

Mathematicaを用いた非線形系のフィードバック線形化と制御のための設計支援法

Fukuzawa, Mitsunori / Masubuchi, Masami / 増淵, 正美 / 福沢, 充孝

(出版者 / Publisher)

法政大学工学部

(雑誌名 / Journal or Publication Title)

法政大学工学部研究集報 / 法政大学工学部研究集報

(巻 / Volume)

29

(開始ページ / Start Page)

115

(終了ページ / End Page)

128

(発行年 / Year)

1993-03

(URL)

<https://doi.org/10.15002/00003858>

Mathematicaを用いた非線形系の フィードバック線形化と制御のための設計支援法*

福沢 充孝†, 増淵 正美‡

A Computer Aided Design Method for Linearization and Control of Nonlinear Systems by using 'Mathematica'

Mitsunori FUKUZAWA and Masami MASUBUCHI

Feedback linearization methods using differential geometric approach for the developments of nonlinear control systems have been regarded as one of the most prospective techniques in these years. However, the nonlinearities in the existing industrial processes have wide varieties and the conditions of linearization require quite intricate calculations and energy consuming time. The authors propose a computer aided design system for linearization and control of nonlinear systems by using the mathematical software system "Mathematica" and show good result.

1 緒言

ボイラプラント、ロボット、化学プロセスなどの実在システムを制御する場合に、その数学モデルを構成すると、非線形システムとなることが多く、一般的な制御系設計が困難である。この時、ある動作点近傍にて近似線形化して制御系を構成することが通常行われている。しかしこの場合、設計時の動作点とは別の条件でシステムを運転させると動特性が設計時とは異なるため、最適に調整された制御系であっても制御性能が劣化してしまうという問題点がある。

しかし、1970年代から非線形システムを統一的に扱える微分幾何学的手法を中心とした非線形制御理論の研究が発展し、線形システムに対する重要な概念（可制御性、可観測性など）が非線形システムに翻訳されてきた^[1]。そして1980年代にはR.W.Brockettの 'feedback invariant' の概念を用いて、非線形座標変換と状態フィードバックによりシステムを厳密に線形化する状態空間線形化法^[2]が提案された。この手法はシステムの状態方程式を近似を用いずに厳密に、か

*第31回計測自動制御学会学術講演会にて一部講演

†システム工学科博士課程在学

‡法政大学工学部

つ大域的に線形化する方法であり、動作点に依存しない一定の正準形に等価変換できることに特徴がある。筆者らもこの手法を用いて、流体流量などの媒体を操作して行う温度制御などの双線形システムを検討し、この手法が双線形システムの設定値変更などの動作点移動による特性変化の補償に有効であることを示し、双線形システムに対する一般解を導出した^[7]。ところが、この手法には対象システムに対する厳しい制約条件があり、場合に応じて線形化条件を緩和した他の手法^{[3]-[6]}の適用を考える必要がある。また、プロセス制御系では動作点変動や原料の運転コスト、製品の需要の変動などからくるパラメータ変化が制御系設計後もしばしば起こり得るので、制御系設計後のシステムのパラメータ変化による影響を考慮した解析を行う必要もある。このような各種線形化法の条件判定や線形化制御則の導出、パラメータ変化を考慮した解析には膨大な計算量を要するため、計算機の援用が不可欠となる。

そこで著者らは、数学ソフトウェアシステムMathematica[§]を用いて非線形系に対して、各種線形化法の条件判定、及び設計計算を自動化し、さらに複数のパラメータ変化に対する影響を極位置変動範囲などで表示する設計支援・解析システムの研究を行った。Mathematicaは近年開発された最新のコンピュータ数学システムで{数式処理/数値計算/グラフィクス/文書作成}などの機能を1つのシステムとして含む最も強力なツールであり、設計支援システムの構築に有効であると考えている。

本報告ではこの設計支援システム上に構築したいくつかの線形化法を紹介し、設計支援システムの特徴、基本構成、実行手順について述べ、非線形システムの解析法とフィードバック線形化による制御系の設計支援法、及びロバスト性解析法などを例題を用いて具体的に示すことにする。

2 非線形系の線形化理論

本設計支援システムに実装した線形化手法の中で、厳密な状態空間線形化法^[2]と近似線形化法の1つである拡張線形化法^[6]について説明する。まず、状態空間線形化法を考える上で、必要な微分幾何学の数学的記述については参考文献を[1]を参照。

2.1 状態空間線形化法

次のSISO非線形系を考える。

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u \quad (1)$$

$$y = h(\mathbf{x}) \quad (2)$$

ただし、 $\mathbf{x} \in R^n, y \in R, u \in R$ であり、 $\mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{x}), h(\mathbf{x})$ は \mathbf{x} に関して解析的であると仮定する。厳密な状態空間線形化法とは(1)のシステム座標変換と状態フィードバック

$$\mathbf{z} = \mathbf{T}(\mathbf{x}), \quad \mathbf{z} \in R^n \quad (3)$$

$$\mathbf{u} = \boldsymbol{\alpha}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x})v \quad (4)$$

§ MathematicaはWolfram Research, Inc.の商標

により、図1の外部入力 v と状態 z との関係をBrunovskyの正準形

$$\dot{z} = Az + bv \tag{5}$$

$$A = \begin{pmatrix} 0 & 1 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & 1 \\ 0 & 0 & \dots & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

