2002년도 대한전자공학회/대한전기학회 시스템 및 제어분야 합동 추계학술대회 논문집(2002.11.30)

LMI 가능성 문제를 위한 타원 알고리즘의 개선

방대인, 최진영
서울대학교 전기 · 컴퓨터 공학부
전화 : 02-872-7283 / 핸드폰 : 011-9726-1906

An improved ellipsoid algorithm for LMI feasibility problems

Dane Bahng, Jin Young Choi
School of Electrical Engineering & Computer Science, Seoul National University
E-mail : dibang@neuro.snu.ac.kr

Abstract

The ellipsoid algorithm solves some feasibility(or optimization) problems with LMI(Linear Matrix Inequality) constraint in polynomial time. Recently, it has been replaced by interior point algorithm due to its slow convergence and incapability of verifying feasibility. This paper proposes a method to improve its convergence by using the deep-cut method of linear programming. Simulation results show that the improved algorithm is more effective than the original one.

Ⅰ. 서 론

타원 알고리즘은 70년대 후반부터 알려진 간단한 형태의 알고리즘으로, 선형 제한조건 또는 선형 행렬 부등식(LMI, Linear Matrix Inequalities) 형태로 주어지는 제한조건을 가진, 몇몇 가능성 문제(feasibility problem)와 최적화 문제를 다항식 시간(polynomial time) 안에 해결하며 주목을 받기 시작했다. 하지만 느린 수렴속도와 문제 자체의 가능성 여부를 판단하는데 부적격하다는 한계를 드러냈으며[1] 근래에는 내점 알고리즘(interior point algorithm) 등으로 대체되고 있다[2]. 타원 알고리즘은 매 단계마다 전 단계에서 구한 타원의 절반 부분을 포함하는 최소부피의 타원을 생성함으로서 가능영역으로의 수렴성을 보장하며, 적용하는 문제에 따라 수렴속도를 증가시키기 위한 몇가지 방안이 제시되어 왔다[1,3]. 본 논문에서는 선형계획법(linear programming)에서 알려진 딥-컷(deep-cut)방법[1]을 적용하여, 보다 일반적인 LMI로 주어지는 제한조건을 가지는 가능성 문제에 대해 타원 알고리즘의 수렴속도를 향상시킬 수 있는 방안을 모색하고 수렴속도 증명과 모의실험을 통해 그 효과를 검증한다. 이를 위하여 Ⅱ에서는 LMI와 LMI 가능성 문제를 정의하고, Ⅲ에서는 기존의 타원 알고리즘을 소개하며, Ⅳ에서 이를 수정하고 그 성능향상을 논한다. 마지막으로 Ⅴ, Ⅵ에서 간단한 모의실험과 함께 결론을 맺는다.

Ⅱ. LMI 가능성 문제

본 장에서는 LMI와 LMI 가능성 문제의 정의를 소개한다[2,6].

2.1 LMI의 정의

LMI(Linear Matrix Inequality)는 (1)과 같이 정의되는 행렬 부등식이다.

$$F(x) \simeq F_0 + \sum_{i=1}^{n} x_i F_i > 0. \qquad (1)$$

단, $x \in R^n$이 변수이며 $F_i = F_i^T \in R^{n \times n}, i = 0, \ldots, n$는 주어진다.

2.2 LMI 가능성 문제(LMI Feasibility Problem)

LMI 가능성 문제는 임의의 주어진 초기치에서 시작하여 $F(x^*)\rangle 0$ 인 x^*를 찾거나 그러한 x^* 가 존재하지 않음을 밝히는 것이다. 이 때 $F(x^*)\rangle 0$인 x^*의 집합을 가능영역(feasible region)이라 한다. Ⅲ에서는 본 논문에서 제시하는 수정된 알고리즘 소개에 앞서 기존의 타원 알고리즘을 소개한다.

Ⅲ. 기존의 타원 알고리즘[2]

본 장에서는 LMI 가능성 문제를 풀기 위한 기존의 타원 알고리즘을 소개한다.

