행열부호함수와 시스템계산에의 그 응용

全 琪 準
(慶北大 工大 助教授)

■ 차 례 ■

1 서 론

Roberts[1] 에 의해 고안된 행열부호함수(matrix sign function)는 행열의 고유치(eigenvalue) 와 고유벡타(eigenvector)를 모르는 상황에서 반복적인 앨고리즘에 의해 계산할 수 있는 행열의 함수이며 이는 고유벡타를 보존하고 있는 까닭에 행열의 고유구조(eigenstructure)를 연구하는데나 시스템공학의 여러가지 문제를 푸는데 유용한 수학적 도구로서 이용되어 오고 있다. 몇가지 응용의 예를들자면, 연속시간(continuous-time)시스템의 선형조정기(linear regulator)문제에서 야기되는 Riccati 대수행열방정식의 해를 구하거나 시스템의 안정도 해석에 필요한 Lyapunov 방정식의 해 및 고유치와 고유벡타의 계산[2] 에 이용할 수 있으며 eigenprojector 의 계산으로부터 일반역행열(generalized inverse) 및 positive semi-definite 행열의 p 제곱근[3] 을 계산하는 것과 시스템의 안정도검사에의 응용[4] 등이다. 최근에는 본래의 정의를 약간 일반화함으로써 이산시간(discrete-time) 시스템의 선형조정기 문제에도 그대로 적용할 수 있음이 밝혀져서 앞으로 이산시간시스템의 연구에도 응용이 기대된다.

이 글에서는 행열부호함수의 정의 및 앨고리즘을 알기 쉽게 소개하고 Riccati 방정식의 해 등 앞에서 말한 몇가지의 예를 제시하므로서 이 함수를 널리 보급하고 여기에 관련된 이론의 발전과 더 많은 응용을 유도하고자 함이다.

2 행열부호함수

일반적인 $n \times n$ 행열 A는 다음과 같이 표시된다. 즉,

$$A = M J M^{-1}, \tag{1}$$

여기서 M은 A의 고유벡타행열이고 J는 Jordan 표준형으로 다시 Jordan 블럭으로 다음과 같이 표시할 수 있다. 즉,

$$J = B_{diag}[J(\lambda_1), J(\lambda_2), \cdots, J(\lambda_r)] \tag{2}$$

여기서 $B_{diag}[\cdot]$은 블럭대각행열을 의미하며 λ_i 는 A의 고유치이다. 이 때 A의 고유치를 포함하는 어떤 영역에서 해석적인 함수 f가 있으면 $f(A)$는

$$f(A) = M f(J) M^{-1} \tag{3}$$

로 표시되며, 여기서

$$f(J) = B_{diag}[f(J_1), f(J_2), \cdots, f(J_r)] \tag{4}$$

으로 나타나며 각 $f(J_i)$블럭은 다음과 같은 형태를 갖는다.

$$f(J_i)=\begin{bmatrix} f(\lambda_i) & f'(\lambda_i)/1! & \cdots & f^{(n_i-1)}(\lambda_i)/(n_i-1)! \\ 0 & f(\lambda_i) & \cdots & f^{(n_i-2)}(\lambda_i)/(n_i-2)! \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & f(\lambda_i) \end{bmatrix} \tag{5}$$

이제 부호함수를

$$sign(\lambda_i)=\begin{cases} +1, & Re\lambda_i>0 \\ -1, & Re\lambda_i<0 \end{cases} \tag{6}$$

와 같이 정의하면 $sign(Re\,J)={}_{diag}[sign(Re\lambda_1), sign(Re\lambda_2), \cdots sign(Re\lambda_n)]$ 으로 되어서

$$S(A) \triangleq sign(A)=M sign(Re\,J) M^{-1} \tag{7}$$

로 $sign(A)$ 가 정의된다. ($\lambda_i=\lambda_j$ 의 경우도 포함됨에 유의)

그러나, 위의 정의는 A의 고유치가 0일 때와 허수일 때는 제외되며 이 까닭은 곧 제시하게 될 반복앨고리즘을 적용할 수 없기 때문이다. 여기서 우리는 위에서 제외시킨 경우를 포함하는 일반부호함수 (generalized sign function)를 요구하게 되는데 이는 단순히 제외된 경우를 다음과 같이 0으로 놓으면

$$\hat{sign}(\lambda_i)=\begin{cases} 1 & Re\lambda_i>0 \\ 0 & Re\lambda_i=0 \\ -1 & Re\lambda_i<0 \end{cases} \tag{8}$$

