコンピュータ演習が重要不可欠なことから、簡便、実用的で、CAE (Computer Aided Engineering) 効果も期待したツールとして、推奨しつつ使い始めたところである。なお、シラバス等を参照にするとすでにいくつかの教育機関では、以前から実際に活用されているようである。

3. 制御工学演習について

演習は、電気工学第4学年および一部を第5学年を対象とし、実験は4年生のみを対象としている。項目の概要としては、各々次の通りである。

- ・伝達関数、過渡応答、周波数応答、極・零点配置、安定性、補償、設計、解析、状態空間法、最適制御法等
- ・各種LCフィルタ回路の設計と周波数特性の実測比較

Scilabの利用法、例題および演習例を以下に記す。

・利用法

Scilabの導入は、ネット上からインストール可能であり、現在のバージョンは、4.1である。基本定数には%が前につき、円周率 %pi、虚数単位 %i 等となる。コマンドプロンプト記号は `-->` であり、例題文末の `//` はコメントを表している。

・例題 1

```
-->s=poly(0,'s') //変数sによる多項式の定義
-->G1=1/(1+s*0.01) //伝達関数の定義
-->sys=syslin('c',G1) //連続時間系'c'の定義
-->bode(sys) //ボード線図の描画
-->t=[0:0.001:0.1] //初期時間:刻み:最終時間
-->y=csim('step',t,sys); //連続時間系ステップ応答
-->plot2d(t,y); // グラフ描画
-->xtitle('step response','time[sec]','response')
```

(図1、2参照)

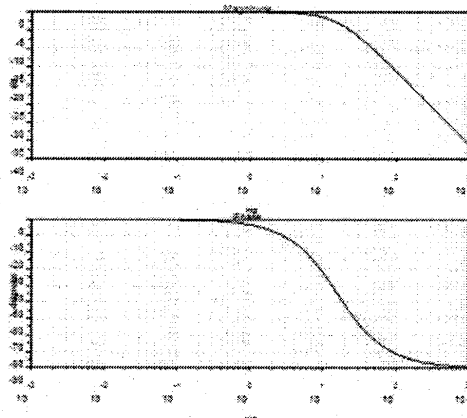


図1 例題1のボード線図

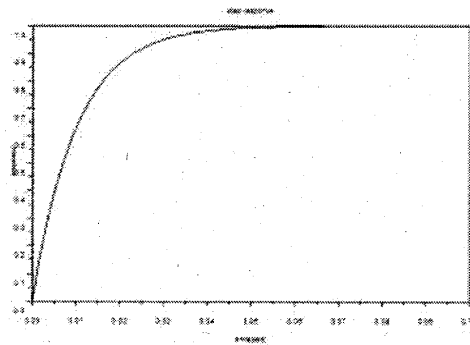


図2 例題1のステップ応答

・例題 2

```
-->s=poly(0,'s') //減衰係数と過渡特性
-->G2=1/(s^2+s*0.1+1) //伝達関数の定義
-->G2=syslin('c',G2) //連続時間系'c'の定義
-->t=[0:1:100]
-->y=csim('impuls',t,G2); //インパルス応答の計算
-->plot2d(t,y)
```

(図3参照)

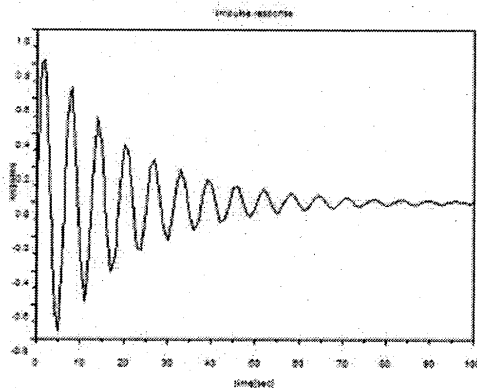


図3 例題2のインパルス応答

・演習 1

$$G(s) = \frac{s - z}{z(s+1)(s^2+s+1)} \quad (1)$$

ここで、 $z=0.4, 0.6, 0.8, 1.0, \dots$ と変化させるときのステップ応答(またはインパルス応答)を観測する。実数零点 z の値が実数極から離れてゆくと、最大行き過ぎ量が大きくなり、逆に近づいてゆくと零点の影響が応答に現れなくなることが確認できる。なお、繰り返し計算には、for z =初期値:刻み:最終値 end を用いる。

