

Research Article

A Modified Gradient Based Algorithm for Solving Matrix Equations $AXB + CX^T D = F$

Kanmin Wang, Zhibing Liu, and Chengfeng Xu

Department of Mathematics, Jiujiang University, Jiujiang 332005, China

Correspondence should be addressed to Zhibing Liu; liuzhibingjju@126.com

Received 22 September 2013; Accepted 16 January 2014; Published 27 February 2014

Academic Editor: Suh-Yuh Yang

Copyright © 2014 Kanmin Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we develop a modified gradient based algorithm for solving matrix equations $AXB + CX^T D = F$. Different from the gradient based method introduced by Xie et al., 2010, the information generated in the first half-iterative step is fully exploited and used to construct the approximate solution. Theoretical analysis shows that the new method converges under certain assumptions. Numerical results are given to verify the efficiency of the new method.

1. Introduction

Consider a linear matrix equation of the following form:

$$AXB + CX^T D = F, \quad (1)$$

where $A \in \mathcal{R}^{r \times m}$, $B \in \mathcal{R}^{m \times s}$, $C \in \mathcal{R}^{r \times n}$, $D \in \mathcal{R}^{m \times s}$, and $F \in \mathcal{R}^{r \times s}$ are the given constant matrices and $X \in \mathcal{R}^{m \times n}$ is the unknown matrix to be solved. A Sylvester equation $AX + X^T B = F$ is the special case of (1) with $r = s = n$ and $B = C = I_n$, the notation I_n is the identity matrix of $n \times n$. Such kind of problems frequently arise from many areas of applications in control and system theory [1], stability of linear systems [2], analysis of bilinear systems [3], power systems [4], signal and image processing [5], and so forth.

The exact solutions of matrix equations, such as Lyapunov and Sylvester matrix equations, can be obtained from matrix inversion by using the Kronecker product. The drawback of this approach is that considerable computational costs and storage requirements are needed, so that this approach is only applicable for small sized Sylvester equations. Some direct methods have also been proposed in [6–9] which are based on the idea of transforming the coefficient matrix into a Schur or Hessenberg form, by which the original equation can be solved by a backward substitution.

In the numerical linear community, iterative methods are becoming more and more popular. Several iterative schemes

for Sylvester equations have been proposed; see, for example, [10–15]. Recently, some efficient gradient based and least squares based iterative algorithms for solving generalized Sylvester equations and coupled (general coupled) Sylvester equations have been presented [16–28]. The basic idea of these approaches is based on a hierarchical identification principle [16–18], which regards the unknown matrix as the system parameter matrix to be identified and then constructs a recursive formula to approximate the unknown solution. Particularly, for general linear matrix equations of form (1), it is illustrated in [3, 4] that the unknown matrix to be identified can be computed by a gradient based iterative algorithm. The convergence properties of the methods are also investigated in [3]. In this paper, a modified gradient based iterative algorithm is proposed for solving linear matrix equations of form (1). The information generated in the first half-iterative step is fully exploited and used to construct the approximate solution by the modified method. The convergence condition of the method is analyzed. The numerical performance of the method is compared with the algorithms in [3, 4]. Numerical results show that the new method is efficient and robust.

The paper is organized as follows. In Section 2, the gradient based iterative method is recalled, and the modified gradient based method is introduced and analyzed in Section 3. In Section 4, performance of the modified gradient based method is compared with the original one. Finally, we conclude the paper in Section 5.

2. A Brief Review of the Gradient Based Iterative Method

We firstly recall an iterative method proposed by Xie et al. [3] for solving (1). The basic idea is regarding (1) as two linear matrix equations as follows:

$$AXB = F - CX^T D, \quad CX^T D = F - AXB. \quad (2)$$

Then, define two recursive sequences as follows:

$$X_k^{(1)} = X_{k-1} + \mu A^T (F - AX_{k-1} B - CX_{k-1}^T D) B^T, \quad (3)$$

$$X_k^{(2)} = X_{k-1} + \mu D (F - AX_{k-1} B - CX_{k-1}^T D)^T C, \quad (4)$$

where μ is the iterative step size. The above procedures can be regarded as two separate iterative procedures for solving two matrix equations in (3).