로 되어서 A의 일반행열부호함수는 행열부호함수와 마찬가지로 표기가 된다.
즉,

$$\hat{S}(A)=M\,\hat{sign}(J)\,M^{-1} \tag{9}$$

이 행열부호함수를 계산하는 반복앨고리즘은 처음에 $S^2=I$의 근을 구하는 Newton의 앨고리즘에 근거하여 A의 고유치의 실수부가 0 이 아닐 때는 다음과 같이 주어졌었다. 즉,

$$S_{k+1}=\frac{1}{2}(S_k+S_k^{-1}),\quad S_0=A \tag{10}$$

따라서, 이 앨고리즘은 2차로 수렴(quadratic convergence)하는 성질을 가졌으며 미리 주어진 ε 이 $|T_r S_k^2-n|<\varepsilon$ 을 만족할 때 반복을 멈출 수 있다. 또한, 이 앨고리즘은 나중에 Hoskins 와 Walton[6] 및 Balzer[7] 에 의해 수렴성이 더욱 향상되었다.

한편, 앨고리즘 (10)은 A의 고유치가 0이거나 허수일 때는 사용할 수 없으므로 약간의 수정이 불가피하다. 즉, A의 스펙트럼을 ρ만큼 좌우로 평행이동하므로서 0 고유치를 피할 수가 있으며 이 때

$$A+\rho I=M(J+\rho I)M^{-1} \tag{11}$$

가 되어서 S_k^+ 및 S_k^-를 각각 초기조건 $S_0^+=A+\rho I$ 및 $S_0^-=A-\rho I$에서의 k번째 반복앨고리즘에서 얻어진 행열부호라고 하면 일반행열부호 앨고리즘은

$$\hat{S}_k=(S_k^++S_k^-)/2 \tag{12}$$

로 됨을 쉽게 알 수 있다. 이때 $A+\rho I$와 $A-\rho I$가 또 다른 0고유치를 만들지 않도록 ρ가 선택되어야 함은 물론이며 모든 고유치에 대해 $\rho<|Re\lambda_i|$를 만족하여야 한다. 이렇게 함으로써 $Re\lambda_i\neq 0$ 인 고유치는 평행이동으로 인하여 그 본래의 부호를 잃지 않게 된다.

3 Riccati형 방정식의 해

3.1 Riccati 대수행열방정식의 해

Riccati 대수행열방정식은 선형시불변연속 시간시스템의 최적제어 뿐만 아니라 Kalman 및 Wiener 여파문제에서도 나타나는 등 비교적 널리 쓰이는 방정식이며 그 해법 또한 여러가지가 알려져 있다. 여기서는 앞서 소개한 행열부호함수를 이용하여 일반적인 형태의 Riccati 대수행열방정식의 해를 구하는 방법을 설명하고자 한다.

Riccati 대수행열방정식의 일반 형태는

$$A+BP+PC+PDP=0 \tag{13}$$

로 표시되며 여기서 A, B, C, D 및 미지행열 P는 $n\times n$ 행열이다. 계수행열들로부터 $2n\times 2n$ 행열 H를

$$H=\begin{bmatrix} B & A \\ -D & -C \end{bmatrix} \tag{14}$$

와 같이 구성하면 이 행열 H는 또한

$$H=\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}\begin{bmatrix} \Lambda_1 & O \\ O & \Lambda_2 \end{bmatrix}\begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \tag{15}$$

와 같이 표준형태로 변경될 수 있다. 여기서 고유벡타행열 M은 네 블럭으로 나뉘어졌으며 Λ_1 과 Λ_2는 Jordan 형의 분할이다. W_{ij}블럭은 M의 역행열의 분할 즉 $W\triangleq M^{-1}$이다. 물론 M과 W의 분할된 블럭들은 이후의 대수연산에 적합하도록 분할되어야 한다.

