Applied Mathematics and Computation 219 (2013) 4176–4184

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Iterative solution to linear matrix inequality arising from time delay descriptor systems

Qing Wang [a,*], Pengcheng Xu [b], Edmund Y. Lam [a], Ngai Wong [a]

[a] Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong
[b] Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 10080, PR China

ARTICLE INFO

Keywords:
Descriptor system
Linear matrix equation (LME)
Linear matrix inequality (LMI)
Steepest descent method (SDM)
Time delay system (TDS)

ABSTRACT

Linear matrix inequalities (LMIs) are widely used to analyze the stability or performance of time delay descriptor systems (TDDSs). They are solved by the well known interior-point method (IPM) via minimizing a strictly convex function by transforming the matrix variable into an expanded vector variable. Newton's method is used to get the unique minimizer of the strictly convex function by the iteration involving its gradient and Hessian. The obvious disadvantage of the IPM is the high storage requirement for the Hessian. Hence, this often renders that the IPM cannot solve "large" LMI problems due to finite memory limit. To overcome this shortcoming, for the first time, an iterative algorithm based on the steepest descent method (SDM) is proposed to solve LMIs by keeping matrix variable form instead of transforming it to an expanded vector and without using Hessian matrix. The gradient of the proposed objective function is explicitly given by a matrix function with the same dimension of the original matrix variable. The efficiency of the proposed method is verified with numerical examples.

1. Introduction

Linear matrix inequalities (LMIs) are encountered in the stability or stabilizability, controller design and performance analysis of the time delay descriptor system (TDDS) Σ [1–5]:

$$\begin{aligned}\Sigma : &E\dot{x}(t) = Ax(t) + Bx(t-d),\\ &x_{t_0}(\theta) = x(t_0+\theta) = \psi(t), \quad \forall \theta \in [-d, 0], \quad (t_0, \psi) \in \mathbb{R}^+ \times \mathcal{C}([-d, 0], \mathbb{R}^n),\end{aligned}$$

where $x(t) \in \mathbb{R}^n$ is the state vector and d is the constant delay; $E \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$ are real constant matrices. $\mathcal{C}([-d, 0], \mathbb{R}^n)$ is the Banach space of continuous vector functions mapping the interval $[-d, 0]$ into $\mathbb{R}^n$ with the topology of uniform convergence and designates the norm of an element ψ in $\mathcal{C}([-d, 0], \mathbb{R}^n)$ by $\|\psi\|_c = sup_{\theta\in[-d,0]}|\psi(\theta)|$. The order of TDDS Σ is defined as n, the number of state vector $x(t)$. TDDS Σ can be used to model the transmission line for large RLC networks with time delay [6], networked control systems [7,8] and references therein. The matrix E is assumed to be nonsingular and $E^{-1}A$ is assumed to be Hurwitz (all eigenvalues of $E^{-1}A$ have negative real part). Of particular importance are the positive definite solutions of LMIs since they are used to test the stability [9–11] of the TDDS Σas well as to design controllers in H_∞ control [12–14].

* Corresponding author.
E-mail address: wangqing@eee.hku.hk (Q. Wang).

0096-3003/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2012.11.017

If there exist $P = P^T \in \mathbb{R}^{n\times n}$ and $Q = Q^T \in \mathbb{R}^{n\times n}$ satisfying LMI

$$f(X) = \begin{bmatrix} A^TPE + E^TPA + Q & E^TPB \\ B^TPE & -Q \end{bmatrix} < 0, \quad X = \begin{bmatrix} P \\ Q \end{bmatrix}, \quad P > 0 \text{ and } Q > 0, \tag{1}$$