With $X_k^{(1)}$ and $X_k^{(2)}$ at hand, then the k th approximate solution X_k can be defined by taking the average of two approximate solutions, that is,

$$X_k = \frac{X_k^{(1)} + X_k^{(2)}}{2}. \quad (5)$$

By selecting an appropriate initial approximate solution X_0 , and using X_{k-1} to substitute $X_{k-1}^{(1)}$ in (4) and $X_{k-1}^{(2)}$ in (5), then the above (4)–(6) constitute the gradient based iterative method proposed in [3]. It is shown in [3] that the gradient based iterative algorithm converges as long as

$$0 < \mu \leq \frac{2}{\lambda_{\max}(AA^T) \lambda_{\max}(B^T B) + \lambda_{\max}(CC^T) \lambda_{\max}(D^T D)}, \quad (6)$$

where $\lambda_{\max}(AA^T)$ is the largest eigenvalue of AA^T .

According to lots of numerical experiments, GBI algorithm is computationally efficient. However, we observe that the GBI algorithm has some limitations. The convergent rate is slow and the stagnation will happen for ill-conditioned problem. Also, in [3], the authors pointed out that how to choose a best convergence factor is a subject to be studied and deserves further research. In this paper, we present the optimal convergence factor explicitly, and then propose a modified algorithm for solving the linear matrix equations (1).

3. A Modified Gradient Based Iterative Algorithm

The above GBI process can be accomplished by the following algorithm.

Algorithm 1 (see [3]). The gradient based iterative algorithm (GBI algorithm).

- (1) Give two initial approximate solutions $X_0^{(1)}$ and $X_0^{(2)}$

- (2) for $k = 1, 2, \dots$, until converges

- (3) $X_{k-1} = (X_{k-1}^{(1)} + X_{k-1}^{(2)})/2$

- (4) $X_k^{(1)} = X_{k-1} + \mu A^T (F - AX_{k-1} B - CX_{k-1}^T D) B^T$

- (5) $X_k^{(2)} = X_{k-1} + \mu D (F - AX_{k-1} B - CX_{k-1}^T D)^T C$

- (6) end.

In the step of computing $X_k^{(2)}$, the last approximate solution $X_k^{(1)}$ has been computed. Hence, we can use the information of $X_k^{(1)}$ to update the X_{k-1} and present a modification of GBI algorithm.

Algorithm 2. The modified gradient based iterative algorithm (MGBI algorithm).

- (1) Give two initial approximate solutions $X_0^{(1)}$ and $X_0^{(2)}$

- (2) for $k = 1, 2, \dots$, until converges

- (3) $X_{k-1} = (X_{k-1}^{(1)} + X_{k-1}^{(2)})/2$

- (4) $X_k^{(1)} = X_{k-1} + \mu A^T (F - AX_{k-1} B - CX_{k-1}^T D) B^T$

- (5) $X_{k-1} = (X_k^{(1)} + X_{k-1}^{(2)})/2$

- (6) $X_k^{(2)} = X_{k-1} + \mu D (F - AX_{k-1} B - CX_{k-1}^T D)^T C$

- (7) end.

Let $B \in \mathcal{R}^{n \times s}$, $D = [d_1, d_2, \dots, d_s]$, $d_i \in \mathcal{R}^m$, $X = [x_1, x_2, \dots, x_n]$, $\text{vec}(X) = [x_1^T, \dots, x_n^T]^T$, and

$$S = B^T \otimes A + \begin{bmatrix} C \otimes d_1^T \\ C \otimes d_2^T \\ \vdots \\ C \otimes d_s^T \end{bmatrix} \in \mathcal{R}^{rs \times mn}. \quad (7)$$

Lemma 3 (see [3]). *The Sylvester equation given by (1) has a unique solution if and only if $\text{rank } S, \text{vec}(F) = \text{rank}[S] = mn$, in this case, the unique solution is given by*

$$\text{vec}(X) = (S^T S)^{-1} S^T \text{vec}(F). \quad (8)$$

The corresponding homogeneous equation $AXB + CX^T D = 0$ has a unique zero solution $X = 0$.

The following result discusses the convergence conditions of the Algorithm 2.