이 Riccati 방정식의 해 P는 M_{22}의 역행열이 존재하고 그 차수가 $n \times n$ 이면

$$P = M_{12}\ M_{22}^{-1} \tag{16}$$

로 주어짐은 이미 알려져 있다.[8),9)] 여기서 행열 M의 칼럼과 고유치의 순서를 함께 바꿀 수 있으므로 일반적으로 식 (16)의 P는 유일하지 않을 수 있으며 실제로 $P = M_{11}\ M_{21}^{-1}$ 도 만일 M_{21}^{-1}이 존재하면 이 방정식의 해가 될 수 있다.

만일 식 (15)에서 H의 고유치가 허수축 좌우로 n 개씩 분포되어 있다면 이들을 각각 Λ_1과 Λ_2에 수용시킬 수 있으며 따라서 행열 H에 행열부호함수를 적용하면 아래와 같게 된다.

$$sign(H) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} I_{n\times n} & O \\ O & -I_{n\times n} \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \tag{17}$$

여기서 $I_{n\times n}$ 은 $n \times n$ 단위행열이다. 이제 새로운 행열 F를

$$F = sign(H) + \begin{bmatrix} I_{n\times n} & O \\ O & -I_{n\times n} \end{bmatrix} \tag{18}$$

으로 정의하면

$$F = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \left\{ \begin{bmatrix} I_{n\times n} & O \\ O & -I_{n\times n} \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} + \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} I_{n\times n} & O \\ O & -I_{n\times n} \end{bmatrix} \right\} \tag{19}$$

으로 되어서 이는 곧 다음과 같이 간단히 표현된다. 즉,

$$F = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} 2W_{11} & O \\ O & -2W_{22} \end{bmatrix} \triangleq \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \tag{20}$$

필요한 역행열이 존재하면 이제 Riccati 방정식 (13)의 해 P는

$$P = M_{11}\, M_{21}^{-1} = F_{11}\, F_{21}^{-1} \tag{21}$$

및

$$P = M_{12}\, M_{22}^{-1} = F_{12}\ F_{22}^{-1} \tag{22}$$

로 주어진다.

여기서 당연히 제기되는 문제는, 항상 고유치가 허수축 좌우로 균등히 분산되어 있느냐 하는 점인데 실제로 대부분의 응용에서는 이와 같이 분포되어 있으며 만일 그렇지 못한 경우에는 Walsh 행열함수를 이용하여 H의 스펙트럼을 이동하므로서 Riccati 방정식의 해를 구할 수 있다.[10)]

계산의 예를 하나 들면, 식 (13)에서 계수행열이 각각

$$A = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix},\quad B = \begin{bmatrix} 0 & 0 \\ 1 & -2 \end{bmatrix},\quad C = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix},$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$$

일때

$$H = \begin{bmatrix} 0 & 0 & 4 & 0 \\ 1 & -2 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 2 \end{bmatrix}$$

로 되며 앨고리즘 (10)에 의해

$$sign(H) = \begin{bmatrix} 0.0 & 0.667 & 4.667 & 1.667 \\ 0.333 & -0.667 & 1.667 & 1.0 \\ 0.167 & 0.0 & 0.0 & -0.333 \\ 0.0 & 0.333 & -0.667 & 0.667 \end{bmatrix}$$

가 얻어진다. 다음에 식 (18)로 정의된 행열 F는 쉽게 계산된다. 즉,

$$F = \begin{bmatrix} 1.0 & 0.667 & 4.667 & 1.667 \\ 0.333 & 0.333 & 1.667 & 1.0 \\ 0.167 & 0.0 & -1.0 & -0.333 \\ 0.0 & 0.333 & -0.667 & -0.333 \end{bmatrix}$$

이제 Riccati 방정식의 해

$$P = F_{11}\ F_{21}^{-1} = \begin{bmatrix} 6.0 & 2.0 \\ 2.0 & 1.0 \end{bmatrix}$$

으로 식 (21)에 의해 구해진다.

다음에는 이 Riccati 형 방정식의 해를 필요로 하는 경우의 예를 몇가지 들어 보기로 하겠다.