に等価変換することである。

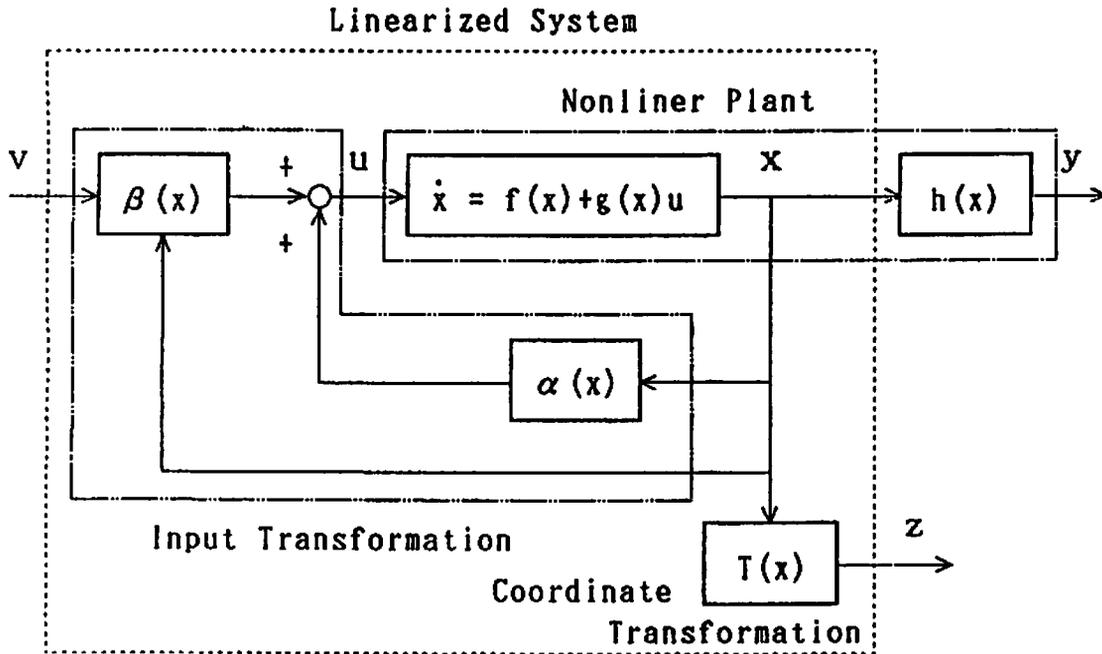


Fig.1 状態空間線形化法

[定理1] 非線形系(1)に対して厳密な状態空間線形化問題が可解であるための必要十分条件は

- (i) 可制御性行列 $[g(x), ad_f g(x), \dots, ad_f^{n-2} g(x), ad_f^{n-1} g(x)]$ がランク n であることと、
- (ii) ベクトル場の集合 $\{g(x), ad_f g(x), \dots, ad_f^{n-2} g(x)\}$ がインボリュートイブであることと。

(証明) 文献[2]参照

この時、変換(3),(4)式は次式によって決定される。

$$T(x) = col[\lambda(x), L_f \lambda(x), \dots, L_f^{n-1} \lambda(x)]^T \tag{6}$$

$$\alpha(x) = \frac{-L_f^n \lambda(x)}{L_g L_f^{n-1} \lambda(x)}, \quad \beta(x) = \frac{1}{L_g L_f^{n-1} \lambda(x)} \tag{7}$$

ただし $\lambda(\mathbf{x})$ は次の関係を満たす関数である。

$$L_g \lambda(\mathbf{x}) = L_{ad,fg} \lambda(\mathbf{x}) = \dots = L_{ad,fg}^{n-2} \lambda(\mathbf{x}) = 0 \tag{8}$$

2.2 拡張線形化法

近似線形化の 1 つである本手法は次の SIMO の非線形系

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u), \quad \mathbf{f}(0, 0) = 0 \tag{9}$$

に対して、制御則

$$u = v - k(\mathbf{x}), \quad k(\mathbf{x}) : R^n \rightarrow R, \quad k(0) = 0 \tag{10}$$

を用いた閉ループ系を平衡点の集合上の任意の点 $\{(\mathbf{x}_s, u_s) \mid \mathbf{f}(\mathbf{x}_s, u_s) = 0\}$ まわりでテーラー展開した近似線形系

$$\dot{\mathbf{x}} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_s, u_s) - \frac{\partial \mathbf{f}}{\partial u}(\mathbf{x}_s, u_s) \frac{\partial k}{\partial \mathbf{x}}(\mathbf{x}_s) \right] \mathbf{x} + \frac{\partial \mathbf{f}}{\partial u}(\mathbf{x}_s, u_s) v \tag{11}$$

の固有値が一定となるような解析的フィードバック $k(\mathbf{x})$ を見つける問題である。

閉ループ線形系(11)の固有値が任意の指定された値を持つような解析的フィードバック $k(\mathbf{x})$ が存在するための条件は $\left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_s, u_s), \frac{\partial \mathbf{f}}{\partial u}(\mathbf{x}_s, u_s) \right]$ が可制御であることである。この時、 $k(\mathbf{x})$ は次式により求められる。

$$k(\mathbf{x}) = k'(x_{1s}^{-1}(x_{1s})) + \sum_{j=2}^n \frac{\partial k}{\partial \mathbf{x}}(x_{1s}^{-1}(x_{1s})) [x_j - x_{js}(x_{1s}^{-1}(x_{1s}))] \tag{12}$$

ただし、 $\mathbf{x}_{js}(u_s)$ ($j = 1 \sim n$) は平衡点の集合を $u_s \in R$ をパラメータとして表現したものであり、 $k'(u_s)$ は $\frac{\partial k}{\partial \mathbf{x}}(u_s) \frac{\partial \mathbf{x}}{\partial u}(u_s)$ を u_s で積分したものである。