3.1 기존의 타원 알고리즘

[2]에서 제시한, LMI 가능성 문제를 위한 타원 알고리즘을 그림 1에서 소개한다. k 단계에서 (2)의 타원을 얻는다고 하자. 단, $k=0$일 경우 타원이 충분히 커서 찾고자 하는 가능영역을 그 안에 포함한다는 가정이 필요하다. 이제 $k+1$ 단계의 타원을 구성하는 방법을 알아본다.

$$(x-x_k)^T A_k^{-1}(x-x_k)\le 1,$$
$$A_k=A_k^T,\ A_k^{-1}\rangle 0\ ,\ k=0,1,2\ldots \qquad (2)$$

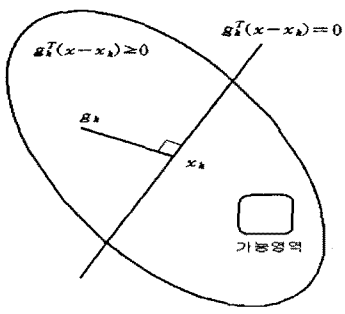

그림 1. 기존의 타원 알고리즘

우선 가능영역을 포함하지 않는 반타원을 제거하기 위해 g_k를 설계해야 한다. x_k는 (1)을 만족하지 않으므로(만족하면 알고리즘 중단) (3)을 만족하고 영이 아닌 벡터 u_k가 존재한다.

$$u_k^T F(x_k)u_k\le 0. \qquad (3)$$

이제 $g_{k,i}\simeq -u_k^T F_i u_k\in R^n,\ i=1,\ldots,n$ 을 정의하면 $g_k^T(x-x_k)\ge 0$ 을 만족하는 임의의 $x\in R^n$에 대해 식(4)와 같이 전개할 수 있다. 여기서 $x_{k,i}, g_{k,i}$ 는 각각 x_k, g_k의 i번째 요소를 나타낸다.

$$\begin{aligned} &u_k^T F(x)u_k \\ &= u_k^T(F_0+\sum_{i=1}^n x_{k,i}F_i-\sum_{i=1}^n x_{k,i}F_i+\sum_{i=1}^n x_i F_i)u_k \\ &= u_k^T F(x_k)u_k - g_k^T(x-x_k)\le 0. \end{aligned} \qquad (4)$$

따라서 $g_k^T(x-x_k)\ge 0$ 를 만족하는 x는 가능(feasible)하지 않으며 이 반타원 부분은 버린다. 따라서 $g_k^T(x-x_k)\langle 0$ 인 x들로 구성된 반타원만이 고려대상이 된다. 이제 이 반타원을 포함하는 최소부피의 타원을 형성함으로서 $k+1$ 단계에서의 타원식을 얻는다. (5)로 정의되는 반타원을 포함하는 최소부피의 타원은 (6)으로 주어진다[1,2]. 이후 $k+1$ 단계에서도 같은 과정을 반복한다.

$$(x-x_k)^T A_k^{-1}(x-x_k)\le 1,\ g_k^T(x-x_k)\langle 0. \qquad (5)$$

$$\begin{aligned} x_{k+1} &= x_k-\frac{1}{n+1}\frac{A_k g_k}{\sqrt{g_k^T A_k g_k}}, \\ A_{k+1} &= \frac{n^2}{n^2-1}\left(A_k-\frac{2}{n+1}\frac{A_k g_k g_k^T A_k}{g_k^T A_k g_k}\right). \end{aligned} \qquad (6)$$

3.2 기존 타원 알고리즘의 수렴속도

기존 타원 알고리즘의 매 단계 타원의 부피감소율은 (6)에 의해 (7)과 같이 유도할 수 있다.

$$det(A_{k+1})=\left(\frac{n^2}{n^2-1}\right)^n\times det(A_k).$$

$$\times det(I-\frac{2}{(n+1)}\frac{g_k g_k^T A_k}{g_k^T A_k g_k})$$

$$=(\frac{n^2}{n^2-1})^n \times det(A_k)\times(1-\frac{2}{(n+1)})$$

$$=(\frac{n^2}{n^2-1})^n \times \frac{(n-1)}{(n+1)} \times det(A_k),$$

$$\frac{S_{k+1}}{S_k}=\sqrt{\frac{det(A_{k+1})}{det(A_k)}}=(\frac{n}{n+1})(\frac{n^2}{n^2-1})^{\frac{n-1}{2}}. \quad (7)$$

여기서 S_k는 k단계에서의 타원의 부피이며 det는 행열식을 나타낸다.