Theorem 4. *If the linear matrix equation (1) has a unique solution X and*

$$0 < \mu < \min \left\{ \frac{2}{[\lambda_{\max}(AA^T) \lambda_{\max}(B^T B)]}, \frac{2}{[\lambda_{\max}(CC^T) \lambda_{\max}(D^T D)]} \right\}, \quad (9)$$

then the iterative sequence X_k generated by Algorithm 2 converges to X ; that is, $\lim_{k \rightarrow \infty} X_k = X$; or the error $E_k = X_k - X$ converges to zero for any initial value X_0 .

Proof. In the following discussions, we always assume that the Frobenius matrix norm is used. For the clarity of proof, we introduce another variable \widehat{X}_{k-1} instead of X_{k-1} in the fifth iteration. Define the following error matrices:

$$\begin{aligned} E_k &= X_k - X, & E_k^{(1)} &= X_k^{(1)} - X, \\ E_k^{(2)} &= X_k^{(2)} - X, & \widehat{E}_k &= \widehat{X}_k - X, \\ \xi_k &= AE_{k-1}B, & \eta_k &= CE_{k-1}^T D, \\ \delta_k &= A\widehat{E}_{k-1}B, & \zeta_k &= C\widehat{E}_{k-1}^T D. \end{aligned} \quad (10)$$

By using (1) and (10), the proof of the following equalities is trivial:

$$\begin{aligned} E_k^{(1)} &= E_{k-1} + \mu A^T (-\xi_k - \eta_k) B^T, \\ E_k^{(2)} &= \widehat{E}_{k-1} + \mu D (-\delta_k^T - \zeta_k^T) C. \end{aligned} \quad (11)$$

Taking the Frobenius norm of both sides of (11), it follows that

$$\begin{aligned} \|E_k^{(1)}\|^2 &= \text{tr} \left[(E_k^{(1)})^T E_k^{(1)} \right] \\ &= \|E_{k-1}\|^2 + 2\mu \text{tr} \left\{ (E_{k-1})^T A^T [-\xi_k - \eta_k] B^T \right\} \\ &\quad + \mu^2 \|A^T [-\xi_k - \eta_k] B^T\|^2 \\ &\leq \|E_{k-1}\|^2 + 2\mu \text{tr} \left\{ \xi_k^T [-\xi_k - \eta_k] \right\} \\ &\quad + \mu^2 \lambda_{\max}(AA^T) \lambda_{\max}(B^T B) \|\xi_k + \eta_k\|^2, \\ \|E_k^{(2)}\|^2 &= \text{tr} \left[(E_k^{(2)})^T E_k^{(2)} \right] \\ &\leq \|\widehat{E}_{k-1}\|^2 + 2\mu \text{tr} \left\{ [-\delta_k^T - \zeta_k^T] \zeta_k \right\} \\ &\quad + \mu^2 \lambda_{\max}(CC^T) \lambda_{\max}(D^T D) \|\delta_k + \zeta_k\|^2. \end{aligned} \quad (12)$$

From $E_k = [E_k^{(1)} + E_k^{(2)}]/2$, we have

$$\begin{aligned} \|E_k\|^2 &\leq \frac{\|E_k^{(1)}\|^2 + \|E_k^{(2)}\|^2}{2} \\ &\leq \frac{\|E_{k-1}\|^2}{2} + \mu \text{tr} \left\{ \xi_k^T [-\xi_k - \eta_k] \right\} \\ &\quad + \frac{\mu^2}{2} \lambda_{\max}(AA^T) \lambda_{\max}(B^T B) \|\xi_k + \eta_k\|^2 \end{aligned}$$