3.2 연속시간시스템의 선형조정기(Regulator)

선형연속시간시불변 시스템

$$\dot{x}(t) = A\,x(t) + B\,u(t) \tag{23a}$$

$$y(t) = C\,x(t) \tag{23b}$$

(A, B, C는 각각 $n \times n$, $n \times m$, $p \times n$ 행열)에서 최소화하고자 하는 성능지수 (performance index)가 다음과 같이 주어졌을 때, 즉

$$J = \int_0^{\infty} [\, x^T(t)\, Q\, x(t) + u^T(t)\, R\, u(t) \,]\, dt \tag{24}$$

(Q는 $n \times n$ semi-positive definite, R은 $m \times m$ positive definite 행열) 시스템 (23) 이 완전 제어가능이면 정상상태에서의 최적제어법칙은 다음과 같으며

$$u(t) = -R^{-1}B^T P x(t) \tag{25}$$

이때 $n \times n$ 행열 P는 Riccati 행열방정식

$$Q + A^T P + PA - PBR^{-1}B^T P = 0 \tag{26}$$

의 해이다.[11] 따라서, 행열 H를

$$H = \begin{bmatrix} A^T & Q \\ BR^{-1}B^T & -A^T \end{bmatrix} \tag{27}$$

와 같이 정의하면 이의 고유치는 허수축에 존재하지 않고 원점에 대해서 대칭으로 분포되어 있으므로[12] 3. 1절에서와 같이 P를 구할 수 있다.

3.3 이산시간시스템의 선형조정기

선형이산시간시불변 시스템

$$x(k+1) = Ax(k) + Bu(k) \tag{28a}$$

$$y(k) = Cx(k) \tag{28b}$$

(A, B, C는 $n \times n$, $n \times m$, $p \times n$ 행열이고 A는 정칙 (non-singular)) 에서 성능지수가 다음과 같이 주어지면

$$J = \sum_{i=0}^{\infty} [x^T(i)Qx(i) + u^T(i)Ru(i)] \tag{29}$$

($Q = \hat{Q}\hat{Q}^T$ 는 $n \times n$ semi-positive definite 대칭 행열, R은 $m \times m$ positive definite 행열){A, $\hat{Q}$}는 detectable 이고 {A, B} 는 안정화가능이라는 가정 아래서 정상상태의 최적제어법칙은

$$u(k) = -(R + B^T PB)^{-1} B^T PAx(k) \tag{30}$$

로 주어지며, 이때 semi-positive definite 행열 P는 다음의 대수 비선형이산시간 Riccati 방정식의 해가 된다.

$$P = Q + A^T PA - A^T PB(R + B^T PB)^{-1} B^T PA \tag{31}$$

이때 $2n \times 2n$ Hamiltonian 행열 H를

$$H = \begin{bmatrix} A^{-1} & A^{-1}BR^{-1}B^T \\ QA^{-1} & A^T + QA^{-1}BR^{-1}B^T \end{bmatrix} \tag{32}$$

와 같이 정의하면 H의 고유치는 단위원의 외부와 내부에 각각 n개씩 분포하게 되며 식 (31) 의 Riccati 이득행열 P는 다음과 같이 주어진다.[13] 즉,

$$P = M_{21} M_{11}^{-1} \tag{33}$$

여기서 M_{ij} 는 앞에서와 마찬가지로 H의 고유벡타 행열 M의 분할된 블럭이다. 이러한 경우에도 행열부호앨고리즘 (10) 에서 초기조건을 $S_0 = (A-I)(A+I)^{-1}$ 로 두면 3.1절에서와 유사한 과정을 거쳐서 이득행열 P는 행열부호함수로부터

$$P = F_{21} F_{11}^{-1} \tag{34}$$

와 같이 얻을 수 있으며* F_{ij}는 3. 1절에서와 같이 분할된 행열이다.

4 Eigenprojector의 계산과 응용

4.1 Eigenprojector의 계산

$n \times n$ 행열 A의 전 스펙트럼을 포용하는 eigenprojector 를 다음과 같이 정의하자.

