数理解析研究所講究録
第1568巻 2007年 169-175

代数的 Riccati 方程式の定義多項式の計算法について

北本卓也
TAKUYA KITAMOTO
山口大学

山口哲
TETSU YAMAGUCHI
サイバネットシステム

1 序論

代数的 Riccati 方程式

$$PA + A^T P - PWP + Q = 0 \tag{1}$$

は H_2 制御、または H_∞ 制御系設計問題に用いられる非常に重要な方程式の1つであるが、A, W, Q がパラメータを含む場合を考えると、従来の標準的な数値的算法をそのまま適用することができない。(この方程式を解く標準的な方法は、固有ベクトル もしくは Schur 分解を数値的に計算し、それをもとに解を計算する方法である)。

A, W, Q がパラメータを含む場合には、代数的 Riccati 方程式 (1) の解はそもそも明示的な形で書き表せるとは限らない (アーベルの定理より5次以上の多項式の根は、明示的な形で表すことはできない)。しかしながら、その代数的 Riccati 方程式を根とする多項式 (定義多項式) は明示的な形で表すことが可能である。そこで本稿ではこの定義多項式の計算法について述べる。

この定義多項式は計算機代数で多用されるグレブナー基底を用れば、以下の方法で計算できる。例として A, W, Q が

$$A = \begin{bmatrix} k & 1 \\ 1 & -1 \end{bmatrix}, \ W = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \ Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

で与えられた場合を考えると、まず、(1) の解 P を

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} \\ p_{1,2} & p_{2,2} \end{bmatrix}$$

とおく。(1) の (1,1),(1,2),(2,2) 要素より連立代数方程式

$$\begin{cases} 2kp_{1,1} + 2p_{1,1}^2 + 2p_{1,2} - 2p_{1,2}^2 + 1 = 0 \\ p_{1,1} - p_{1,2} + kp_{1,2} - 2p_{1,1}p_{1,2} + p_{2,2} - 2p_{1,2}p_{2,2} = 0 \\ 2p_{1,2} - 2p_{1,2}^2 - 2p_{2,2} - 2p_{2,2}^2 + 1 = 0 \end{cases}$$

を得るので、この多項式系に対して項順序を $p_{1,1} \succ p_{1,2} \succ p_{2,2}$ としたの辞書式順序のグレブナー基底を計算すると

$$f_4(k)p_{2,2}^4 + f_3(k)p_{2,2}^3 + f_2(k)p_{2,2}^2 + f_1(k)p_{2,2} + f_0(k) = 0$$

を得る。これが (1) の解 P の (2,2) 成分の定義多項式である。ただし

$$\begin{aligned} f_4(k) &= 4k^3 + 4k^2 + 12k - 20, \\ f_3(k) &= 8k^3 + 8k^2 + 24k - 40, \end{aligned}$$

$$
\begin{aligned}
f_2(k) &= -4k^2+8k-4, \\
f_1(k) &= -4k^3-8k^2-4k+16, \\
f_0(k) &= k^3+3k^2-3k-1
\end{aligned}
$$

である。

同様にして、P の (1,1) 成分、(1,2) 成分の定義多項式を計算することができる。このグレブナー基底による定義多項式の計算法は理論的には簡単であるが、A, W, Q の行列のサイズ n に対し、変数の数が $\frac{n(n+1)}{2}$ となるため、比較的小さい n に対しても実用的ではない。そこで本稿では、この定義多項式を効率的に計算する方法について述べる。

2 Riccati 方程式の解の行列式表示

Riccati 方程式の解を行列式を用いて表す方法について述べる。これは、パラメータを残したまま計算する方法 [2] を応用したものである。この方法では、はじめに

$$
H = \begin{bmatrix} A & -W \\ -Q & -A^T \end{bmatrix} \tag{2}
$$

の固有ベクトルを固有値 λ を未定変数として残してを求める。

アルゴリズム 1 $v(\lambda)$ の計算

⟨1⟩ x を $x = \begin{bmatrix} x_1 & \cdots & x_{2n} \end{bmatrix}^T$ と置き、$2n$ 個の線形方程式 $(H-\lambda E)x = 0$（ベクトル $(H-\lambda E)x=0$ の要素 1 つ 1 つが 1 つの線形方程式）を構成する（ここで λ は不定元である）。