$$\begin{aligned} &+ \frac{\|\widehat{E}_{k-1}\|^2}{2} + \mu \text{tr} \left\{ [-\delta_k^T - \zeta_k^T] \zeta_k \right\} \\ &+ \frac{\mu^2}{2} \lambda_{\max}(CC^T) \lambda_{\max}(D^T D) \|\delta_k + \zeta_k\|^2 \\ &\leq \frac{\|E_{k-1}\|^2}{2} \\ &\quad - \mu \left[1 - \frac{\mu}{2} \lambda_{\max}(AA^T) \lambda_{\max}(B^T B) \right] \|\xi_k + \eta_k\|^2 \\ &\quad + \frac{\|\widehat{E}_{k-1}\|^2}{2} \\ &\quad - \mu \left[1 - \frac{\mu}{2} \lambda_{\max}(CC^T) \lambda_{\max}(D^T D) \right] \|\delta_k + \zeta_k\|^2 \\ &\leq \frac{\|E_0\|^2}{2^k} \\ &\quad - \mu \left[1 - \frac{\mu}{2} \lambda_{\max}(AA^T) \lambda_{\max}(B^T B) \right] \sum_{i=1}^k \|\xi_i + \eta_i\|^2 \\ &\quad + \sum_{i=1}^k \frac{\|\widehat{E}_{k-i}\|^2}{2^i} \\ &\quad - \mu \left[1 - \frac{\mu}{2} \lambda_{\max}(CC^T) \lambda_{\max}(D^T D) \right] \sum_{i=1}^k \|\delta_i + \zeta_i\|^2. \end{aligned} \quad (13)$$

Obviously, $\sum_{i=1}^k (\|\widehat{E}_{k-i}\|^2/2^i) < \infty$. In fact, the iterative sequence \widehat{X}_k , $k = 0, 1, \dots$ generated by Algorithm 2 can also be viewed as the sequence generated by the double-side iteration in [3], so $\lim_{k \rightarrow \infty} \widehat{X}_k = 0$. As $0 < \mu < \min\{2/[\lambda_{\max}(AA^T)\lambda_{\max}(B^T B)], 2/[\lambda_{\max}(CC^T)\lambda_{\max}(D^T D)]\}$, we have

$$\sum_{i=1}^{\infty} \|\xi_i + \eta_i\|^2 < \infty, \quad \sum_{i=1}^{\infty} \|\delta_i + \zeta_i\|^2 < \infty. \quad (14)$$

It follows that

$$\xi_k + \eta_k \rightarrow 0, \quad \text{as } k \rightarrow \infty, \quad (15)$$

or

$$AE_{k-1}B + CE_{k-1}^T D \rightarrow 0, \quad \text{as } k \rightarrow \infty. \quad (16)$$

According to Lemma 3, we have $E_{k-1} \rightarrow 0$ as $k \rightarrow \infty$. \square

4. Numerical Experiments

Example 5. Consider the matrix equation $AXB + CX^T D = F$ with

$$\begin{aligned} A &= \begin{pmatrix} 2 & 5 \\ 4 & -7 \end{pmatrix}, & B &= \begin{pmatrix} 6 & -3 \\ 1 & 2 \end{pmatrix}, & C &= \begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix}, \\ D &= \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}, & F &= \begin{pmatrix} 317 & 9 \\ 41 & 27 \end{pmatrix}. \end{aligned} \quad (17)$$

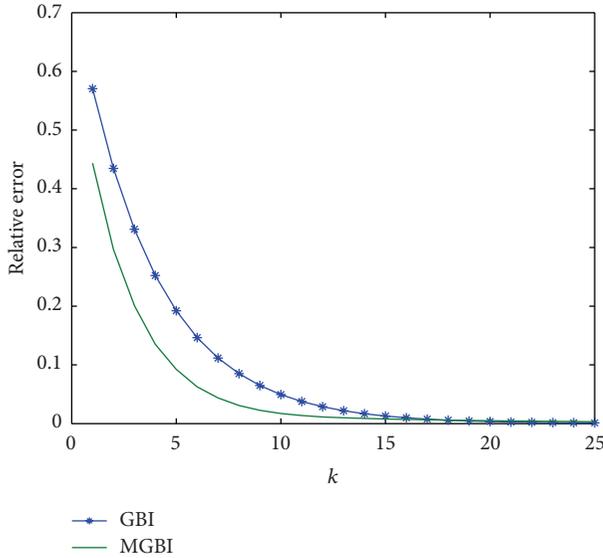


FIGURE 1: Comparison of convergence curves.