(1) P^+(양 eigenprojector) : 행열 A의 양의 실수부를 가진 고유치에의 투사

(2) P^- (음 eigenprojector) : 행열 A의 음의 실수부를 가진 고유치에의 투사

(3) P^0 (영 eigenprojector) : 행열 A의 영의 고유치에의 투사

(4) P^I (허 eigenprojector) :행열 A의 허수부만 가진 고유치에의 투사

이 eigenprojector 들을 행열 A에 곱하면 각기 해당하는 부공간 (subspace) 을 얻을 수 있다. 즉,

$$A^+ = AP^+ = P^+A$$
$$A^- = AP^- = P^-A$$
$$A^0 = AP^0 = P^0A$$
$$A^I = AP^I = P^IA$$

A^0 는 분명히 0행열이지만 P^0 는 그렇지 않고 eigenprojector 의 고유치는 1 혹은 0이지만 고유벡타는 A의 그것과 같음을 쉽게 알 수 있다. 또한, 이 eigenprojector 들은 다음과 같은 성질을 갖고 있다.

(P.1) Eigenprojector 들의 합은 단위행열이다. 즉, $P^+ + P^- + P^0 + P^I = I$

(P.2) Eigenprojector 들은 상호 직교한다. 즉, $P^+P^- = P^-P^+ = 0$ 등

(P.3) Eigenprojector 들은 idempotent 이다. 즉, $P^+P^+ = P^+$, $P^-P^- = P^-$ 등

이들의 계산은 일반행열부호를 이용하면 쉽게 얻

*S_0의 고유치는 허수축 좌우에 분포하게 되므로 앨고리즘(10)을 이용할 수 있으며 여기에 대한 구체적인 설명은 참고문헌[5]을 보기를 권한다.

어진다. 임의의 $n \times n$ 행열 A의 일반행열부호는

$$\hat{S}(A) = P^{+} - P^{-} \tag{35}$$

로 될 것이며 양변에 제곱을 하면 위의 성질(P.2)와 (P.3)에 의해

$$\hat{S}^{2}(A) = P^{+} + P^{-} \tag{36}$$

로 된다. 따라서 P^{+}와 P^{-}는 각각 다음과 같이 구할 수 있다.

$$P^{+} = [\hat{S}^{2}(A) + \hat{S}(A)]/2 \tag{37}$$

$$P^{-} = [\hat{S}^{2}(A) - \hat{S}(A)]/2 \tag{38}$$

(P.1)과 식 (36)에 의해

$$P^{0} + P^{I} = I - P^{+} - P^{-} = I - \hat{S}^{2}(A) \tag{39}$$

가 얻어진다. 식(41)에 A를 곱하면

$$A_1 \triangleq A^{0} + A^{I} = A(P^{0} + P^{I}) = A(I - \hat{S}^{2}(A)) \tag{40}$$

로 되며 $A^{0} = 0$인 점을 감안하면 A_1의 제곱은 모두 실수 고유치를 갖게 됨을 알 수 있다. 따라서 (P.2), (P.3) 및 식 (39)에 의해

$$A_2 \triangleq A + A_1^{2} = A + A^{2}(P^{0} + P^{I})^{2}$$
$$= A + A^{2}(I - \hat{S}^{2}(A)) \tag{41}$$

가 되며 A_2는 실수, 복소수 및 0고유치에 대한 스펙트럼을 갖게 되고 모든 허수축에 존재한 고유치 $\lambda_i = j\omega_i$는 $-\omega_i^{2}$으로 변해서 실수 고유치에 포함되어 버렸으며 그 밖의 고유치는 이 연산과정에서 그대로 남게 된다. 그러므로, 허 eigenprojector P^{I}는 다음과 같이 구해진다.

$$P^{I} = \hat{S}(A) - \hat{S}(A_2) \tag{42}$$

따라서, 식(36)과 (42)에 의해 마침내 P^{0}는

$$P^{0} = I - P^{+} - P^{-} - P^{I} = I - \hat{S}^{2}(A)$$
$$- \hat{S}(A) + \hat{S}(A_2) \tag{43}$$

으로 구할 수 있다. 이 영 eigenprojector P^{0}는 positive semi-definite 행열의 p제곱근과 비정칙(singular) 행열의 일반역행열 (generalized inverse) 을 구하는데 이용될 수 있다.