Ⅳ. 수렴속도 향상을 위한 새 타원 알고리즘

본 장에서서는 Ⅲ의 알고리즘을 수정하고 그 수렴속도가 향상되었음을 증명한다.

4.1 새 알고리즘의 유도

(4)에서 $g_k^T(x-x_k)\geq 0$ 는 $u_k^T F(x) u_k \leq 0$가 되기 위한 충분조건일 뿐이므로 $u_k^T F(x) u_k \leq 0$에 대한 필요충분조건은 $\gamma \in R^{1+}$에 대해서 보다 일반적으로,

$$g_k^T(x-x_k)\geq -\gamma_k g_k^T g_k , \quad (8)$$

혹은,

$$g_k^T(x-x_{tmp,k})\geq 0, x_{tmp,k}=x_k-\gamma_k g_k. \quad (9)$$

로 쓸 수 있다. 이제 (9)를 만족하는 x에 대해서도 (4)가 여전히 만족되도록 γ를 설계하기 위해 (8)을 (4)에 적용하면

$$\gamma_k \leq -\frac{u_k^T F(x_k) u_k}{g_k^T g_k} \simeq \gamma_{\max,k}. \quad (10)$$

을 얻는다($\gamma_{\max,k}\in R^{1+}$). 이제부터는 $\gamma_k=\gamma_{\max,k}$라고 정한다. 따라서 (10)에 의해, 그림 2에서와 같이, 가능영역에 접하는 (9)의 직선과 타원의 일부로 둘러쳐진 모양만이 남게되며(이는 (5)의 반타원보다 작거나 같다.) 이를 포함하는 최소부피의 타원을 구해야 한다.

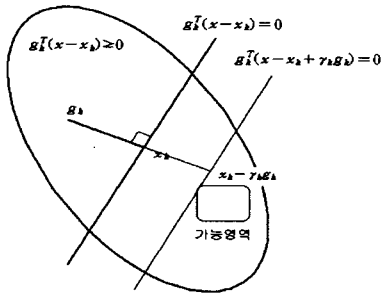

그림 2. 수정된 타원 알고리즘

A_k^{-1}는 양정치 행렬(positive definite matrix)이므로 $A_k^{-1}=N_k^T N_k$가 성립하도록 역행렬을 가지는 행렬 N_k을 정의할 수 있다[5]. 따라서 (11)로 정의되는 어파인 변환(affine transform)을 통해 (2)의 타원을 단위원으로 사상시킬수 있으며(그림 3), 이 변환을 통해 $g_k^T(x-x_{tmp,k})=0$, $x_{tmp,k}=x_k-\gamma_k g_k$을 만족하는 $x\in R^n$는 (12)의 직선으로 사상된다.

$$z_k \simeq N_k(x-x_k). \quad (11)$$

$$((N_k^{-1})^T g_k)^T(z_k+\gamma_k N_k g_k)=0. \quad (12)$$

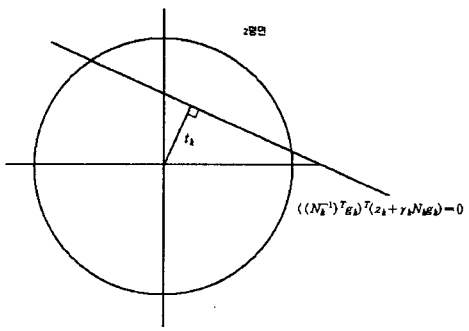

그림 3. 어파인 사상 후의 타원

z 평면에서 (12)직선과 원점의 거리(= t_k)는 (13)과 같이 구할 수 있으며 (11)에 의해 타원을 자르는 직선은 단위원을 자르는 직선이 되므로 $0 \le t_k < 1$을 만족한다.