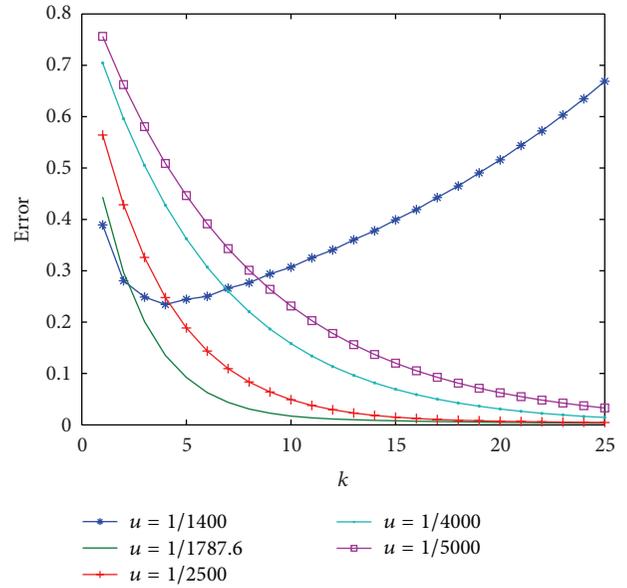


FIGURE 2: Comparison of convergence curves.

From (8), the exact solution is

$$X = \begin{pmatrix} 7 & 5 \\ 4 & 3 \end{pmatrix}. \tag{18}$$

The coefficient matrices used in this example are taken from [3]. Taking $X_0 = 10^{-6}I_2$, we apply the GBI algorithm and MGBI algorithm to compute X_k ; the convergence factor μ is set to be $2/[\lambda_{\max}(AA^T)\lambda_{\max}(BB^T) + \lambda_{\max}(CC^T)\lambda_{\max}(DD^T)] = 1/1983.1$ in GBI algorithm and to be $\min\{2/[\lambda_{\max}(AA^T)\lambda_{\max}(BB^T)], 2/[\lambda_{\max}(CC^T)\lambda_{\max}(DD^T)]\} = 1/1787.6$ in MGBI algorithm. The relative error $\delta := \|X_k - X\|/\|X\|$ is recorded and plotted in Figure 1 by MATLAB command semilogy. From the figure, we can see that the MGBI algorithm converges faster than the GBI algorithm.

Remark. The choice of the convergence factor μ is an important issue. We experimentally study its influence on the convergence. The effect of changing of the convergence factor μ for MGBI algorithms in Example ?? is illustrated in Figure 2. We see that

$$\mu = \min \left\{ \frac{2}{[\lambda_{\max}(AA^T)\lambda_{\max}(BB^T)]}, \frac{2}{[\lambda_{\max}(CC^T)\lambda_{\max}(DD^T)]} \right\} \tag{19}$$

$$= \frac{1}{1787.6}$$

is a better convergence factor. However, the convergence factor is problem dependent, so seeking a best convergence factor is still a difficult task.

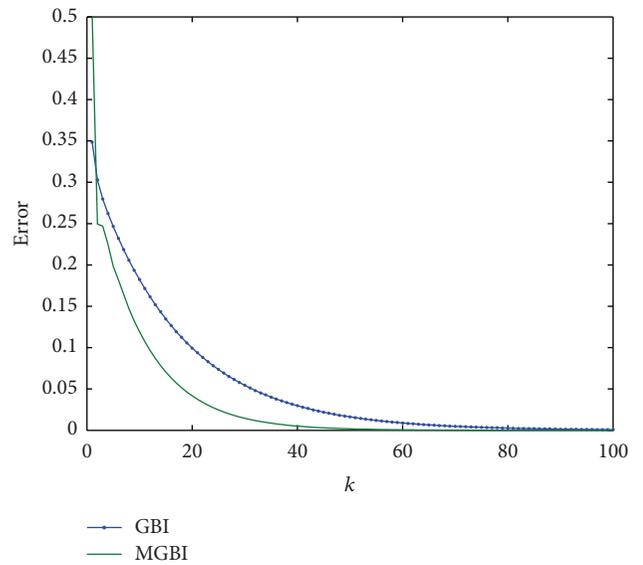


FIGURE 3: Comparison of convergence curves.