⟨2⟩ ⟨1⟩ の $2n$ 個の線形方程式より $(2n-1)$ 個の線形方程式を選び、変数 $x_1, \cdots, x_{2n-1}$ の方程式として解く。

⟨3⟩ 解いた $x_1, \cdots, x_{2n-1}$ を x に代入した後、x の各要素が多項式となるように x をスカラー倍する。

⟨4⟩ $v(\lambda) \leftarrow x/x_{2n}$ と置き、$v(\lambda)$ を出力する。

上のアルゴリズムで得られたベクトル $v(\lambda)$ は固有値 λ に対応する固有ベクトルを表している。ゆえによく知られているように $\bar{\lambda}_1, \cdots, \bar{\lambda}_n$ を H の実部が負である固有値 (n 個ある) とすると (1) の正定解 P は

$$
\begin{aligned}
P &= X_2 X_1^{-1}, \\
\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} &= \begin{bmatrix} v(\bar{\lambda}_1) & \cdots & v(\bar{\lambda}_n) \end{bmatrix}
\end{aligned}
$$

で与えられる。故に $y_1, \cdots, y_n$ を変数として $\Gamma_1(y_1, \cdots, y_n)$, $\Gamma_2(y_1, \cdots, y_n)$ を

$$
\begin{bmatrix} \Gamma_1(y_1, \cdots, y_n) \\ \Gamma_2(y_1, \cdots, y_n) \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} v(y_1) & \cdots & v(y_n) \end{bmatrix} \tag{3}
$$

$\Lambda(y_1, \cdots, y_n)$ を

$$
\Lambda(y_1, \cdots, y_n) \stackrel{\text{def}}{=} \Gamma_2(y_1, \cdots, y_n)\Gamma_1(y_1, \cdots, y_n)^{-1} \tag{4}
$$

で定義すると $\Lambda(y_1, \cdots, y_n) \in \mathbf{Z}[y_1, \cdots, y_n]^{n,n}$ は $\Lambda(\lambda_1, \cdots, \lambda_n) = P$ を満たす多項式行列となる。P の (i,j) 要素 $p_{i,j}$ を次のように表すことができる。

定理 1 P を H の固有値 $\lambda_1,\cdots,\lambda_n$ に対応する代数的 *Riccati* 方程式の解、すなわち $P=\Lambda(\lambda_1,\cdots,\lambda_n)$ とする。このとき、P の (i,j) 要素 $p_{i,j}$ は次のように表される。

$$p_{i,j}=\frac{|\bar{\Gamma}_{i,j}(\lambda_1,\cdots,\lambda_n)|}{|\Gamma_1(\lambda_1,\cdots,\lambda_n)|} \tag{5}$$

ここで $\bar{\Gamma}_{i,j}(\lambda_1,\cdots,\lambda_n)$ は以下で定義される行列である（ただし、$v_i(y)$ は アルゴリズム *1* で計算されたベクトル $v(y)$ の第 i 要素を表す）。

$$\bar{\Gamma}_{i,j}(\lambda_1,\cdots,\lambda_n)=\begin{bmatrix} v_1(y_1) & \cdots & v_1(y_n) \\ \vdots & \vdots & \vdots \\ v_{j-1}(y_1) & \cdots & v_{j-1}(y_n) \\ v_{n+i}(y_1) & \cdots & v_{n+i}(y_n) \\ v_{j+1}(y_1) & \cdots & v_{j+1}(y_n) \\ \vdots & \vdots & \vdots \\ v_n(y_1) & \cdots & v_n(y_n) \end{bmatrix} \tag{6}$$

