SIAM J. MATRIX ANAL. APPL.
Vol. 27, No. 1, pp. 238–255

A GLOBAL SOLUTION FOR THE STRUCTURED TOTAL LEAST SQUARES PROBLEM WITH BLOCK CIRCULANT MATRICES*

AMIR BECK† AND AHARON BEN-TAL†

Abstract. We study the structured total least squares (STLS) problem of system of linear equations $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ has a block circulant structure with N blocks. We show that by applying the discrete Fourier transform (DFT), the STLS problem decomposes into N *unstructured* total least squares (TLS) problems. The N solutions of these problems are then assembled to generate the optimal global solution of the STLS problem. Similar results are obtained for elementary block circulant matrices. Here the optimal solution is obtained by assembling two solutions: one of an *unstructured* TLS problem and the second of a multidimensional TLS problem.

Key words. structured total least squares, block circulant matrices, discrete Fourier transform

AMS subject classifications. 65F20, 65T50

DOI. 10.1137/040612233

1. Introduction.

Many problems in data fitting and estimation give rise to an overdetermined system of linear equations $\mathbf{Ax} \approx \mathbf{b}$, where both the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ are contaminated by noise. The total least squares (TLS) approach to this problem [4, 1, 6] is to seek a perturbation matrix $\Delta\mathbf{A}$ and a perturbation vector $\Delta\mathbf{b}$ that minimize $\|\Delta\mathbf{A}\|^2 + \|\Delta\mathbf{b}\|^2$ subject to the consistency equation $\mathbf{b} - \Delta\mathbf{b} \in \text{Range}(\mathbf{A} - \Delta\mathbf{A})$.[1] The TLS approach was extensively used in a variety of scientific disciplines such as signal processing, automatic control, statistics, physics, economic, biology, and medicine. One of the main reasons for the wide use of TLS is the fact that the problem has essentially an explicit solution, expressed by the singular value decomposition (SVD) of the augmented matrix $(\mathbf{A}, \mathbf{b})$ [4, 1, 6] (cf. section 2).

In many applications, the matrix $\mathbf{A}$ has a specific structure, e.g., Toeplitz or Hankel, which imposes a requirement on the perturbation matrix $\Delta\mathbf{A}$ to possess a corresponding special structure. The TLS solution does not take into account this requirement, and consequently many methods addressing the structured TLS (STLS) problem were introduced in the literature [7, 8, 9, 5, 10, 11].

The STLS problem, even for *linearly structured* matrix $\Delta\mathbf{A}$ (i.e., a structure that can be represented as $\mathcal{L}(\Delta\mathbf{A}) = 0$, where $\mathcal{L}$ is a linear operator) is generically a nonconvex problem, and as a result the algorithms designed to solve it converge at best to a local solution [7, 9, 5]. This state of affairs is prevailing even for the special case of Toeplitz–Hankel structure [10, 11]. An exception is the case where some given columns of $\mathbf{A}$ must remain fixed (i.e., the corresponding columns of $\Delta\mathbf{A}$ are zero). A global solution for this case is obtained in [12] in terms of the SVD of an appropriate matrix.

---

*Received by the editors July 26, 2004; accepted for publication (in revised form) by S. Van Huffel January 27, 2005; published electronically August 31, 2005. This research was partially supported by BSF grant 2002038.

http://www.siam.org/journals/simax/27-1/61223.html

†MINERVA Optimization Center, Department of Industrial Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (becka@tx.technion.ac.il, morbt@ie.technion.ac.il).

[1]Here and elsewhere in this paper a matrix norm is always the Frobenius norm, and a vector norm is the Euclidean one.

In this paper we study the STLS problem, where the matrix $\mathbf{A}$ has either a *block circulant* (BC) structure,

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{N-1} \\ \mathbf{A}_{N-1} & \mathbf{A}_0 & \cdots & \mathbf{A}_{N-2} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_0 \end{pmatrix},$$

or a (more special) *elementary block circulant* structure (EBC),

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{A}_0 & \cdots & \mathbf{A}_1 \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_1 & \cdots & \mathbf{A}_0 \end{pmatrix},$$

where $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1}$ are $m \times n$ matrices. A particular example from signal processing, which gives rise to elementary block circulant structure, is described next.