Example 6. Suppose that $AX + X^T B = F$, where

$$A = \begin{pmatrix} 1 & 1 \\ 2 & -1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \quad F = \begin{pmatrix} 8 & 8 \\ 5 & 2 \end{pmatrix}. \tag{20}$$

Then, the solution of X from (8) is

$$X = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}. \tag{21}$$

The coefficient matrices used in this example are taken from [4]. Taking $X_0 = 10^{-6}I_2$, we apply the GBI algorithm and MGBI algorithm to compute X_k , the convergence factor μ is set to be $2/[\lambda_{\max}(AA^T) + \lambda_{\max}(BB^T)] = 0.27$ in GBI algorithm

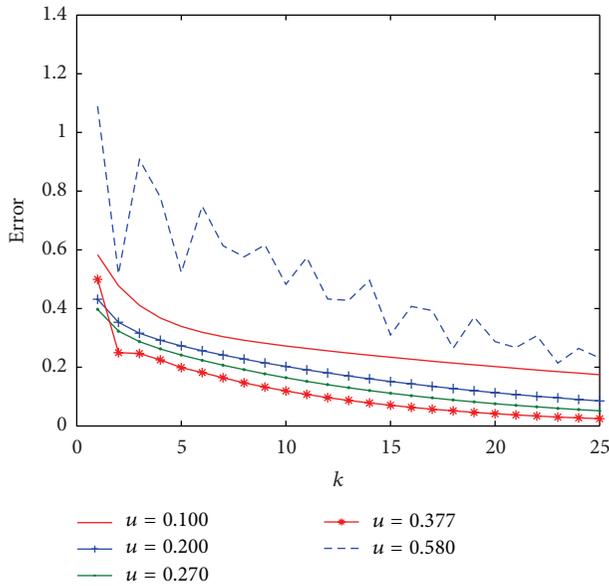


FIGURE 4: Comparison of convergence curves.

and to be $\min\{2/\lambda_{\max}(AA^T), 2/\lambda_{\max}(BB^T)\} = 0.377$ in MGBI algorithm. The relative error $\delta := \|X_k - X\|/\|X\|$ is recorded in Figure 3. From the figure, we can also see that the MGBI algorithm converges faster than the GBI algorithm.

The effect of changing of the convergence factor μ for MGBI algorithms in Example 6 is illustrated in Figure 4.