□

3 基本的なアイデア

3.1 問題設定

$A\in\mathbf{Z}[k]^{n,n}$, $W\in\mathbf{Z}[k]^{n,n}$, $Q\in\mathbf{Z}[k]^{n,n}$ を k の多項式行列とする。このとき、(1) の正定対称解 P の (i,j) 要素 $p_{i,j}$ の定義多項式をそれぞれ求める。まず、次の補題に注意する。

補題 1 H の固有値 $\lambda_1,\cdots,\lambda_n$ に対応する *Riccati* 方程式を $P(\lambda_1,\cdots,\lambda_n)$ とする。次の 2 つの条件

(C1) H の固有値は全て相異なる。

(C2) $\lambda_i+\lambda_j=0\ \ (i\neq j)\Rightarrow P(\lambda_1,\cdots,\lambda_n)$ が対称行列でない。

を満たす k の値 k_0 が存在するならば、有限個の k の値を除き、*Riccati* 方程式 *(1)* の対称解は 2^n 個存在する。□

この補題 1 より次の定理が成り立つ。

定理 2 補題 *1* の仮定が満たされるとする。$p_{i,j}$ を正定対称解 P の (i,j) 要素とするとき、$p_{i,j}$ を根とする多項式で

$$f_{2^n}(k)p_{i,j}^{2^n}+\cdots+f_1(k)p_{i,j}+f_0(k)=0, \tag{7}$$

$$f_l(k)\in\mathbf{Z}[k]\ \ (l=0,\cdots,2^n) \tag{8}$$

となるものが存在する。□

$p_{i,j}$ の定義多項式は (7) を割り切るので、(7) の因子となっている。よって、(7) が計算できれば、それを因数分解することにより $p_{i,j}$ の定義多項式を求めることが可能である。よって以下では、(7) の多項式を求める算法について述べる。

3.2 多項式補間を用いたアルゴリズム

$f_{2^n}(k)$ を因子として持つ多項式

$$\phi(k) = f_{2^n}(k)\bar{f}(k) \ (\in \mathbf{Z}[k]) \tag{9}$$

が求まったとする。このとき、(7) の定義多項式が計算可能であることを示す。(7) の 2^n 個の根を $\alpha_l(k_r)$ $(l = 1, \cdots, 2^n)$ と書くと、(7) は

$$f_{2^n}(k_r)\{(p_{i,j} - \alpha_1(k_r)) \cdots (p_{i,j} - \alpha_{2^n}(k_r))\} \tag{10}$$

と書ける。ここで $\alpha_l(k_r)$ は $k = k_r$ における (1) の 2^n 個の対称解の (i,j) 成分なので、(1) に $k = k_r$ の代入を行い、$\alpha_l(k_r)$ を数値的に求めることが可能である。よって (9) の多項式 $\phi(k)$ が求まったとすると、(10) より

$$\begin{aligned}
&\phi(k_r)\{(p_{i,j} - \alpha_1(k_r)) \cdots (p_{i,j} - \alpha_{2^n}(k_r))\} \\
&= f_{2^n}(k_r)\bar{f}(k_r)\{(p_{i,j} - \alpha_1(k_r)) \cdots (p_{i,j} - \alpha_{2^n}(k_r))\} \\
&= \bar{f}(k_r)\left\{f_{2^n}(k_r)p_{i,j}^{2^n} + \cdots + f_1(k_r)p_{i,j} + f_0(k_r)\right\} \\
&= \sum_{l=0}^{2^n} \bar{f}(k_r)f_l(k_r)p_{i,j}^l
\end{aligned} \tag{11}$$