then the TDDS Σ is stable by using the Lyapunov–Krasovskii functional

$$V(x) = x^T(t)E^TPEx(t) + \int_{t-d}^{t} x^T(s)Qx(s)ds.$$

Notice that $E^{-1}A$ being Hurwitz is necessary for (1) to have feasible solution X with $P > 0$ and $Q > 0$. One popular way to solve LMI (1) is based on the interior-point method (IPM) [15] together with Newton's method by two steps: (1) Since $X \in \mathbb{R}^{2n\times n}$ is a matrix not a vector, in order to use the IPM, the problem of finding X for LMI (1) is first transformed into minimizing a strictly convex function by transforming the matrix X to a single vector variable $y \in \mathbb{R}^{n(n+1)\times 1}$ containing all variables of X. (2) Newton's method, which is a powerful tool to find the minimizer of a nonlinear function, is used to get the unique minimizer of the objective function. However, it should be pointed out that the iteration by Newton's method needs the gradient with size $n(n+1)\times 1$ and Hessian with size $n(n+1)\times n(n+1)$ of the objective function. Furthermore, the order of TDDS Σ, n, especially from the electronic systems [16–19], usually is larger than 500. For example, if we employ the IPM to solve LMI (1) with $n = 500$, the size of the transformed vector y would be 250500×1 and the size of Hessian would be 250500×250500. Obviously it is impossible to deal with this Hessian in practice as it requires more than 460 GB computing memory. Hence, though Newton's method is theoretically perfect, it is of little practical use due to the high storage requirement coming from Hessian.

As the major difficulty in the IPM comes from the high storage requirement for the Hessian, it is reasonable to find an alternative method to get the solution of LMI (1) without using it. In this paper, we introduce an iterative method to minimize a strictly convex function based on the steepest descent method (SDM) (also called the gradient method [20,21]) to solve LMI (1). The proposed method has two advantages to avoid the shortcomings shown in the IPM: (1) The proposed iteration is for the matrix variable X not for the transformed vector variable y. (2) The new iteration only needs the gradient of the defined objective function without using its Hessian, which renders that the proposed method uses lower computing memory than the IPM. Although the convergence speed of the proposed method is slower than the IPM since the SDM could not provide the optimal direction and step length for every iteration, the proposed iterative method can be applied to LMI (1) with large n.

The rest of the paper is organized as follows. In Section 2, the shortcomings of the IPM are given and the proposed iterative algorithm is presented. Numerical examples to demonstrate the effectiveness of the proposed algorithm are shown in Section 3. Finally, Section 4 draws the conclusion.

Notation. Throughout this paper, the notation $X \geqslant Y$ (respectively, $X > Y$) for real symmetric matrices X and Y means that the matrix $X - Y$ is positive semi-definite (respectively, positive definite). $X = [x_{ij}] \succeq 0$ for matrix X denotes a nonnegative matrix with $x_{ij} \geqslant 0$. For a matrix $Q = [q_{ij}]$, $|Q| = [|q_{ij}|] \succeq 0$, where $|q_{ij}|$ are the absolute value q_{ij}, $i,j = 1,2,\ldots,n$. If α is a complex number, $|\alpha|$ is the modulus of α. M^T represents the transpose of the matrix M and if M is invertible, M^{-1} is the inverse of M. The vec of a matrix $S = [s_1 \quad s_2 \quad \cdots \quad s_n]$ is defined as $\text{vec}(S) = [s_1^T \quad s_2^T \quad \cdots \quad s_n^T]^T$. He$(X)$ means $X + X^T$. The notation $\|\cdot\|_F$ refers to the Frobenius norm and $\|\cdot\|$ denotes the consistent matrix norm. Identity matrices are invariably denoted by I and zero matrices are denoted by 0. The Kronecker product of matrices is denoted by $\otimes$ and the spectral radius of the matrix X is represented by $\rho(X)$.

2. Main results

2.1. Brief analysis of IPM

Let $\overline{P}_1,\ldots,\overline{P}_{\frac{n(n+1)}{2}}$ be a basis for symmetric $n\times n$ matrices and $\overline{Q}_1,\ldots,\overline{Q}_{\frac{n(n+1)}{2}}$ also be a basis for symmetric $n\times n$ matrices. So P and Q are linear combinations of $\overline{P}_1,\ldots,\overline{P}_m$ and $\overline{Q}_1,\ldots,\overline{Q}_m$, respectively,

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{12} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1n} & p_{2n} & \cdots & p_{nn} \end{bmatrix} = \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}\overline{P}_i,$$

$$Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{12} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{1n} & q_{2n} & \cdots & q_{nn} \end{bmatrix} = \sum_{i=1}^{n}\sum_{j=i}^{n} q_{ij}\overline{Q}_i.$$