同様に、いろいろな $G(s)$ を設定して、極・零点配置と過渡特性との関係やダイポールの影響、オーバーシュート、アンダーシュート(逆応答)および周波数応答、帯域幅について考察してみよう。

・例題 3

図4は、PID調節による直結フィードバック系を表している。制御対象が $(s-1)/(s^2+s*5+1)$ 、PID補償器のゲイン K が、 $K=Kp(1+Td*s+1/(Ti*s))$ としたとき、PIDパラメータ Kp, Td, Ti を変化させて応答を考察せよ。図5、図6は、 $Kp=-1, Td=1, Ti=0.1$ のときのステップ応答とボード線図を表している。

同様に、他の補償法を用いた場合の特性解析をしてみよう。

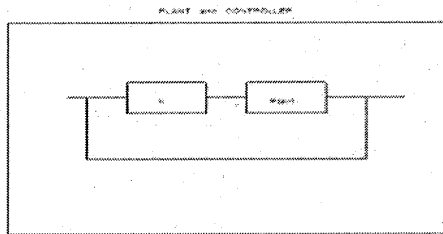


図4 PID調節

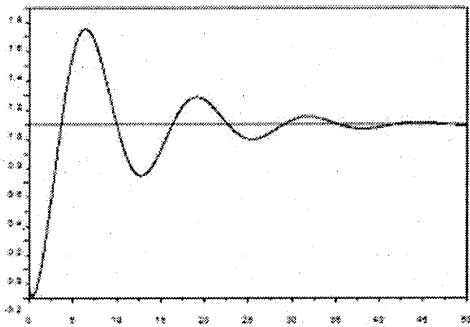


図5 例題3のステップ応答

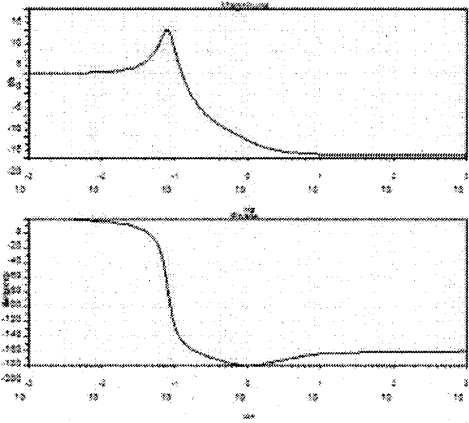


図6 例題3のボード線図

・例題 4

$$G(s) = \frac{1}{(s-1)(s+1)(s+3)} \quad (3)$$

ナイキスト線図、根軌跡を求めよ。(図7、8参照)

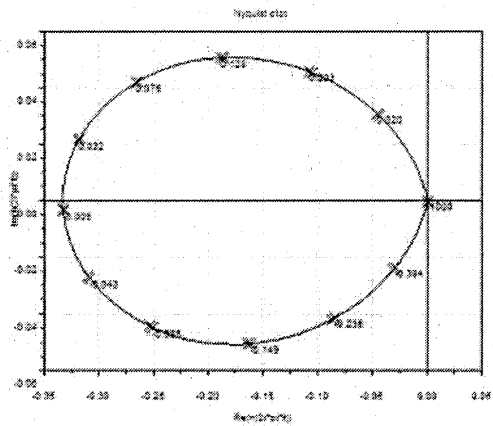


図7 例題4のナイキスト線図

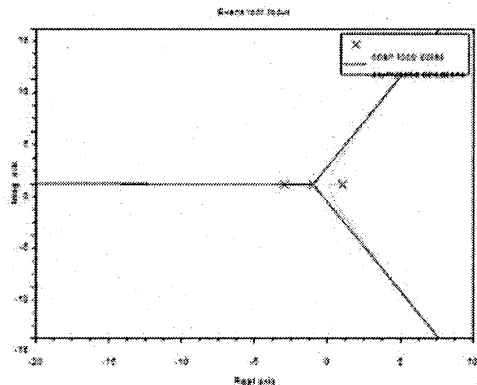


図8 例題4の根軌跡

状態空間法表現によるシステムの記述として、次のような連続時間線形システムS2を考える。

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad C = [1, 1], \quad D = 0 \quad (4)$$

