数理解析研究所講究録
第1843巻 2013年 1-7

行列指数関数の Pade近似 について

北本 卓也
TAKUYA KITAMOTO*
山口大学
YAMAGUCHI UNIVERSITY

Abstract

本論文では、行列指数関数の Pade 近似、およびその制御系設計への応用について議論する。制御系が線形微分方程式で与えられた時、その解が行列指数関数を用いて書き表せることが知られているが、微分方程式にパラメータが含まれるときは行列指数関数はパラメータの複雑な関数となり、その解を制御系設計に用いることは容易でない。そこで、行列指数関数を Pade 近似を用いて有理関数化し、それを制御系設計に活用することを考える。もちろん、近似による誤差が発生するので誤差をなるべく抑える必要があるが、精度を上げると近似計算の結果も複雑になるので、誤差と計算結果の複雑さとの間でバランスをとる必要がある。

1 はじめに

制御系設計では、制御対象を微分方程式で表し、数式モデルをもとに制御器（コントローラー）を設計する。微分方程式として線形微分方程式を用いる場合は、制御対象を次の形に表すことが一般的である（本論文では1入力1出力のシステムを考える）。

$$\frac{dx}{dt} = Ax(t) + Bu(t), \quad y(t) = Cx(t) \tag{1}$$

ただし、x は n 次元ベクトル、A は $n \times n$ 行列、B は $n \times 1$ 行列、C は $1 \times n$ 行列である。

今、入力としてステップ入力、すなわち

$$u(t) = \begin{cases} 0 & (t < 0) \\ 1 & (t \geq 0) \end{cases} \tag{2}$$

としたとき、(1) の解 $y(t)$ は

$$y(t) = C(e^{At} - E)A^{-1}B \tag{3}$$

で与えられることが知られている。ここで、E は単位行列、e^{At} は行列 At の指数関数であり、

$$e^{At} = E + \frac{1}{1!}At + \frac{1}{2!}A^t t^2 + \cdots + \frac{1}{n!}A^n t^n + \cdots \tag{4}$$

で与えられる。(4) 式からわかるように、特に A がパラメータを含む時、e^{At} は複雑な形となり、パラメータと解の関係は簡単ではないが、本稿ではこの行列指数関数を Pade 近似することにより、$y(t)$ をパラメータの有理関数の形で近似する。また、本論文ではパラメータを k と置き、パラメータが1つの場合を考えるが、本論文の内容はパラメータが複数の場合にも拡張可能である。

*kitamoto@yamaguchi-u.ac.jp

2 行列指数関数の近似計算

行列指数関数の計算法は古くから研究されており、[4] には、19 もの計算法とその特徴が述べられている。本論文では、その中でも特に「固有値、固有ベクトルを用いた方法」と「Pade 近似を用いた方法」について注目する。

2.1 固有値、固有ベクトルを用いた方法

行列 At の固有値を $\lambda_1(t),\cdots,\lambda_n(t)$、固有ベクトルを $v_1(t),\cdots,v_n(t)$ とするとき、e^{At} は次のように表せる。

$$e^{At} = V(t)\begin{pmatrix} e^{\lambda_1(t)} & & & \\ & e^{\lambda_2(t)} & & \\ & & \ddots & \\ & & & e^{\lambda_n(t)} \end{pmatrix} V^{-1}(t), \quad \text{ただし、} V(t) = \begin{pmatrix} v_1(t) & v_2(t) & \cdots & v_n(t) \end{pmatrix} \tag{5}$$

上式の左辺と右辺は正確に等しく、式は近似式でなく等式であるが、行列 A がパラメータを含む場合には、固有値、固有ベクトルはパラメータの複雑な関数となるため、実際に計算するためには右辺の固有値、固有ベクトルをパラメータの打ち切りべき級数の形で計算する必要がある。ここに誤差が入るため、上式を用いて e^{At} を近似計算する時は誤差の影響を考慮する必要がある。