が計算可能である。ここで定数 $M \in \mathbf{N}$ を次式を満たす定数とする。

$$M > \max_l \left(\deg\left(\bar{f}(k)f_l(k)\right)\right) \tag{12}$$

(11) の $\bar{f}(k)f_l(k)$ は k の多項式なので M 個の点 k_r $(r = 1, \cdots, M)$ での値 $\bar{f}(k_r)f_l(k_r)$ が求まれば、これより多項式補間を用いて $\bar{f}(k)f_l(k)$ を求めることが可能である。$\bar{f}(k)f_l(k)$ が求まれば

$$\sum_{l=0}^{2^n} \bar{f}(k)f_l(k)p_{i,j}^l = \bar{f}(k)\left\{f_{2^n}(k)p_{i,j}^{2^n} + \cdots + f_1(k)p_{i,j} + f_0(k)\right\} \tag{13}$$

より $\sum_{l=0}^{2^n} \bar{f}(k)f_l(k)p_{i,j}^l$ を因数分解することで $f_l(k)$ $(l = 0, \cdots, 2^n)$ を計算し、定義多項式を求めることができる。

以上より、定義多項式を求める以下のアルゴリズムを得る。

アルゴリズム 2 定義多項式の計算

⟨1⟩ 式 *(9)* の $\phi(k)$ を求める。

⟨2⟩ $r = 1$ から $r = M$ *(式 (12))* まで以下の計算を行う。$k_r \in \mathbf{Z}$ を適当に取り、*(7)* の 2^n の根（すなわち *Riccati* 方程式 *(1)* の 2^n の対称解*)* を求め、*(11)* より $\bar{f}(k_r)f_l(k_r) \in \mathbf{Z}$ を計算する。

⟨3⟩ ⟨2⟩ の $\bar{f}(k_r)f_l(k_r) \in \mathbf{Z}$ $(r = 1, \cdots, M)$ より多項式補間を用いて $\bar{f}(k)f_l(k) \in \mathbf{Z}[k]$ を求める。

⟨4⟩ $\sum_{l=0}^{2^n} \bar{f}(k)f_l(k)p_{i,j}^l$ を因数分解し、*(13)* より $f_l(k)$ を求め、定義多項式を計算する。

(1) の 2^n 個の対称解は H の $s_1\bar{\lambda}_1, \cdots, s_n\bar{\lambda}_n$ $(s_1, \cdots, s_n = \pm 1)$ の固有値に対応するので、**定理 1** より 2^n 個の対称解の (i,j) 成分は

$$\frac{|\bar{\Gamma}_{i,j}(s_1\lambda_1, \cdots, s_n\lambda_n)|}{|\Gamma_1(s_1\lambda_1, \cdots, s_n\lambda_n)|} \quad (s_l = \pm 1,\ l = 1, \cdots, n) \tag{14}$$

で与えられる。ここで $|\Gamma_1(s_1\lambda_1,\cdots,s_n\lambda_n)|$ と $|\bar{\Gamma}_{i,j}(s_1\lambda_1,\cdots,s_n\lambda_n)|$ はともに $s_1\lambda_1,\cdots,s_n\lambda_n$ の交代式なので $s_1\lambda_1,\cdots,s_n\lambda_n$ の差積と対称式の積で書き表せ、

$$|\Gamma_1(s_1\lambda_1,\cdots,s_n\lambda_n)| = g(s_1\lambda_1,\cdots,s_n\lambda_n)\prod_{i<j}(s_i\lambda_i - s_j\lambda_j) \tag{15}$$

$$|\bar{\Gamma}_{i,j}(s_1\lambda_1,\cdots,s_n\lambda_n)| = h(s_1\lambda_1,\cdots,s_n\lambda_n)\prod_{i<j}(s_i\lambda_i - s_j\lambda_j) \tag{16}$$

を満たす $s_1\lambda_1,\cdots,s_n\lambda_n$ の対称式 $g(s_1\lambda_1,\cdots,s_n\lambda_n)$ と $h(s_1\lambda_1,\cdots,s_n\lambda_n)$ が存在する。このとき、次の定理が成り立つ。