$$t_k = -\frac{u_k^T F(x_k) u_k}{\sqrt{g_k^T A_k g_k}}. \tag{13}$$

t_k를 구성하였으면 이제 [1]에 의해 (6)을 수정하여 (14)-(17)의 알고리즘을 제시한다.

$$x_{k+1} = x_k - \frac{1+nt_k}{n+1}\frac{A_k g_k}{\sqrt{g_k^T A_k g_k}}, \tag{14}$$

$$A_{k+1} = \frac{n^2(1-t_k^2)}{n^2-1}\left(A_k - \frac{2(1+nt_k)}{(n+1)(1+t_k)}\frac{A_k g_k g_k^T A_k}{g_k^T A_k g_k}\right), \tag{15}$$

$$g_{k,i} = -u_k^T F_i u_k, \quad i = 1, \ldots, n, \tag{16}$$

$$t_k = -\frac{u_k^T F(x_k) u_k}{\sqrt{g_k^T A_k g_k}}. \tag{17}$$

여기서 u_k는 $u_k^T F(x_k) u_k \le 0$를 만족하는 임의의 벡터이다.

4.2 선형 제한조건의 경우

다음으로, 본 논문에서 제시한 알고리즘이 [1]에서 다루는, 선형 제한 조건을 가지는 가능성 문제를 그 특수형으로 포함함을 증명한다. 이 경우 제한조건은

$$Cx < B, \quad C \in R^{m \times n}, B \in R^{m \times 1} \tag{18}$$

이다. C의 i번째 행벡터를 $c_i \in R^n$, B의 i번째 원소를 $b_i \in R^1$ 이라 하자. [1]에 의해 $g_k \simeq c_i$ 로 설정하면 (19)-(21)을 얻는다.

$$F(x_k) = b_i - c_i^T x_k = b_i - \sum_{j=1}^{n} c_{ij} x_{k,j}, \tag{19}$$

$$g_{k,j} = -u_k^T F_j u_k = u_k^T c_{ij} u_k = c_{ij}\|u_k\|_2^2 \simeq c_{ij} \quad (\because \|u_k\|_2 = 1), \tag{20}$$

$$t_k = -\frac{u_k^T F(x_k) u_k}{\sqrt{g_k^T A_k g_k}} = \frac{-\|u_k\|^2 (b_i - c_i^T x_k)}{\sqrt{c_i^T A_k c_i}} = \frac{(c_i^T x_k - b_i)}{\sqrt{c_i^T A_k c_i}}. \tag{21}$$

선형 제한조건의 경우 (20), (21)이 성립하고, 이는 [1]의 경우와 일치하므로, (14)-(17)의 알고리즘은 선형 제한조건을 가지는 가능성 문제를 그 특수형으로 포함한다.

4.3 새 알고리즘의 수렴속도

(7)에서와 동일한 방법으로 (14)-(17)로 주어지는 알고리즘의 매 단계 부피감소율을 (22)와 같이 구할 수 있다.

$$\begin{aligned} det(A_{k+1}) &= \left(\frac{n^2(1-t_k^2)}{n^2-1}\right)^n \times det(A_k) \\ &\times det\left(I - \frac{2(1+nt_k)}{(n+1)(1+t)}\frac{g_k g_k^T A_k}{g_k^T A_k g_k}\right) \\ &= \left(\frac{n^2(1-t_k^2)}{n^2-1}\right)^n \times det(A_k) \\ &\times \left(1 - \frac{2(1+nt_k)}{(n+1)(1+t_k)}\right) \\ &= \left(\frac{n^2(1-t_k^2)}{n^2-1}\right)^n \times \frac{(n-1)(1-t_k)}{(n+1)(1+t_k)} \times det(A_k), \end{aligned}$$

$$\frac{S_{k+1}}{S_k} = (1-t_k)(1-t_k^2)^{\frac{n-1}{2}}\left(\frac{n}{n+1}\right)\left(\frac{n^2}{n^2-1}\right)^{\frac{n-1}{2}}. \tag{22}$$

$F(x_k) \le 0$이면 $0 \le t_k < 1$이므로 (22)에 의해 (14)-(17)의 알고리즘은 (7)의 알고리즘에 비해 수렴속도가 향상됨이 증명되었다.