2.2 Pade 近似を用いた方法

この方法では、e^{At} を次式で近似する。

$$e^{At} \cong (E + \alpha_1 At + \cdots + \alpha_p A^p t^p)^{-1}\left(E + \beta_1 At + \cdots + \beta_l A^l t^l\right) \tag{6}$$

ただし、$\alpha_i, \beta_j \in \mathbf{R}$ は指数関数 e^t の Pade 近似における係数、すわなち

$$e^t = 1 + \frac{1}{1!}t + \frac{1}{2!}t^2 + \cdots + \frac{1}{n!}t^n + \cdots \cong \frac{1 + \beta_1 t + \cdots + \beta_l t^l}{1 + \alpha_1 t + \cdots + \alpha_p t^p} \tag{7}$$

を満たす実数とする。(6) の右辺は e^{At} の近似式であり、誤差を含んだものである（$p, l \in \mathbf{N}$ を大きくすれば、近似の精度を上げることは可能)。よって、(6) の右辺そのものに誤差が含まれるが、行列 A にパラメータが含まれていてもそのまま計算することが可能なので、右辺を計算する際に誤差が発生することはない。

2.3 数値実験

行列 A を次のようにおき、上の 2 つの方法で e^{At} を近似計算し、その誤差を評価した。

$$A = \begin{pmatrix} 0 & k+1 \\ -2 & -1 \end{pmatrix} \tag{8}$$

「固有値、固有ベクトルを用いた方法」では、打ち切りべき級数の次数を k, t の全次数を 10 に、「Pade 近似を用いた方法」では p, l をそれぞれ 3 に設定した（この場合、近似計算の結果における k, t の全次数は最大 9 次となる)。$k = 1$ における誤差を $0 \leq t \leq 3$ の範囲でプロットしたものを図 1 に示す。縦軸は行列

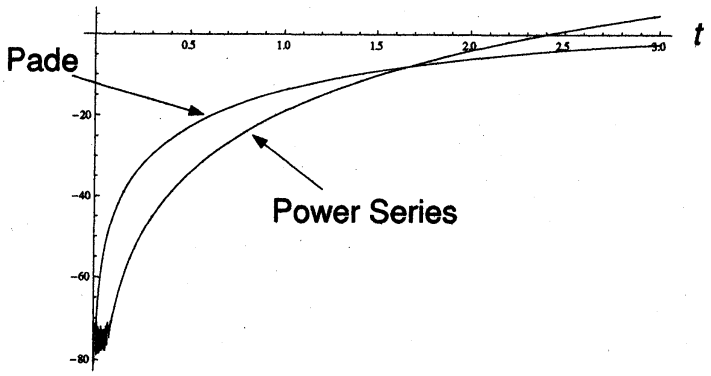

図 1: 近似計算の誤差

の各要素の誤差の２乗和の対数を取ったものであり（値が低いほど、精度が良い）、横軸は t（時間）である。図では、「Pade」で「Pade 近似を用いた方法」の誤差を、「Power Series」で「固有値、固有ベクトルを用いた方法」の誤差を表している。このグラフから次の傾向が読み取れる。

- 「固有値、固有ベクトルを用いた方法」は t が小さい所では、「Pade 近似による方法」より精度が高いが、t が大きくなる所では精度が逆転する。
- 高い精度が必要な場合は、「固有値、固有ベクトルを用いた方法」が有利。２，３桁の精度で十分な時は「Pade 近似による方法」が有利。

上のグラフは $k=1$ におけるものであるが、k の他の値においても同様の傾向がみられた。工学における応用においては、２，３桁の精度で十分な場合がほとんどなので、「Pade 近似による方法」の方が有利であると思われる。