Consider the problem of estimating N unknown vectors $\mathbf{x}_k, 0 \le k \le N-1$, from N vector observations $\mathbf{b}_k, 0 \le k \le N-1$, where each observation vector $\mathbf{b}_k$ is related to all of the parameter vectors $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{N-1}$ through the linear model

$$\mathbf{b}_k = \mathbf{A}_0\mathbf{x}_k + \sum_{i\neq k} \mathbf{A}_1\mathbf{x}_i + \mathbf{w}_k, \quad 0 \le k \le N-1. \tag{1}$$

Here $\mathbf{A}_0$ and $\mathbf{A}_1$ represent the *within channel* and *cross channel* transfer matrices, respectively, and $\mathbf{w}_k$ is the kth noise vector. For the two channel case ($N = 2$), this system is illustrated in Figure 1.

System (1) reflects the situation where the effect of all of the interfering vectors $\mathbf{x}_i, i \neq k$ on the kth output $\mathbf{b}_k$ is the same (i.e., independent of k).

Systems with BC structure appear in the context of multichannel signal estimation [13, 14], image restoration [15], cyclic convolution filter banks [3], texture synthesis and recognition [16], and more.

We will present efficient algorithms, which obtain a (global) solution for both the BC and the EBC cases. The analysis relies heavily on the theory of discrete Fourier transform (DFT) for block circulant matrices. Elements of this theory that are needed for our purposes are collected in section 3. We show that for the BC case, under the DFT, the original STLS problem, which is of size $Nm \times Nn$, decomposes into N (unstructured) TLS problems of size $m \times n$. The solution is then obtained by solving the N small problems (possibly in parallel), using the SVD of each system, and then taking the inverse DFT. We thus obtain that the solution of the STLS problem with a BC matrix is explicitly expressed by N singular value decomposition of N appropriate matrices (N being the number of different blocks in $\mathbf{A}$). The solution of the EBC is similarly derived with one exception. In the EBC case the STLS problem decomposes under the DFT to two smaller problems: a TLS problem and a multidimensional TLS problem (cf. section 2).

The paper is organized as follows. In section 2 we briefly review both the classical TLS problem and the multidimensional TLS problem. We recall the SVD-based solution of both problems. Section 3 contains a summary of the results on the DFT of BC matrices and block vectors. Sections 4 and 5 present the solution of the STLS problem with BC and EBC matrices, respectively. Section 6 presents computational

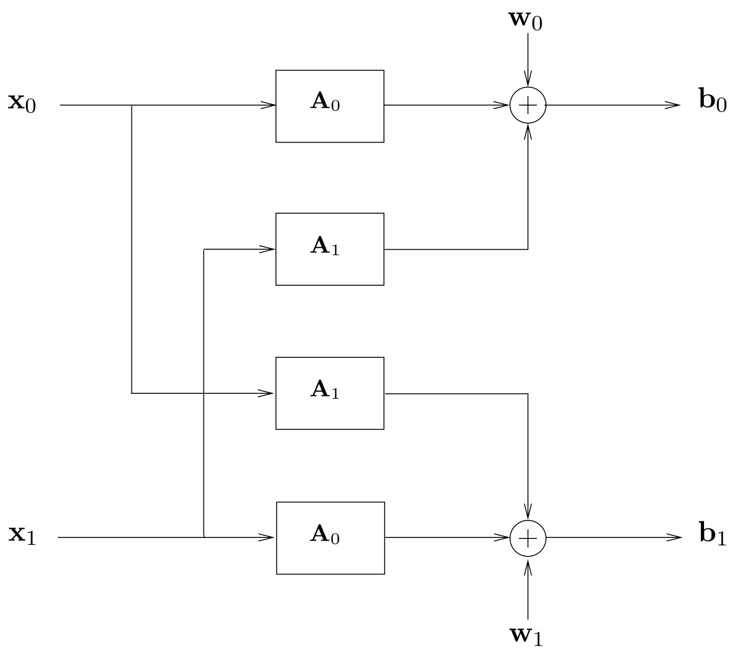

FIG. 1. *Two channel model.*

results that demonstrate the fact that the algorithm devised in this paper gives the global optimum while other algorithms in the literature do not necessarily converge to the global optimum.