証明は文献 [6] を参照。

その他、動的フィードバック線形化、疑似線形化などがあるが紙面の関係で省略する。表 1. 参照。

3 Mathematicaを用いた設計支援システム

3.1 設計支援システムの特徴

システム設計者が制御系設計を行うため、計算機を援用する際に求められる一般的機能を大別すると、

1. 制御系設計の自動化アルゴリズムの計算実行
2. man-machine インターフェイス

3. 数値シミュレーション

4. システムの拡張性

などであり、これらが互いに協調して動作することが重要である。一般に、数式処理ソフトを用いた設計支援システムではシミュレーションのような数値計算の処理速度を上げるためACSL, Fortranなどのコンパイラを併用することが多いが^{[8],[9]}、Mathematicaでは数値計算の高速処理も考慮されているため、同一言語による統合的な環境が作りやすい。実際に Mathematica は数値計算/数式処理/グラフィックスのすべてに強力であり、800以上の組み込み関数を持っている。さらに各種のスタイルでプログラミングが可能であり、数式処理機能が必要不可欠な非線形系の設計支援システムの構築に最適である。

本設計支援システムの特徴は

〔特徴1〕 システムの拡張性および移植性を考え、1.~4.の機能をすべて Mathematica の関数群により実現する。また、線形化制御則の数式を Mathematica の組み込み関数 **TeXForm** を用いて変換し、**L^AT_EX** およびそのプレビューア(xtex)を用いて、見やすい形で表示する機能を持たせた。

〔特徴2〕 各種線形化法の条件の判定、制御則の導出に至る複雑な計算を Mathematica の数式処理機能を用いて自動化した。これにより複数の手法による制御系設計ができ、その制御性能の比較検討が容易に行える。また各設計法には自由度があるため、これらは設計者が選択しコマンドラインから入力する対話形式での設計法を採用した。

〔特徴3〕 設計された制御系が動作点変動やパラメータ変化によりどのような影響を受けるかをあらかじめ解析する機能を持たせた。先に著者らはいくつかの数値例で調べることを行ったが^[7]、ここでは計算機的能力をフルに生かして極位置の変動範囲を計算、表示する方法を4.4.1で示す。

3.2 設計支援システムの構成と実行手順

本設計支援システムの構成は、あるデータ構造のルールに従ってアプリケーションモデルを記述する定義部(Model definition)、モデルの予備解析を行う解析部(Analysis)、各種線形化法のアルゴリズムを納めた設計部(Design)、シミュレーションなどを行う表示部(Graphics)、各部で共通に用いるライブラリ(Math Library)から成る。各部はそれぞれファイルとして登録され、階層構造を持つ関数の集合体(ユーザ定義関数)で構成される。各部はファイルをロードすれば、独立に実行でき機能の追加、修正はこのユーザ定義関数単位で行われる。図2に本システムの階層構造を示す。以下に定義部の例を、また表1にライブラリ、解析部、設計部の主要なユーザ定義関数とその機能を示す。

【定義部】 ...cadmodels.m

```

MBoil 2 [title] = " 2-Vessels Thermal System ";
MBoil 2 [mtype] = 1;
MBoil 2 [n] = 2;
MBoil 2 [f] = {{-(a1 + b1) x[1] + a1 x[2] + b1 theta1},
              {a2 x[1] + a2 x[2]}};
MBoil 2 [g] = {{0}, {(theta2i - x[2])/C2}};
MBoil 2 [h] = x[1];
MBoil 2 [param] = {a1-> 0.1, a2-> 0.05, b1-> 0.05,
                  theta1-> 20, theta2i-> 80, C2-> 40};
    
```

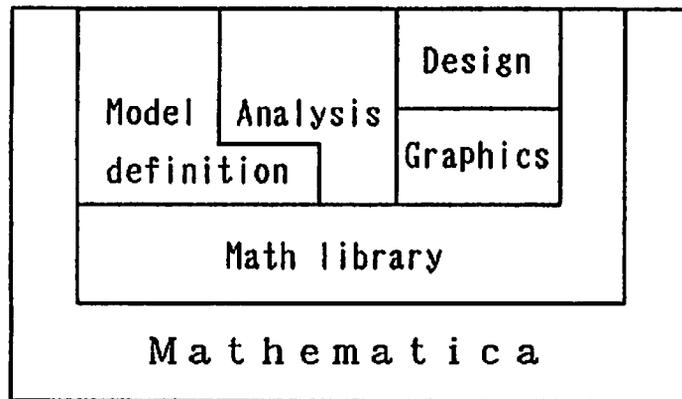


Fig.2 階層構造

File name	Function name	Output result
libMath.m (Math library)	Jac[f,x,n,m]: LieD[f,h,x,n]: LieB[f,g,x,n]: Involutive[mat]: FullRankQ[mat]:	Jacobian matrix of a vector field "f" Lie derivative of "h" with respect to "f" Lie bracket of "f" and "g" Involutivity check of a matrix "mat" Rank check of a matrix "mat"
cadAnalysis.m (Analysis)	AEquilib[model,us]: APole[model,us]: ANonlinearCO[model]: ALinearize[model,wxs]: ARobust[model]:	Equilibrium point at the input "us" Openloop poles at the input "us" Controllability matrix and observability matrix of the nonlinear system Linearized system about the operating point "wxs" Equilibrium points and closedloop poles by the changes of system parameters
cadDExact.m cadDlo.m cadDDyna.m	DExact[model]: Dlo[model]: DDynamic[model]:	Design of state space linearization method ⁽²⁾ Design of input-output linearization method ⁽³⁾ Design of dynamic feedback linearization method ⁽⁴⁾
cadDPseudo.m cadDExtend.m (Design)	DPseudo[model]: DExtend[model]:	Design of pseudo linearization method ⁽⁵⁾ Design of extended linearization method ⁽⁶⁾