3 制御系設計への応用

先に述べたように、(1) の解 $y(t)$ は (3) となるので、e^{At} が計算出来れば $y(t)$ は容易に求めることができる。行列 A,B,C の要素がパラメータ k の多項式となっている場合は $y(t)$ は k と t（時間）の有理関数で近似される。そこでこの $y(t)$ の有理関数近似を用いて、rise time, peaktime, peakvalue を計算する。rise time とは、$y(t)=y(\infty)$ を満たす最少の t (>0) である。また、peakvalue は $y(t)$ $(t>0)$ の最大値、peaktime はその時の t である（図２を参照）。

rise time は、$y(t)=y(\infty)$ を t について解けばよい。これは k,t の有理関数の方程式となるので、結局、k,t の多変数多項式の解を求めることになる。そこで、記号的ニュートン法を用いて、この多変数多項式の t に関する解をパラメータ k のべき級数の形で求め、さらに Pade 近似を用いて有理関数化すれば、rise time を k の有理関数の形で近似できる。

peaktime は $\frac{dy}{dt}=0$ を満たす t なので、$\frac{dy}{dt}=Ce^{At}B$ の分子部分（これは k,t の多変数多項式）の t に関する解を考えればよい。よって、rise time の時と同様にすれば、peaktime を k の有理関数で近似できる。また、この有理関数近似を t_0 とすると、peakvalue は $y(t_0)$ となるので、そのまま、peakvalue の有理関数近似を得る。

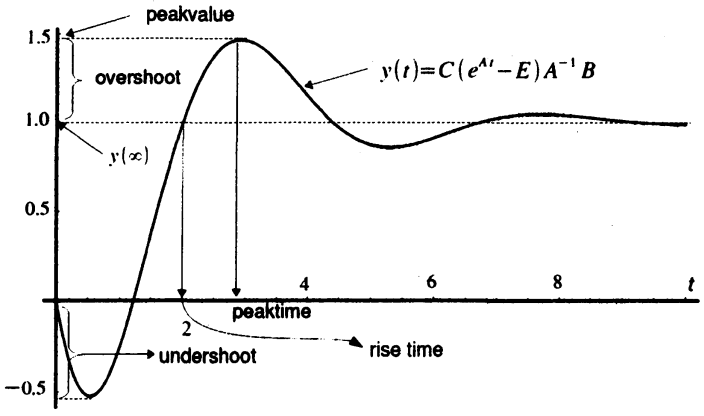

図 2: rise time, peaktime, peakvalue

3.1 アルゴリズム

以上より、$y(t)$, rise time, peaktime, peakvalue の計算アルゴリズムをまとめると、以下のようになる。

3.1.1 $y(t)$ の計算法

入力：(1) の行列 A, B, C
出力：$y(t)$ の有理関数近似

〈1〉 α_i, β_j を (7) が満たされるように決める。

〈2〉 $y(t)$ の有理関数近似を次式で求める。

$$y(t) \cong C\left\{\left(E+\alpha_1 At+\cdots+\alpha_p A^p t^p\right)^{-1}\left(E+\beta_1 At+\cdots+\beta_l A^l t^l\right)-E\right\}A^{-1}B \qquad (9)$$

3.1.2 rise time の計算法

入力：(1) の行列 A, B, C
出力：rise time の有理関数近似

〈1〉 「$y(t)$ の計算法」を用いて $y(t)$ の有理関数近似 $y(t) \cong q(k,t)/r(k,t)$ を計算する。

〈2〉 $w(k,t)$ を $w(k,t) = q(k,t) - y(\infty)r(k,t)$ で定義し、$w(k,t) = 0$ の t に関する根をべき級数の形で求め、Pade 近似を用いて有理関数化する。

3.1.3 peak time の計算法

入力：(1) の行列 A, B, C
出力：peak time の有理関数近似

〈1〉 α_i, β_j を (7) が満たされるように決める。

⟨2⟩ $\frac{dy}{dt}$ の Pade 近似を次式で計算する。

$$\frac{dy}{dt}(t) \cong C\left\{(E+\alpha_1 At+\cdots+\alpha_p A^p t^p)^{-1}\left(E+\beta_1 At+\cdots+\beta_l A^l t^l\right)\right\}B \tag{10}$$