定理 3 $\prod_{s_l=\pm 1} g(s_1\lambda_1,\cdots,s_n\lambda_n)$ は k の多項式として表すことが可能である。すなわち

$$\bar{g}(k) = \prod_{s_l=\pm 1} g(s_1\lambda_1,\cdots,s_n\lambda_n) \tag{17}$$

を満たす k の多項式 $\bar{g}(k)\in\mathbf{Z}[k]$ が存在する。また、$f_{2^n}(k)$ は $\bar{g}(k)$ を割り切る。□

定理 3 より $\bar{g}(k)$ は $f_{2^n}(k)$ を因子として含む k の多項式であるので、アルゴリズム 2 の ⟨1⟩ において $\phi(k)=\bar{g}(k)$ と置くことができる。$\bar{g}(k)$ は k の多項式であるので、多項式補間を用いて計算することが可能である。すなわち、$k_l\in\mathbf{Z}$ $(l=1,\cdots,L)$ に対して $\bar{g}(k_l)\in\mathbf{Z}$ が求めることにより、$\bar{g}(k)$ を計算できる（ただし、L は $L>\deg_k(\bar{g}(k))$ を満たす自然数）。

アルゴリズム 3 $\bar{g}(k)$ $(=\phi(k))$ **の計算**

⟨1⟩ k_l $(l=1,\cdots,L)$ を適当な整数におく。

⟨2⟩ $l\leftarrow 1$ とする。

⟨3⟩ $H_{k=k_l}$ の固有値を計算し、実部が負であるものを $\bar{\lambda}_1,\cdots,\bar{\lambda}_n$ と置く。

⟨4⟩ $\bar{g}(k_l)$ を下記の式より求める。

$$\bar{g}(k_l) = \prod_{s_l=\pm 1} g(s_1\bar{\lambda}_1,\cdots,s_n\bar{\lambda}_n) \tag{18}$$

ただし

$$g(s_1\bar{\lambda}_1,\cdots,s_n\bar{\lambda}_n) = \frac{|\Gamma_1(s_1\bar{\lambda}_1,\cdots,s_n\bar{\lambda}_n)|}{\prod_{i<j}(s_i\bar{\lambda}_i - s_j\bar{\lambda}_j)} \tag{19}$$

である。

⟨5⟩ $l<L$ ならば ⟨3⟩ へ行く。そうでなければ、$\bar{g}(k_l)$ $(l=1,\cdots,L)$ より多項式補間を用いて $\bar{g}(k)\in\mathbf{Z}[k]$ を求める。

注意：先に述べたように $|\Gamma_1(s_1\bar{\lambda}_1,\cdots,s_n\bar{\lambda}_n)|$ は $s_1\bar{\lambda}_1,\cdots,s_n\bar{\lambda}_n$ の交代式なので、(19) の $g(s_1\bar{\lambda}_1,\cdots,s_n\bar{\lambda}_n)$ は $s_1\bar{\lambda}_1,\cdots,s_n\bar{\lambda}_n$ の対称式である。$|\Gamma_1(s_1\bar{\lambda}_1,\cdots,s_n\bar{\lambda}_n)|$ の行列式の計算において、列操作によりあらかじめ $(s_i\lambda_i - s_j\lambda_j)$ $(i\neq j)$ の因子を括り出しておくことが可能なので実際には (19) の分母は計算する必要はない。

4 H_2 最適制御問題への適用

アルゴリズム 3 を用れば、$f_{2^n}(k)$ を割り切る k の多項式 $\phi(k)$ が計算できることを示したが、問題を H_2 最適制御系設計に限定すればもっと簡単に $\phi(k)$ を求めることが可能である。

4.1 H_2 最適制御系問題

次の微分方程式で表されるプラント（制御対象）が与えられているとする。

$$\frac{dx}{dt} = Ax + Bu$$

このとき、評価関数

$$\int_0^\infty \left(x^T Q x + u^T R u\right)\, dt$$

を最小化するように入力 u を定めることを H_2 最適制御という。状態変数 x を用いることができるとき、この問題の解は $u = -R^{-1}B^T P x$ で与えられることが知られている。ただし、P は Riccati 方程式