Table 1 ユーザ定義関数とその機能

モデルシンボル "MBoil 2", 次元 n , 式(1), (2)の f, g, h , パラメータの値などを定義している。

表示部にはシミュレーション関数を有している。次に本設計支援システムの実行手順の流れを図3に示す。まず、データ構造のルールにしたがってテストするモデルを定義する。解析部によりモデルの非線形性を解析し、設計部から線形化法を選択し線形化制御則の決定、及び外部コントローラ (I, PIコントローラなど) を決定して閉ループ系を構成する。

この時、プログラムの修正に伴う再計算を最少限にするため、解析結果、設計結果はそれぞれファイルに保存される。

パラメータ変化における影響を調べるため閉ループ系を解析し、必要なパラメータの値、状態量の初期値などを設定してシミュレーションを行う。

なおシステムの起動はコマンドラインから `<< cadStart.m` と入力する事により、必要なファイルをロードして使用する。

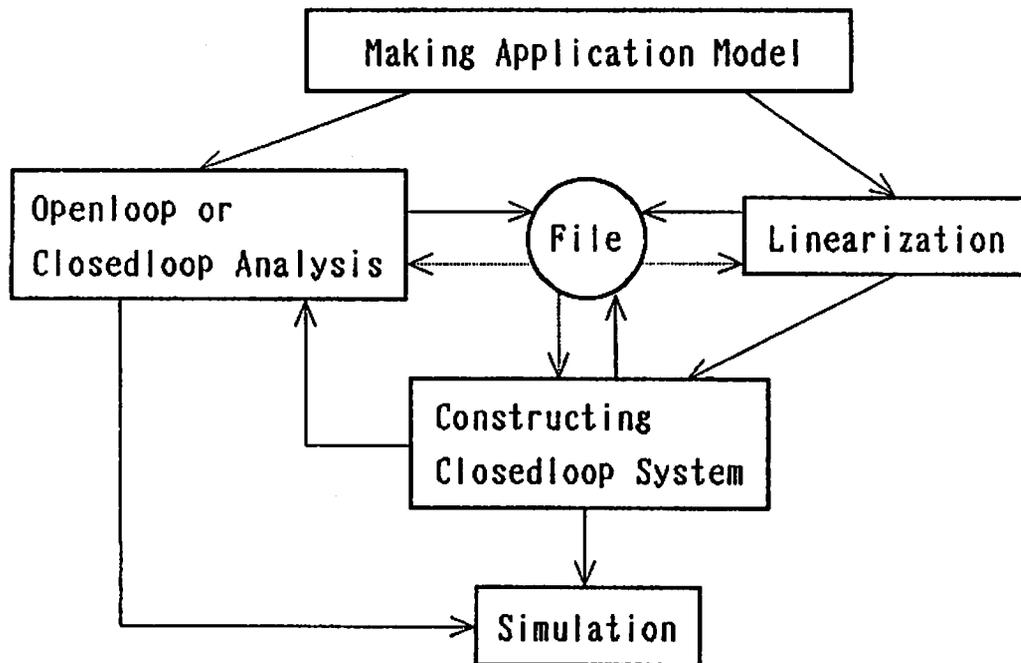


Fig. 3 実行手順

4 線形化と制御系構成例による数式処理CAD

4.1 双線形系としての制御対象

先に著者らが手計算で行った例^[7]を壁の熱容量を省略した2次の双線形系として図4に示す。この2重缶式加熱器の制御系を本研究のCADで構成、解析して示す。第1、第2流体の熱収支から次式が得られる。

$$\begin{aligned}
 C_1 \frac{d\theta_1}{dt} &= q_1(\theta_{1i} - \theta_1) - kA(\theta_1 - \theta_2) \\
 C_2 \frac{d\theta_2}{dt} &= q_2(\theta_{2i} - \theta_2) + kA(\theta_1 - \theta_2)
 \end{aligned}
 \tag{13}$$

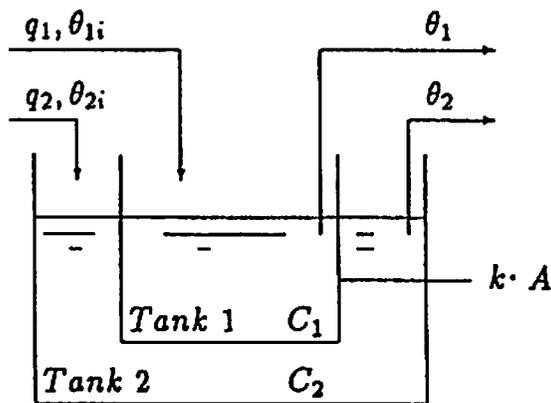


Fig.4 2重缶式加熱器

C : 熱容量 [$kJ/^\circ C$], k : 熱通過率 [$kJ/(m^2 \cdot ^\circ C \cdot s)$], θ : 流体温度 [$^\circ C$],
 q : 熱流入量 [$kJ/(^\circ C \cdot s)$], A : 熱移動面積 [m^2], θ_i : 流体の入口温度 [$^\circ C$]
 状態変数 $x = [\theta_1, \theta_2]$ 、操作量 $u = q_2$ 、制御量 $y = \theta_1$