⟨3⟩ 上式の分子を $q(k,t)$ と置き、$q(k,t)=0$ の t に関する根をべき級数の形で求めた後、Pade 近似を用いて有理関数化する。

3.1.4 peak value の計算法

入力：(1) の行列 A,B,C
出力：peak value の有理関数近似

⟨1⟩ 「peak time の計算法」を用いて、peak time の有理関数近似 $g(k)/h(k)$ を求める。

⟨2⟩ 「$y(t)$ の計算法」を用いて、$y(t)$ の有理関数近似 $p(k,t)/r(k,t)$ を求め、$p(k,g(k)/h(k))/r(k,g(k)/h(k))$ を peak value の有理関数近似として返す。

4 近似の改良の試み

4.1 基本的なアイデア

行列指数関数の Pade 近似による近似計算法について、「行列 At のノルム $|At|$ が小さければ、小さいほど計算精度が高い」ことがわかっている。容易にわかるように、任意の自然数 m に対して $e^{At}=\left(e^{\frac{At}{m}}\right)^m$ が成り立つので、行列 A がパラメータを含まない数値行列である場合は、計算精度を上げるために次のような方法がよく取られている。

⟨1⟩ $A_1 \leftarrow \frac{At}{m}$ とする。

⟨2⟩ e^{A_1} を Pade 近似を用いて計算し、M と置く。すなわち次のようにする。

$$M \leftarrow (E+\alpha_1 A_1+\cdots+\alpha_p A_1^p)^{-1}\left(E+\beta_1 A_1+\cdots+\beta_l A_1^l\right)^{-1} \tag{11}$$

⟨3⟩ $e^{At} \leftarrow M^m$ とする。

行列 A にパラメータが含まれる場合について考える。e^{At} の Pade 近似の精度を上げるためには、⟨1⟩ の m を大きくするか、あるいは ⟨2⟩ の p,l を大きくすれば良いが、どちらの方法も計算結果における k,t の次数を上げる。そこで、一定の精度を保ったまま、計算結果における k,t の次数をなるべく低くするにはどちらか有利かを検討する。

4.2 数値実験

4.2.1 数値実験 1

まず、A を (8) と置いてみる。$p=l=3,m=1$ と置いた場合と $p=l=1,m=3$ とした場合の $k=1.5$ における誤差を図 3 に示す（ともに k,t の全次数は 9 次である）。縦軸は誤差で、行列の各要素の誤差の 2 乗和の対数を取ったものである。横軸は時間 t である。図より、$p=l=3,m=1$ の方が誤差が少ない事がわかる。

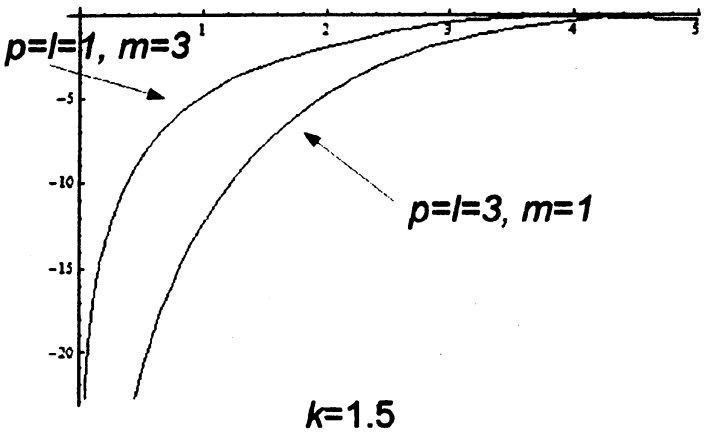

図 3: 近似の誤差 (1)