4.2. 비정칙행열의 일반역행열의 계산

$n \times n$ 행열 A의 0인 고유치와 0이 아닌 고유치를 구분하여 표시하면

$$A = M\, diag(\lambda_1, \cdots, \lambda_k, 0, \cdots, 0, \lambda_r, \cdots, \lambda_n)\, M^{-1} \tag{44}$$

와 같이 되며 (이때 2차이상의 Jordan 블럭이 없다고 가정), A의 일반역행열 A^{*}는

$$A^{*} = M\, diag(\lambda_1^{-1}, \cdots, \lambda_k^{-1}, 0, \cdots, 0, \lambda_r^{-1}, \cdots, \lambda_n^{-1})\, M^{-1} \tag{45}$$

로 표시된다. 이때 A^{*}는 Penrose[15]의 아래 네가지 성질을 모두 만족한다.

(1) $AA^{*}A = A$

(2) $A^{*}AA^{*} = A^{*}$

(3) $(A^{*}A)^{T} = A^{*}A$

(4) $(AA^{*})^{T} = AA^{*}$

만일 P^{0}가 식(43)에서와 같이 구해졌다고 가정하고 $n \times n$ 행열 A_3를

$$A_3 \triangleq A + P^{0} \tag{46}$$

로 정의하면 A_3의 rank는 n이 될 것이다. 즉,

$$rank(A) + rank(P^{0}) = n \tag{47}$$

따라서, A_3의 역행열은 존재하며 P^{0}의 고유벡타는 A의 그것과 같기 때문에

$$A_3^{-1} = (A + P^{0})^{-1} = A^{*} + P^{0} \tag{48}$$

즉,

$$A^{*} = A_3^{-1} - P^{0} \tag{49}$$

로 구해진다.

4.3 Positive semi-definite 행열의 P제곱근

Positive definite 행열의 p제곱근을 구하는 문제는 Hoskins 와 Walton[6] 등에 의해 많이 다루어졌으나 행열이 정칙이어야 하는 제약이 있었다. 여기서는 P^{0}를 이용하여 보다 일반적인 방법을 소개하고자 한다.

$n \times n$ 행열 A를 고유치가 0일 수도 있는 positive semi-definite 라 하고 A_4를 A의 p제곱근, 즉

$$A = A_4^{p} \tag{50}$$

라고 놓자, 정칙행열의 p제곱근을 구하는 Newton의 앨고리즘은 Hoskins 와 Walton[6]에 의해

$$X_{i+1} = [(p-1)X_i + AX_i^{1-p}]/p, \quad X_0 = I,$$
$$\| X_{i+1} - X_i \| < \varepsilon \tag{51}$$

로 주어졌으나 A에 0 고유치가 있으면 이 앨고리즘은 부적당하다.

그런데, 이렇게 0 고유치를 갖는 행열 A의 일반행열부호는

$$\hat{S}(A) = P^{+} \tag{52}$$

가 되며 이로 부터 영 eigenprojector 는

$$P^{0} = I - P^{+} = I - \hat{S}(A) \tag{53}$$

로 된다. 여기서 P^{0}는 A와 같은 고유벡타를 갖지만 A의 0 고유치 대신에 +1의 고유치를 가지고 A의 0이 아닌 고유치는 0 고유치로 바뀐다. 따라서 $A + P^{0}$의 고유벡타는 A 및 P^{0}의 그것과 같지만 고유치는 A의 0 고유치 대신에 +1이 되므로 정칙행열이 된다. 이제,

$$A_D = A + P^0 \tag{54}$$

라 놓으면 A_D 는 엘고리즘 (51)에 의해 p 제곱근을 구할 수 있다. 그러나, A_D 의 p 제곱근에는 A의 0 고유치 대신에 P^0의 고유치 1이 들어 있으므로 이는 $(A_D)^{1/p}$ 에 P^+ 를 곱하면 eigenprojector 의 직교성에 의해 제거 가능하다. 즉,

$$A_4 = (A_D)^{1/p} P^+ \tag{55}$$

다음 계산의 예에서 A의 고유치는 $\lambda_1 = 0$, $\lambda_2 = 0$, $\lambda_3 = 3.258$ 및 $\lambda_4 = 10.741$ 이다.

$$A = \begin{bmatrix} 2.0 & -1.0 & 1.0 & -1.0 \\ -1.0 & 4.0 & 3.0 & -3.0 \\ 1.0 & 3.0 & 4.0 & -4.0 \\ -1.0 & -3.0 & -4.0 & 4.0 \end{bmatrix}$$