The results in this paper are valid both for the real and complex case. We denote by $\mathbb{F}$ the real field ($\mathbb{R}$) or the complex field ($\mathbb{C}$). Vectors in $\mathbb{F}^m$ are denoted by boldface lowercase letters, e.g., $\mathbf{y}$. Matrices in $\mathbb{F}^{m\times n}$ are denoted by boldface uppercase letters e.g., $\mathbf{A}$. The Hermitian conjugate and the transpose of a matrix $\mathbf{A}$ are denoted by $\mathbf{A}^*$ and $\mathbf{A}^T$, respectively. Note that when $\mathbb{F}=\mathbb{R}$, $\mathbf{A}^*=\mathbf{A}^T$. The boldface letter $\mathbf{i}$ denotes $\sqrt{-1}$. For a positive integer N and an integer j, we denote by $[j]_N$ the value of j modulo N, e.g., $[N]_N=0,[-2]_N=N-2$.

2. Review of the TLS Method. For the sake of completeness, we briefly review the known results on the (unstructured) TLS and the multidimensional TLS problems [4, 1, 6] . Given a linear system $\mathbf{Ax}\approx\mathbf{b}$, where $\mathbf{A}\in\mathbb{F}^{m\times n}(m>n),\mathbf{b}\in\mathbb{F}^m$, and $\mathbf{x}\in\mathbb{F}^n$. The TLS problem is to find a perturbation matrix $\Delta\mathbf{A}\in\mathbb{F}^{m\times n}$ and a perturbation vector $\Delta\mathbf{b}\in\mathbb{F}^m$ of minimum norm such that the system $(\mathbf{A}-\Delta\mathbf{A})\mathbf{x}=\mathbf{b}-\Delta\mathbf{b}$ is consistent. More precisely, for some positive constant $\alpha>0$ we seek to solve the following minimization problem:

$$\text{(TLS)}\qquad \begin{aligned} &\min_{\Delta\mathbf{A},\Delta\mathbf{b},\mathbf{x}} \ \|\Delta\mathbf{A}\|^2+\alpha\|\Delta b\|^2\\ &\text{subject to } (\mathbf{A}-\Delta\mathbf{A})\mathbf{x}=\mathbf{b}-\Delta\mathbf{b}.\end{aligned}$$

The algorithm for the solution of this problem was derived in [4] and is based on *one* SVD calculation.

Algorithm TLS (see [4]).
Input: $\mathbf{A} \in \mathbb{F}^{m\times n}, \mathbf{b} \in \mathbb{F}^m, \alpha > 0$.
Output: $\Delta\mathbf{A} \in \mathbb{F}^{m\times n}, \Delta\mathbf{b} \in \mathbb{F}^m, \mathbf{x} \in \mathbb{F}^n$.