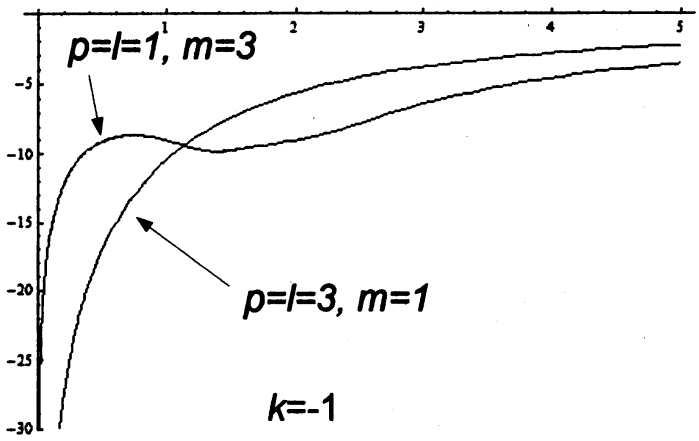

図 4: 近似の誤差 (2)

4.2.2 数値実験 2

次に行列 A を

$$A = \begin{pmatrix} k-1 & 1 \\ 2 & -3 \end{pmatrix} \tag{12}$$

と置いた時を考える。$p = l = 3, m = 1$ と置いた場合と $p = l = 1, m = 3$ とした場合の $k = -1$ における誤差を図 4 に示す（ともに k, t の全次数は 9 次である)。縦軸、横軸は図 3 と同じである。図より t の値によって、誤差が少ない方が変わってくる事がわかる。また、$p = l = 1, m = 3$ とした場合は t の値が大きいほうが誤差が少なくなっている所がある。

以上より、下記のことがわかる。

- p, l を上げた方が有利な場合も、m を上げた方が有利な場合も共にある。
- どのような場合にどちらを上げた方が有利かはまだはっきりしない。
- p, l を上げた時の方が、計算誤差が安定している。

5 結論

行列指数関数の計算法について述べた。行列がパラメータを含んでいる場合、べき級数展開すると展開点近傍では正確であるが、それから外れると急激に精度が落ちる。それほど計算精度が必要でない場合、Padeによる近似を用いた方が有利な場合もある。このPade近似による行列指数関数の近似を用いて、線形微分方程式を解を求め、制御系設計に応用した。行列指数関数の計算法については、数値計算で用いられる精度向上のテクニックがつかえる場合もあるが、どのような場合にそれが有効であるかは、更なる調査が必要である。

参考文献

[1] C. Abdallah, P. Dorato, W. Yang, R. Liska and S. Steinberg., "Application of Quantifier Elimination Theory to Control System Design," *Proc. of 4th IEEE Mediterranean Symposium of Control and Automation* pp. 340–345, Maleme, Crete, 1996.

[2] P. Dorato, W. Yang and C. Abdallah. "Robust Multi-Objective Feedback Design by Quantifier Elimination," *J. Symbolic Computation*, Vol. 24, pp. 153–159, 1997.

[3] I. A. Fotiou, P. Rostalski, P. A. Parrilo, and M. Morari. "Parametric optimization and optimal control using algebraic geometry methods," *International Journal of Control*, Vol. 79, No. 11, 1340-1358, 2006.

[4] C. Moler and C. Van Loan, "Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later," *SIAM Review*, Vol. 45, No. 1, pp. 3–49, 2003.

[5] T. Kitamoto, "Computation of the Peak of Time Response in the Form of Formal Power Series," *The IEICE Trans. Funda. (Japanese Edition)*, Vol. E86-A, No. 12, pp. 3240–3250, 2003.

[6] J. D. Lipson. "Newton's method: A great algebraic algorithm," *ACM Symposium on Symbolic and Algebraic Computations Proceedings*, pp. 260–270, 1976.

[7] T. Kitamoto "Accurate Computation of a High Degree Coefficient of a Power Series Root," *The IEICE Trans. Funda.*, Vol. E88-A, No. 3, pp. 718–727, 2005.

[8] T. Kitamoto "On computation of a power series root with arbitrary degree of convergence ," *Jpn J. Indus. Appl. Mathe.*, Vol. 25, No. 3, pp. 255–279, 2008.