とすると次の2次双線形系が得られる。

$$\begin{aligned}
 \dot{x}_1 &= -(a_1 + b_1)x_1 + a_1x_2 + b_1\theta_{1i} \\
 \dot{x}_2 &= a_2x_1 + a_2x_2 + \frac{\theta_{2i} - x_2}{C_2}u \\
 y &= x_1
 \end{aligned}
 \tag{14}$$

ただし、 $a_1 = k \cdot A/C_1, a_2 = k \cdot A/C_2, b_1 = q_1/C_1$ とした。このシステムは状態 x_2 と入力 u の積の項が出現する双線形系である。

4.2 CADによる制御対象システムの初期解析

本設計支援システムは定義された制御対象システムの数学モデルに対して、まずモデルの動特性を表示し、基準の操作量を用いたときの平衡点とその点まわりでテーラー展開した近似線形系の開ループ極を計算して表示する。またキー操作によってメニュー

- (1) モデルの図示…(a)
- (2) ベクトル場…(b)
- (3) Bode線図 (ゲイン曲線) …(c)
- (4) Bode線図 (位相曲線) …(d)

から選択して、表示させる解析機能を持たせた。1~4のテンキーを押せば、これらの図を表示することができる。4.1の例題に対して(1)~(4)を表示させた本設計支援システムの初期解析画面を図5に示す。このメニューから抜けて先に進むときは(5) Continue キーを押せば良い。そして各線形化法を選択して、そのコマンド(表1)を入力する。

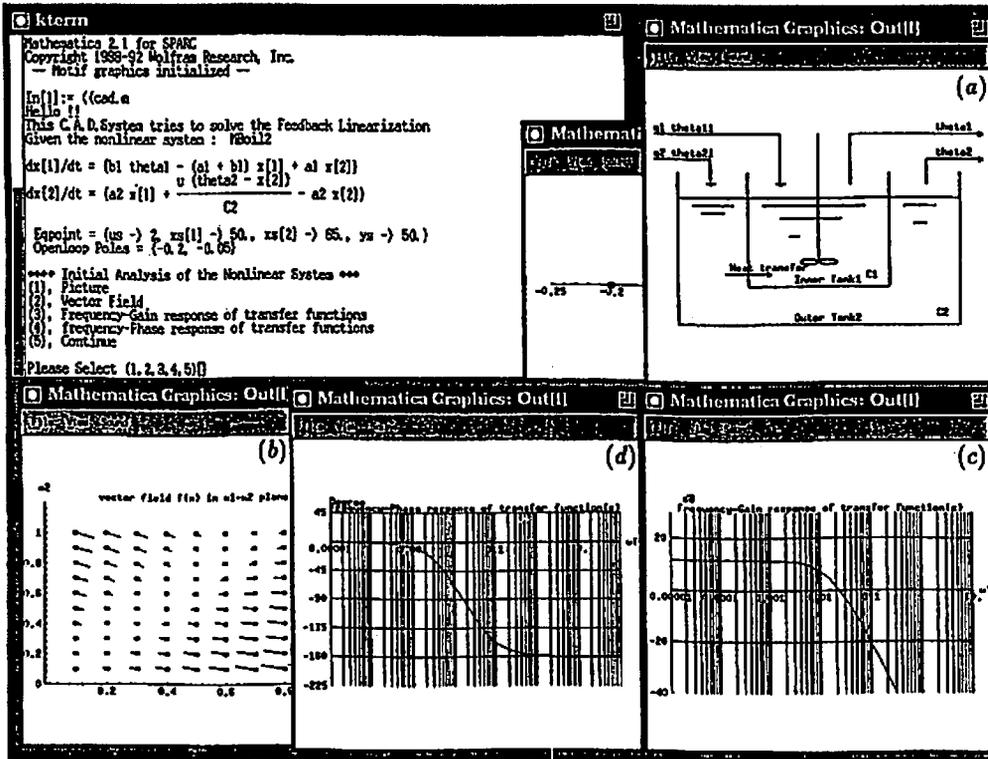


Fig. 5 CADによる初期解析画面

4.3 フィードバック線形化法による制御系設計

4.1の例題に対し、2.1で示した厳密な状態空間線形化法の本設計支援システムによる適用を試みる。図6に状態空間線形化法の対話式設計例を示し、その設計手順を説明する。

```

In[2]:= DExact[model]
State Space Linearization Method
The conditions are controllability and involutivity.
(a) Controllability Check
          a1 (theta2t - x[2])
Controllability Matrix = {{0, -(-----)},
                          C2
> {-----, -----, -----}}
          C2          C2          C2
          2
Determinant = -----
                C2
This matrix is FullRank!!
...Press any key...
(b) Involutivity Check
satisfied, because n <= 2
These conditions are satisfied.
Consequently, state and input transformation exists.
...Press any key...

(1) Input lambda from keyboard (2) Frobenius Theorem ? = 1
lambda(x) = x[1]
    
```

```

Checking the partial differential eqotions for lambda!!
Lad0g lambda = (0)
          a1 (theta2i - x[2])
Lad1g lambda = {-(-----)}
                  C2
Relative degree of MBo112 for function x[1] = 2
...Press any key...
(1) State Transformation
z[1]=((x[1]))
z[2]=((b1 thetai - (a1 + b1) x[1] + a1 x[2]))
Output equation of z coordinate is ...
y = z[1]
...Press any key...
(2) State Feedback
      C2 v
u = {----- - (C2
      a1 (theta2i - x[2])
>      (-(a1 + b1) (b1 thetai - (a1 + b1) x[1] + a1 x[2])) +
>      a1 (a2 x[1] - a2 x[2]))} / (a1 (theta2i - x[2]))
    
```