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \tag{20}$$

の正定解である。

4.2 Riccati 方程式の解と可制御性行列

H_2 最適制御問題で出てくる Riccati 方程式 (20) は可制御であれば一意解を持つことが知られている。システムが可制御であることの必要十分条件は可制御性行列 $\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$ がフルランクを持つことなので B が列ベクトル (すわなち $n \times 1$ 行列) の時は

$$\begin{vmatrix} B & AB & \cdots & A^{n-1}B \end{vmatrix} \neq 0 \tag{21}$$

である。これは言い換えると、この条件が満たされる限り Riccati 方程式は一意解を持つ。よって主係数は $\begin{vmatrix} B & AB & \cdots & A^{n-1}B \end{vmatrix}$ のべき乗に限られることになる。実際、先の数値例では

$$\begin{aligned} \begin{vmatrix} B & AB & \cdots & A^{n-1}B \end{vmatrix} &= 2k-1 \\ f_4(k) &= (2k-1)^4 \end{aligned}$$

である。いくつかの数値実験を見ると、H_2 最適制御から出た Riccati 方程式の場合、主係数 $f_{2^n}(k)$ は

$$f_{2^n}(k) = \begin{vmatrix} B & AB & \cdots & A^{n-1}B \end{vmatrix}^n \tag{22}$$

で与えられると予想される。

5 結論

代数 Riccati 方程式の解の定義多項式の計算法を示した。提案した方法は多項式補間を用いているので、並列計算が可能であり、今後出てくるであろうマルチコアの CPU を用いれば計算の高速化が期待できる。定義多項式の主係数の計算に時間がかかるが、H_2 最適制御から出てくる Riccati 方程式に関しては可制御性行列の行列式を計算することにより、この計算を効率化できる。H_∞ 制御問題に関しても同様に、主係数の計算が効率化可能であるか検討中である。

[1] T. Kitamoto and T. Yamaguchi: Parametric Computation of H_∞ Norm of a System; Proc. SICE-ICCAS2006.

[2] T. Kitamoto and T. Yamaguchi: On the parametric state feedback H_∞ control, 投稿中

[3] S. Boyd, V. Balakrishnan and P. Kabamba: A bisection method for computing the H_∞ norm of a transfer matrix and related problems; Math. Control, Signals and Systems, **2**-3, 207/220 (1989)

[4] N. A. Bruinsma and M. Steinbuch: A fast algorithm to copmute the H_∞-norm of a transfer function matrix; Systems and Control letters, **14**, 287/293 (1990)

[5] K. Zhou, J. Doyle and K. Glover: Robust and Optimal Control; Prentill-hall. Inc, New Jersey (1996)

[6] C. Abdallah, P. Dorato, W. Yang, R. Liska and S. Steinberg: Application of Quantifier Elimination Theory to Control System Design; Proc. of 4th IEEE Mediteranean Symposium of Control and Automation, Maleme, Crete, 340/345.

[7] H. Anai and H. Yanami, "SyNRAC: A maple-package for solving real algebraic constraints," Proc. of CASA'2003, P.M.A. Sloot et al. (ICCS 2003) editors, Vol. 2657 of LNCS, Springer-Verlag, 2003.

[8] D. Nesic and I. M. Y. Mareels: Dead Beat Controllability of Polynomial Systems: Symbolic Computation Approarch; IEEE Trans. Auto. Cont., **43**-2, 162/175 (1998)

[9] P. Dorato, W. Yang and C. Abdallah: Robust Multi-Objective Feedback Design by Quantifier Elimination; J. Symbolic Computation, **24**, 153/159 (1997)

[10] H. Hong, R. Liska and S. Steinberg: Testing Stability by Quantifier Elimination; J. Symbolic Computation, **24**, 161/187 (1997)

[11] 西村、狩野: 制御のためのマトリクス・リカッチ方程式; システム制御情報学会編, 朝倉書店 (1996)