・例題5

```
-->G1=(s+1)/(s^2+s+1)
-->S1=syslin('c',G1) //連続時間線形システムS1の定義
-->S1=syslin('c', (s+1)/s^2+s+1)
//S1の伝達関数を直接代入する別定義
-->S1=syslin('c', s+1, s^2+s+1)
//S1を分母多項式, 分子多項式で表現する別定義
//分母多項式は denom と分子多項式は numr で取り出せる。
-->A=[1 2;3 4];B=[1;0];C=[1 1];D=0; //行列の定義
-->S2=syslin('c', A, B, C, D)
-->[A, B, C, D]=abcd(S2)
//システムS2の (A, B, C, D) の各行列は、コマンドabcdで取り出すことができる。
-->tf2ss(s1)
//伝達関数S1から状態空間表現のS2への変換
-->ss2tf(S2)
//状態空間表現S2から伝達関数表現S1への変換
-->plzr(S1)
//極零点配置の描画。図中の○は極、×は零点
```

線形システムの接続形式における各接続のコマンド表記は次の通りである。

- 直列接続 : S1*S2
- 並列接続 : S1+S2
- フィードバック接続 : S1/.S2

```
-->ss2tf(S1/.S2)
//S1に対してS2をフードバック接続した場合のシステムの全体の伝達関数
-->plot2d(t, csim('impuls', t, S1))
//インパルス応答の描画
-->clf() //描画の消去
-->u=cos(t) //COS関数による制御入力の設定
-->plot2d(t, csim(u, t, S1))
//入力をcostとしたときの出力応答の描画
```

以下に実際の演習で用いるコマンド群とその概要を表す。ここでの線形システムとは、例題のS1またはS2に対応する。但し、離散時間系における場合等は割愛する。

- nyquist(S1) : 周波数応答法のナイキスト線図の描画
- black(S1) : ニコルス線図の描画
- routh_t(denom(S1)) : ラウステーブルの導出
- roots(denom(S1)) : S1分母多項式の根の導出
- g_margin(線形システム) : ゲイン余裕
- p_margin(線形システム)+180 : 位相余裕でゲイン0 dBのときの位相値が算出されるので+180度が必要
- kpure(線形システム) : 安定限界ゲイン
- evans(線形システム) : 根軌跡
- expm(A*t) : 遷移行列e^{At}の計算
- 1/A : 逆行列
- spec(A) : 行列Aの固有値
- cont_mat(線形システム) : 可制御
- obsv_mat(線形システム) : 可観測
- canon(線形システム) : 可制御正準形式への変換
- rank(行列) : ランク (階数)
- trzeros(線形システム) : 多入出力系の伝達零点
- h_norm(線形システム) : H[∞]ノルム
- invsyslin(S2) : 逆システム
- lyap(A, C, 'c') : 連続時間リアプノフ方程式の解

・演習2

極配置による状態フィードバックゲインKを設計し、補償前後の特性を考察せよ。また、提示したコマンド群をなるべく多く実行し、結果をまとめて系の設計解析を行ってみよう。

例えば、S2において極指定を[-3+i, -3-i]と設定して系の全体で考えてみる。ゲインKは次のように求められる。

```
-->K=ppol(A, B, [-3+%i -3-%i])
```

Kの出力は、11と18.7となり、次式の通り検算すると

```
-->spec(A-B*K)
```

元の極指定値が確認できる。その他にリカッチ解による最適制御設計、位相面解析等も可能である。

・演習3

Scicos を用いた1入出力系のLQG法による直列補償の設計手順と出力応答は、次のようなステップで順次パラメータを入力することで得られる。各項目を確認し、Scicos によるシミュレーションを実行してみよう。

- 1) 連続時間系、離散時間系の選択
 - 2) 制御対象の伝達関数の設定
 - 3) 重み行列の設定
 - 4) 雑音行列の設定
 - 5) 周波数応答 (ボード線図) の描画
 - 6) 過渡応答 (ステップ応答、インパルス応答) の描画
- 図9は、Scicosによるシミュレーション画面例である。