Fig. 6 状態空間線形化法の対話式設計例

(a)…2.1 [定理 1] の(i)可制御性条件を調べるため、可制御性行列を表示し、その行列式を計算してフルランクになるかどうかを確認する。これにより、特異点（線形化できない点）もわかる。

(b)… [定理 1] の (ii)インボリューティブ性を調べるため、ベクトル場の集合 $[g(x), ad_f g(x), \dots, ad_f^{n-2} g(x)]$ から考え得るすべてのリー括弧積を計算し、それぞれについて元のベクトル場の集合と線形従属になるかどうかを計算する。すべてのリー括弧積が線形従属となれば、インボリューティブであると判定させるようにした。2次系では常にインボリューティブ性が成り立つので、上記の計算を実行することなく satisfied, because $n \leq 2$ と表示させるようにしている。

(c)…(8)式を満たす関数 $\lambda(x)$ を決定する。(8)式は偏微分方程式であるが、その可解性は Frobeniusの定理から、[定理 1] (ii)のベクトル場の集合がインボリューティブである必要がある^[1]。この時、偏微分方程式の求解を常微分方程式の求解に帰着させることができ、このアルゴリズムを用いて $\lambda(x)$ をCAD上において自動的に導出できるようにした。(8)式は $\lambda(x)$ を $n-1$ 回時間微分しても u の項が現れないという条件であり、このような関数が容易に見出せる場合には、 $\lambda(x)$ をキーボードから入力し、その関数が(8)式を満たすかどうかを検算することもできるようにした(例題はこの場合に相当)。

(d)…関数 $\lambda(x)$ が決定したので、(6)式から座標変換、(7)式から状態フィードバックを計算して、表示する。状態空間線形法は出力方程式(2)も線形に変換されとは限らないため、出力フィードバック制御系を設計するためには 座標系での出力方程式も求める必要がある。このため、座標変換の逆写像を計算して z 座標での出力方程式も計算、表示するようにしている。

以上、本設計支援システムを用いれば Mathematica の数式処理機能により、線形化条件の判定、制御則の導出が自動化され高次系への設計にかかる負担も大きく軽減することができる。

4.4 シミュレーション

本設計支援・解析システムを用いて例題に状態空間線形化法を適用した非線形制御系 (図7-1) を構成した際のパラメータ変化による影響を極位置変動による解析とステップ応答から調べる。なお、基準の動作点まわりでテーラー展開した近似線形モデルに基づく従来の線形制御系 (図7-2) も構成して、比較検討を行う。