$\rho = 0.5$ 를 사용하여 $\varepsilon = 10^{-6}$ 일때 일반행열부호는 7번만에 얻어졌으며 이는 곧 P^+와 같다. 즉,

$$\hat{S}(A) = P^+ = \begin{bmatrix} 0.6 & -0.4 & 0.2 & -0.2 \\ -0.4 & 0.6 & 0.2 & -0.2 \\ 0.2 & 0.2 & 0.4 & -0.4 \\ -0.2 & -0.2 & -0.4 & 0.4 \end{bmatrix}$$

위에 소개한 방법에 의해 A의 5제곱근 $A^{1/5}$ 는 $\varepsilon = 10^{-5}$ 로 했을 때 7번만에 다음과 같이 얻어졌다.

$$A^{1/5} = \begin{bmatrix} 0.761942 & -0.492761 & 0.269181 & -0.269181 \\ -0.492761 & 0.835145 & 0.360339 & -0.360384 \\ 0.269181 & 0.360384 & 0.629565 & -0.629565 \\ -0.269181 & -0.360384 & -0.629565 & 0.629565 \end{bmatrix}$$

5 결 론

부호함수 및 이의 연장으로 볼 수 있는 행열부호함수를 소개하고 시스템의 해석과 제어에의 응용의 예를 몇가지 들었다. 이들 응용 중 대부분은 직접 행열의 고유벡타로부터 구해질 수 있음을 행열부호함수의 정의에서 알 수 있으며 따라서 이 방법들은 고유벡타를 구하지 않고 여러가지 해를 구하는 방법으로 볼 수 있다.

참고문헌

1) J. D. Roberts, Linear model reduction and solution of the algebraic Riccati equation by use of the sign function, CUED/B-Control/TR13, Report, Cambridge University, 1971, also published in Int. J. Control, Vol. 32, pp. 677-687, 1980
2) E. D. Denman, and A. N. Beavers, "The matrix sign function and computations in systems," Appl. Math. & Comput., Vol. 2, pp. 63-94 1976
3) E. D. Denman and J. Leyva-Ramos, "Spectral decomposition of a matrix using the generalized sign matrix", Appl. Math. & Compt., Vol. 8, pp. 237-251, 1981
4) F. Attarzadeh, "Relative stability test for continuous and sampled-data control systems using the generalized sign matrix," IEE Proc. D, Control Theory & Appl., 129(5), pp. 189-192, 1982
5) L. S. Shieh, Y. T. Tsay and R. E. Yates, "Some properties of matrix sign functions derived from continued fractions," IEE Proc. Pt. D Vol. 130, No. 3, pp. 111-118, 1983
6) W. D. Hoskins and D. J. Walton, "A faster, more stable method for computing the pth roots of positive definite matrices," Linear Algebra & Appl., Vol. 26, pp. 139-163, 1979
7) L. A. Balzer, "Accelerated convergence of the matrix sign function method of solving Lyapunov, Riccati and other matrix equations", Int. J. Control, Vol. 32, pp. 1057-1078, 1980
8) K. Martensson, "On the matrix Riccati equation," Inf. Sci. Vol. 3 pp. 17-49, 1971
9) J. E. Potter, "Matrix quadratic solutions", SIAM J. Appl. Math. Vol. 14, pp. 496-591, 1966
10) A. N. Beavers and E. D. Denman, "A new solution method for quadratic matrix equations," Math. Biosci., Vol. 20, pp. 135-143, 1974
11) D. E. Kirk, Introduction to Optimal Control Theory, Prentice-Hall, Englewood Cliffs, N. J., 1970, p. 217
12) J. J. O' Donnell, "Asymptotic solution of the matrix Riccati equation and optimal control", Proc. 4th Ann. Allerton Conf. on Circuit and System Theory, Urbana, Ill., Oct. 1966
13) B. C. Kuo, Digital Control System, Holt, Rinehart and Winston, New York, 1980, pp. 622-626
14) S. Barnett and C. Storey, "Some applications of the Lyapunov matrix equation", J. Inst. Math. Appl., Vol. 4, pp. 33-42, 1968
15) R. Penrose, "A generalized inverse of a matrix", Proc. Cambridge Philos. Soc., Vol. 51, pp. 406-413, 1955