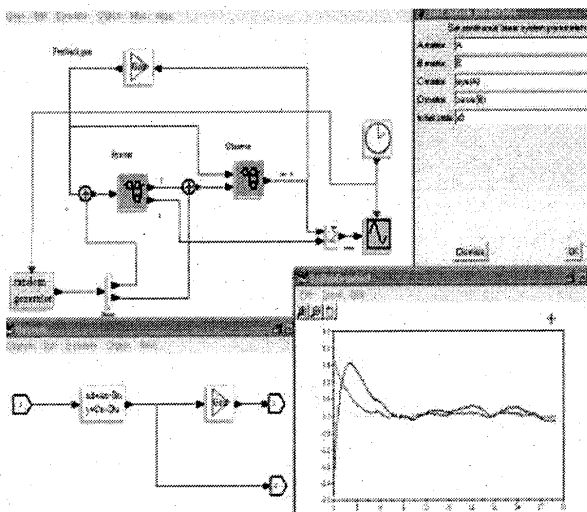


図9 Scicosによるシミュレーション画面例

・演習4

直動型倒立振子のデモソフトを参照して、そのプログラムと制御内容を説明せよ。次にパラメータや制御方法をいろいろと変化させた場合のシミュレーション結果を考察せよ。線形化状態方程式が明示されているのでいろいろな設計や解析法を適用してみよう。(図10参照)

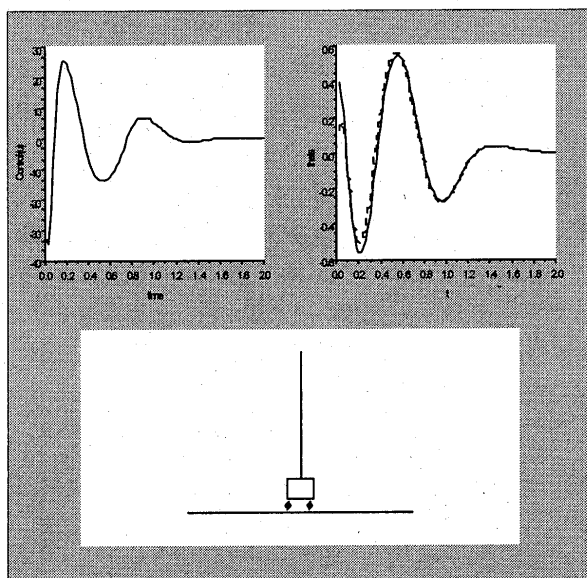


図10 倒立振子モデルの出力画面例

実際のScilab演習では、学生は最初、図形描画・アニメーションに明確な反応が得られた。基本的な使い方や教科書に対応した例題等を随時示すことによって、個々の進度に合わせて自宅や時間外学習での学習効果があり、細かな使用法も学生間の情報交換でうまく行われること

が可能であった。これらの成果をもとに演習内容や学習評価を整理することがなお必要で今後の重要な課題ではあるが、いくつかの点がまとめられた。即ち、グラフ表示(可視化)が効果的に理解を助けること、個別進度可能なこと、自宅や課外学習が容易に可能なこと、ネット検索で参考情報が得られやすくなったこと、対話形式での段階的の進度であり英文エラーも次第に慣れること、操作性も順応性との兼ね合いによるなど概ね良好な評価であった。但し、英語環境・操作簡便性等の大幅な改善、マニュアル・例題類の拡充と精査、他ソフトへの連携性向上、個別進度への細やかな対応、Matlab/Simulinkとの対比等多くの要望も寄せられた。よく相談しながら、効果的な運用法を情報交換しつつ着実に改善していく必要があると考えている。なお、制御以外の適用分野・科目としては、信号処理、応数、応物、回路解析、メカトロ、ファジィ、非線形等が挙げられる。

4. おわりに

Scilab/Scicos は、無料ソフトではあるが、かなり多くの機能を持ち、拡張性も十分あると考えている。今後は、商用ソフトとの使い分けが一層進むものと思われる。今後の普及に伴い多くの利用者からのライブラリや有益な情報が提供されることが大いに期待できる。この点は、工学教育面からもかなり魅力的であり、技術教育関連ソフトのさらなる拡充と操作性、利便性向上等を深く希望している。

参考文献

- 1) 野波, 西村, 平田: MATLABによる制御系設計, 東京電機大学出版, 1998
- 2) 足立: MATLABによる制御工学, 東京電機大学出版, 2002
- 3) 廣川: Scilabによる制御系シミュレーション, Interface, pp.106-115, 10月号, 1999
- 4) 廣川: Scilab入門, URL http://www.geocities.jp/rui_hirokawa/scilab/
- 5) 井上, 川田, 西岡: MATLAB/Simulinkによるわかりやすい制御工学, 森北出版, 2001
- 6) 三谷: Scilabで学ぶデジタル信号処理, CQ出版, 2006
- 7) 細江: システムと制御, オーム社, 2004
- 8) Scilab URL <http://scilabsoft.inria.fr/>