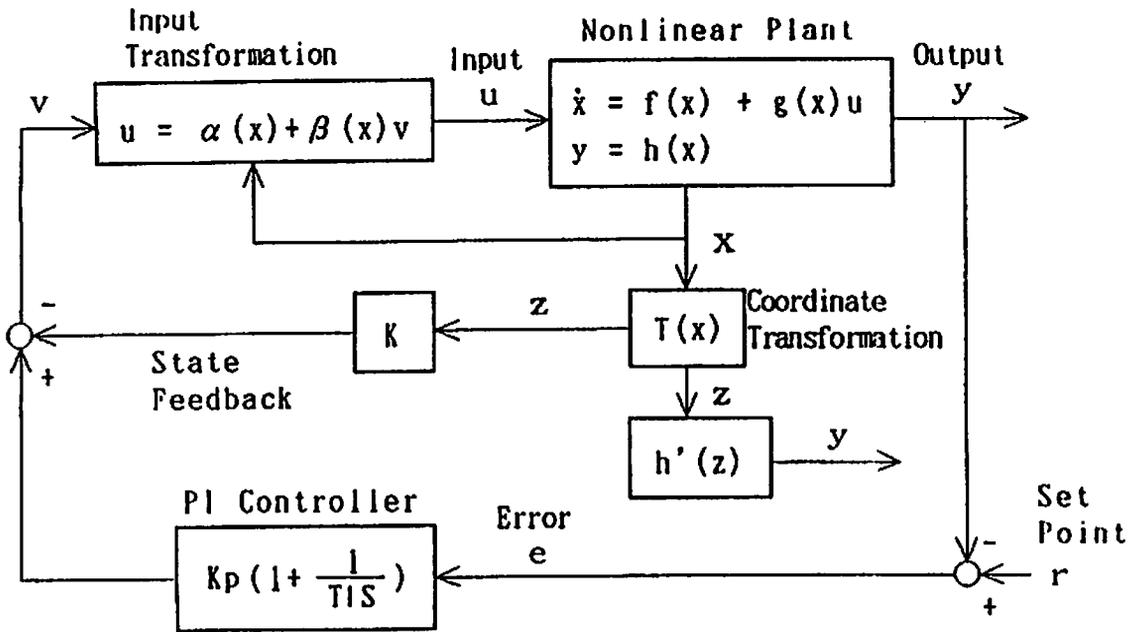


Fig. 7-1 状態空間線形化した非線形制御系

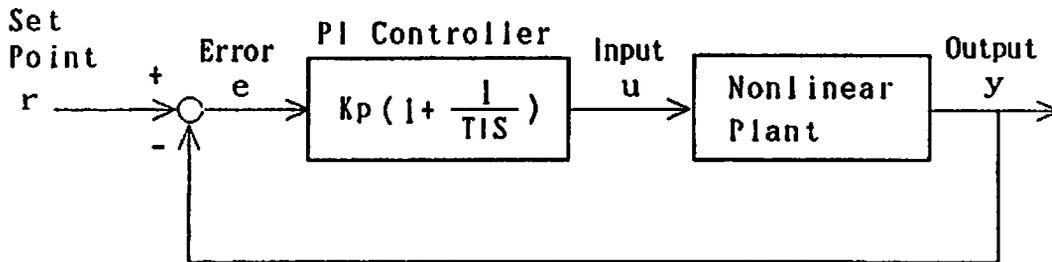


Fig. 7-2 近似線形化した対象に対する線形制御系

両制御系のシミュレーションの設定条件は

- パラメータは以下のように設定した。
 $\theta_{1i}, \theta_{2i} : 20, 80 [^{\circ}C], \quad q_1 : 1 [kJ/(^{\circ}C \cdot s)] \quad C_1, C_2 : 20, 40 [kJ/^{\circ}C],$
 $A : 1 [m^2], \quad k : 2 [kJ/(m^2 \cdot ^{\circ}C \cdot s)]$
- 基準の動作点は
 $\{x_1 = 50 [^{\circ}C], x_2 = 65 [^{\circ}C], u = 2 [kJ/(^{\circ}C \cdot s)]\}$ とした。

- 両制御系とも制御器はPIコントローラを用い、閉ループ系の極、零点が以下のような配置になるように制御器パラメータを調整した。

極： $\{-0.05, -0.1 \pm 0.1j\}$, 零点： $\{-0.05\}$

4.4.1 極位置変動による解析

図8、図9は熱通過率 $k : 2 [kJ/(m^2 \cdot ^\circ C \cdot s)]$ が1.2~2.8まで、また負荷 $q_1 : 1 [kJ/(^\circ C \cdot s)]$ が1~0.5まで変動した時の非線形制御系(図7-1)と線形制御系(図7-2)の極位置の変動範囲を表示したものである。この極はパラメータ k を $0.1 [kJ/(m^2 \cdot ^\circ C \cdot s)]$ ずつ変動させ、その各点において q_1 を $0.05 [kJ/(^\circ C \cdot s)]$ きざみで変動させて平衡点をそれぞれ計算し、その平衡点まわりでテーラー展開して極を計算、表示したものである。図8から非線形制御系では複素極 $-0.1 \pm 0.1j$ が $-0.06 \sim -0.13$ までほぼ左右に広がっているが、支配極 -0.05 がほとんど移動しないため、パラメータ変化の動作点移動による特性劣化は少ない。一方、図9の線形制御系では複素極 $-0.1 \pm 0.1j$ が実軸 $-0.04 \sim -0.16$ 、虚軸 $0.18j \sim -0.18j$ の間で大きく移動し、支配極 -0.05 も原点にかなり近い位置 -0.01 まで移動して非常に遅いモードとなる場合があり、特性劣化が著しいことが分かる。

この様に本設計支援システムにはパラメータ変化の大きさをおさえ、その範囲の特性変動を定量的に評価するための極位置変動範囲を表示させる機能を持たせた。これは、各手法のロバスト性を調べる上で非常に有効である。

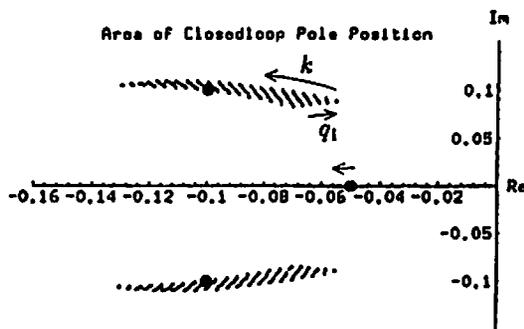


Fig. 8 非線形制御系の極位置変動範囲

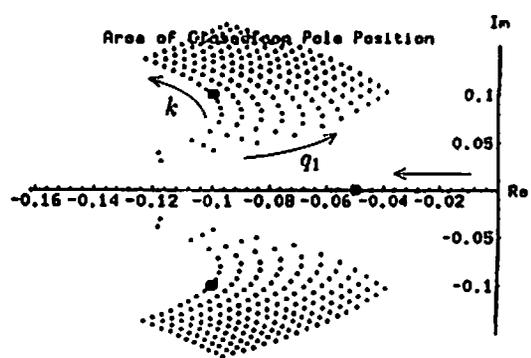


Fig. 9 線形制御系の極位置変動範囲

● Assigned poles

4.4.2 時間応答

図10はパラメータ(1) $k : 2 [kJ/(m^2 \cdot ^\circ C \cdot s)]$ を(2)2.8, (3)1.2に変動させ、平衡状態に達した後の両制御系のステップ応答、(1)', (2)', (3)' はそれぞれの平衡点での閉ループ極である。(1), (2), (3)から、線形PIのみの場合に比べ状態空間線形化法を利用した制御系の応答はかなり良好な応答となっていることが分かる。(1)', (2)', (3)', を見ても、(2)' では複素極の影響から線形PIのみの制御系はやや振動的に、(3)' では原点近くに移動した極の影響で遅い