5. Conclusions

In this paper, a modified gradient based iteration (MGBI) method is proposed for linear matrix equation. The convergence of MGBI is analyzed. The choice of parameter μ is an important issue, and its influence is experimentally studied. The principle idea of this paper can be extended to the more general setting like generalized (coupled) Sylvester matrix equations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] R. Bhatia and P. Rosenthal, "How and why to solve the operator equation $AX - XB = Y$," *The Bulletin of the London Mathematical Society*, vol. 29, no. 1, pp. 1–21, 1997.
- [2] B. N. Datta, *Numerical Methods for Linear Control Systems*, Elsevier Academic Press, 2003.
- [3] L. Xie, Y. Liu, and H. Yang, "Gradient based and least squares based iterative algorithms for matrix equations $AXB + CX^T D = F$," *Applied Mathematics and Computation*, vol. 217, no. 5, pp. 2191–2199, 2010.
- [4] L. Xie, J. Ding, and F. Ding, "Gradient based iterative solutions for general linear matrix equations," *Computers & Mathematics with Applications*, vol. 58, no. 7, pp. 1441–1448, 2009.
- [5] J. Ding, Y. Liu, and F. Ding, "Iterative solutions to matrix equations of the form $A_i X B_i = F_i$," *Computers & Mathematics with Applications*, vol. 59, no. 11, pp. 3500–3507, 2010.
- [6] G. H. Golub, S. Nash, and C. Van Loan, "A Hessenberg-Schur method for the problem $AX + XB = C$," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 909–913, 1979.
- [7] R. H. Bartels and G. W. Stewart, "Algorithm 432: solution of the matrix equation $AX - XB = C$," *Communications of the ACM*, vol. 15, pp. 820–826, 1972.
- [8] D. C. Sorensen and Y. Zhou, "Direct methods for matrix Sylvester and Lyapunov equations," *Journal of Applied Mathematics*, no. 6, pp. 277–303, 2003.
- [9] W. H. Enright, "Improving the efficiency of matrix operations in the numerical solution of stiff ordinary differential equations," *Transactions on Mathematical Software, Association for Computing Machinery*, vol. 4, no. 2, pp. 127–136, 1978.
- [10] D. Calvetti and L. Reichel, "Application of ADI iterative methods to the restoration of noisy images," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 165–186, 1996.
- [11] D. Y. Hu and L. Reichel, "Krylov-subspace methods for the Sylvester equation," *Linear Algebra and its Applications*, vol. 172, pp. 283–313, 1992.
- [12] P. Kirrinnis, "Fast algorithms for the Sylvester equation $AX - XB^T = C$," *Theoretical Computer Science*, vol. 259, no. 1-2, pp. 623–638, 2001.
- [13] G. Starke and W. Niethammer, "SOR for $AX - XB = C$," *Linear Algebra and its Applications*, vol. 154/156, pp. 355–375, 1991.
- [14] Q. Niu, X. Wang, and L.-Z. Lu, "A relaxed gradient based algorithm for solving Sylvester equations," *Asian Journal of Control*, vol. 13, no. 3, pp. 461–464, 2011.
- [15] Z.-Z. Bai, "On Hermitian and skew-Hermitian splitting iteration methods for continuous Sylvester equations," *Journal of Computational Mathematics*, vol. 29, no. 2, pp. 185–198, 2011.
- [16] F. Ding and T. Chen, "On iterative solutions of general coupled matrix equations," *SIAM Journal on Control and Optimization*, vol. 44, no. 6, pp. 2269–2284, 2006.
- [17] F. Ding and T. Chen, "Iterative least-squares solutions of coupled Sylvester matrix equations," *Systems & Control Letters*, vol. 54, no. 2, pp. 95–107, 2005.
- [18] F. Ding and T. Chen, "Gradient based iterative algorithms for solving a class of matrix equations," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216–1221, 2005.
- [19] F. Ding, P. X. Liu, and J. Ding, "Iterative solutions of the generalized Sylvester matrix equations by using the hierarchical identification principle," *Applied Mathematics and Computation*, vol. 197, no. 1, pp. 41–50, 2008.
- [20] F. Ding, Y. Shi, and T. Chen, "Gradient-based identification methods for Hammerstein nonlinear ARMAX models," *Nonlinear Dynamics*, vol. 45, no. 1-2, pp. 31–43, 2006.
- [21] F. Ding, P. X. Liu, and G. Liu, "Gradient based and least-squares based iterative identification methods for OE and OEMA systems," *Digital Signal Processing*, vol. 20, no. 3, pp. 664–677, 2010.
- [22] Y. Liu, J. Sheng, and R. Ding, "Convergence of stochastic gradient estimation algorithm for multivariable ARX-like systems," *Computers & Mathematics with Applications*, vol. 59, no. 8, pp. 2615–2627, 2010.
- [23] Y. Liu, Y. Xiao, and X. Zhao, "Multi-innovation stochastic gradient algorithm for multiple-input single-output systems using the auxiliary model," *Applied Mathematics and Computation*, vol. 215, no. 4, pp. 1477–1483, 2009.

- [24] F. Ding, G. Liu, and X. P. Liu, "Parameter estimation with scarce measurements," *Automatica*, vol. 47, no. 8, pp. 1646–1655, 2011.
- [25] F. Ding and T. Chen, "Performance analysis of multi-innovation gradient type identification methods," *Automatica*, vol. 43, no. 1, pp. 1–14, 2007.
- [26] F. Ding, Y. J. Liu, and B. Bao, "Gradient based and least squares based iterative estimation algorithms for multi-input multi-output systems," *Proceedings of the Institution of Mechanical Engineers I: Journal of Systems and Control Engineering*, vol. 226, no. 1, pp. 43–55, 2012.
- [27] X. Wang, L. Dai, and D. Liao, "A modified gradient based algorithm for solving Sylvester equations," *Applied Mathematics and Computation*, vol. 218, no. 9, pp. 5620–5628, 2012.
- [28] J. Zhou, R. Wang, and Q. Niu, "A preconditioned iteration method for solving Sylvester equations," *Journal of Applied Mathematics*, vol. 2012, Article ID 401059, 12 pages, 2012.