V.모의실험

이제 간단한 LMI 가능성 문제에 대한 모의실험을 통해 본 논문에서 제시한 알고리즘의 성능을 검증한다. 모의실험에서는 R^2 에서 (23)으로 주어진 LMI 가능영역과 초기치 (-5000000, 15000)에 대해 기존의 알고리즘(Ⅲ)과 개선된 알고리즘(Ⅳ)에서 논한 알고리즘의 성능을 각각 비교하였다.

$$-1<x_i<1,\ i=1,2. \tag{23}$$

모의실험 결과, Ⅲ에서 소개한 기존의 알고리즘과 Ⅳ에서 제시한 개선된 알고리즘의 결과는 각각 그림 4, 그림 5와 같다.

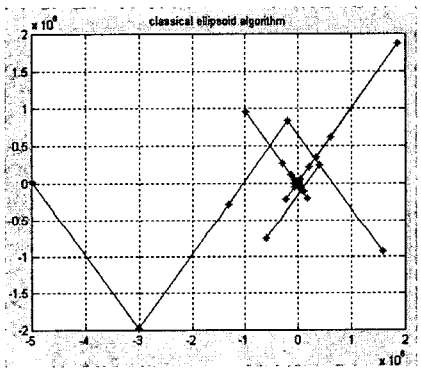

그림 4. 기존의 타원 알고리즘

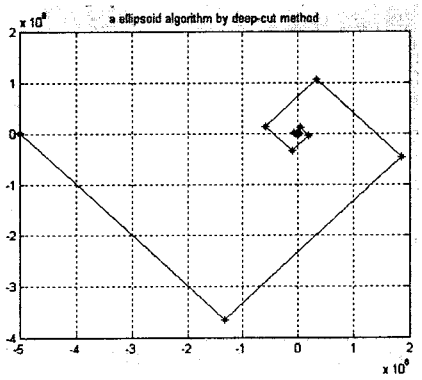

그림 5. 수정된 타원 알고리즘

기존의 알고리즘은 89번째 단계에서 (0.8894, -0.7141)에 도달한 반면, 동일한 가능성 문제에 대해 개선된 알고리즘은 29번째 단계에서 (0.3552, 0.5115)에 도달하였다. 따라서 본 논문에서 제시한 알고리즘의 성능이 Ⅲ의 알고리즘에 비해 우수함을 확인할 수 있다.

VI. 결론

본 논문에서는 LMI 가능성 문제를 풀기 위해 기존의 타원 알고리즘에, 선형계획법에서 알려진 딥-컷(deep-cut) 방법을 적용하고 그 수렴속도 향상을 증명과 모의실험을 통해 확인하였다. 따라서 간단하지만 비교적 느린 수렴속도를 가졌던 타원 알고리즘의 단점을 보완하고 성능향상에 기여할 수 있으리라 기대된다.

참고문헌

[1] Shu-Cherng Fang, Sarat Puthenpura, "*linear optimization and extensions*", p94-106, Prentice-Hall, 1993.

[2] Stephen Boyd, Laurent El Ghaoui, Eric Feron and Venkataramanan Balakrishnan, "*Linear Matrix Inequalities in System and Control Theory*", p13-14, Draft, June, 1993.

[3] Sharmila Shah, John E. Mitchell, Michael Kupferschmid, "An ellipsoid algorithm for equality-constrained nonlinear programs", Computers & operations research, 1999.

[4] Kemin zhou, John C. Doyle, "*Essentials of robust control*", Prentice-Hall, p14, 1998

[5] Chi-Tsong Chen, "Linear system theory and design", Oxford university press.,1999.

[6] Jeremy G. VanAntwerp, Richard D. Braatz, "A turorial on linear and bilinear inequalities", Journal of Process Control, 2000.