応答となっていることが説明できる。

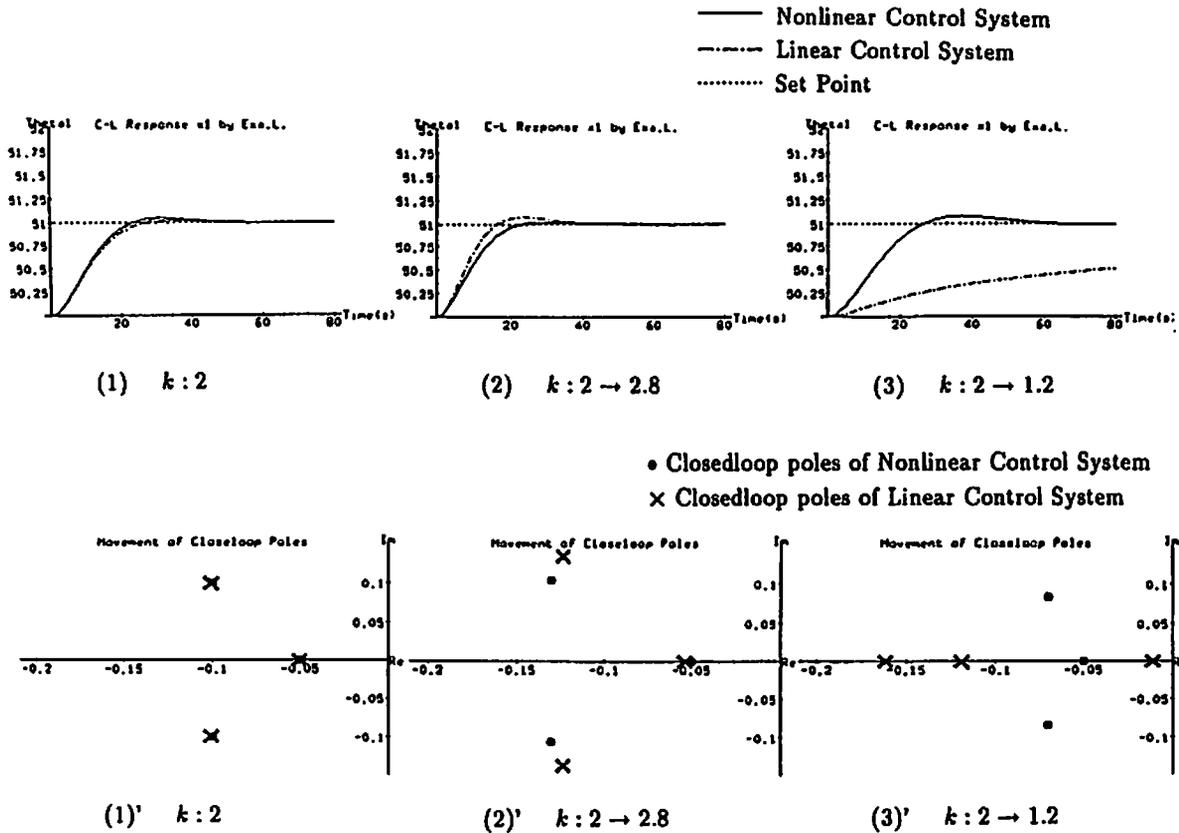


Fig.10 ステップ応答と閉ループ極

5 結言

本研究は Mathematica の優れた数式・数値処理、グラフィクス機能に着目し、非線形系に対して各種線形化法の条件判定及び制御則の導出計算を自動化する設計支援・解析システムを開発した。この設計支援・解析システムは (a) 拡張性に優れた統合的環境 (b) 複数の線形化手法による制御系設計 (c) パラメータ変化に対するロバスト性解析機能 などの特徴を有している。従って本システムを用いれば、線形化条件の判定、特に高次系への設計、解析、異なる線形化手法を用いた制御系間の比較検討が容易に行える。例題として、熱系に対し本システムを用いて状態空間線形化法による非線形制御系と従来の線形制御系とを構成してパラメータ変化による影響を調べたところ、前者の方がある種のパラメータ変化による特性変動の影響をうまく補償することがシミュレーションにより確認された。今後の方針として

1. 個別に設計した関数郡を機能別にパッケージ化すること。
2. 設計計算の途中で設計者の意図が反映できるような設計環境を実現すること。
3. インタフェースを充実させ、より使い易いシステムにすること。
4. 多入力系を扱えるようにすること。

などが課題である。

参考文献

- [1] Isidori, A., *Nonlinear Control Systems: An Introduction*, Lecture Note in Control and Information Science, 72, (1985), 2nd Ed. (1989), Springer-Verlag.
- [2] Hunt, L.R., Su, R., Meyer, G., *Design for multi-input Nonlinear Systems*, R.W. Brockett, R.S. Millman, H.J. Sussmann eds., *Differential Geometric Control Theory*, (1983), 268, Birkhauser.
- [3] Kravaris, C., Chung, C.B., *Nonlinear State Feedback Synthesis by Global Input/Output Linearization*, AIChE Journal, 33-4(1987), 592.
- [4] Fliss, M., *Generalized Controller Canonical Forms for Linear and Nonlinear Dynamics*, IEEE Trans. Automat. Control, 35-9(1990), 994.
- [5] Reboulet, C., Champetier, C., *A New Method for Linearizing Nonlinear Systems: the Pseudolinearization*, Int. J. Control, 40-4(1984), 631.
- [6] Baumann, W.T., Rugh, W.J., *Feedback Control of Nonlinear Systems by Extended Linearization*, IEEE Trans, Automat. Control, 31-1(1986), 40.
- [7] Fukuzawa, M., Ohtani, T., Masubuchi, M., *A Compensation Method for the Effects of the Change of Operating State in Bilinear Systems by State Linearization Technique*, Proceedings of IFAC Symp. on Design Methods of Control Systems, Zurich, Switzerland, (1991), 371.
- [8] Birk, J., Zeitz, M., *Computer Algebra for Analysis and Design of Nonlinear Control Systems*, American Control Conference(1987), 547.
- [9] Akhrif, O., Blankenship, G.L., *Computer Algebra for Analysis and Design of Nonlinear Control Systems*, American Control Conference(1987), 547.