1. Calculate the SVD $(\mathbf{A}, \sqrt{\alpha}\mathbf{b}) = \mathbf{U}\Sigma\mathbf{V}^*$, where $\mathbf{V} \in \mathbb{F}^{(n+1)\times(n+1)}$ is a unitary matrix, $\mathbf{U} \in \mathbb{F}^{m\times(n+1)}$ is a matrix that satisfies $\mathbf{U}^*\mathbf{U} = \mathbf{I}$, and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{n+1})$, where $\sigma_1, \ldots, \sigma_{n+1} > 0$.
2. If $\sigma_n(\mathbf{A}) > \sigma_{n+1}(\mathbf{A}, \sqrt{\alpha}\mathbf{b})$, then the solution of (TLS) is given by
$$(\Delta\mathbf{A}, \Delta\mathbf{b}) = \sigma_{n+1}\mathbf{u}_{n+1}\mathbf{v}_{n+1}^T \mathrm{diag}(\underbrace{1, 1, \ldots, 1}_{n \text{ times}}, \sqrt{\alpha}),$$
$$\mathbf{x} = -\frac{1}{\sqrt{\alpha}V_{n+1,n+1}}(V_{1,n+1}, \ldots, V_{n,n+1})^T,$$
where $\mathbf{u}_{n+1}$ and $\mathbf{v}_{n+1}$ are the $(n+1)$th columns of $\mathbf{U}$ and $\mathbf{V}$, respectively, and for every i, j, $V_{i,j}$ is the (i,j)th component of $\mathbf{V}$.

A known generalization of the TLS problem deals with the case in which we have multiple right-hand side vectors; i.e., we are given k linear systems $\mathbf{A}\mathbf{x}_1 \approx \mathbf{b}_1, \mathbf{A}\mathbf{x}_2 \approx \mathbf{b}_2, \ldots, \mathbf{A}\mathbf{x}_k \approx \mathbf{b}_k$. Here we seek to find minimum weight perturbations $\Delta\mathbf{A} \in \mathbb{F}^{m\times n}$ and $\Delta\mathbf{b}_1, \ldots, \Delta\mathbf{b}_k \in \mathbb{F}^m$ such that the k linear systems $(\mathbf{A} - \Delta\mathbf{A})\mathbf{x}_i = \mathbf{b}_i - \Delta\mathbf{b}_i,\ i = 1, 2, \ldots, k$, are consistent. Denote $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k)$ and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$. The problem is equivalent to finding a minimal weight perturbations $\Delta\mathbf{A} \in \mathbb{F}^{m\times n}$ and $\Delta\mathbf{B} \in \mathbb{F}^{m\times k}$ such that the system $(\mathbf{A} - \Delta\mathbf{A})\mathbf{X} = \mathbf{B} - \Delta\mathbf{B}$ is consistent. This is the *multidimensional TLS* (MTLS):

$$\text{(MTLS)} \qquad \begin{aligned} &\min_{\Delta\mathbf{A}, \Delta\mathbf{B}, \mathbf{X}} \|\Delta\mathbf{A}\|^2 + \alpha\|\Delta\mathbf{B}\|^2 \\ &\text{subject to} (\mathbf{A} - \Delta\mathbf{A})\mathbf{X} = \mathbf{B} - \Delta\mathbf{B}. \end{aligned}$$

Algorithm MTLS (see [6, 1]).
Input: $\mathbf{A} \in \mathbb{F}^{m\times n}, \mathbf{B} \in \mathbb{F}^{m\times k}, \alpha > 0$.
Output: $\Delta\mathbf{A} \in \mathbb{F}^{m\times n}, \Delta\mathbf{B} \in \mathbb{F}^{m\times k}, \mathbf{X} \in \mathbb{F}^{n\times k}$.

1. Calculate the SVD $(\mathbf{A}, \sqrt{\alpha}\mathbf{B}) = \mathbf{U}\Sigma\mathbf{V}^*$, where here
$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_1 & \mathbf{O} \\ \mathbf{O} & \Sigma_2 \end{pmatrix},$$
and where $\mathbf{U}_1 \in \mathbb{F}^{m\times n}, \mathbf{U}_2 \in \mathbb{F}^{m\times k}, \mathbf{V}_{11} \in \mathbb{F}^{n\times n}, \mathbf{V}_{12} \in \mathbb{F}^{n\times k}, \mathbf{V}_{21} \in \mathbb{F}^{k\times n}, \mathbf{V}_{22} \in \mathbb{F}^{k\times k}, \Sigma_1 \in \mathbb{F}^{n\times n}$, and $\Sigma_2 \in \mathbb{F}^{k\times k}$. Σ_1 and Σ_2 are both diagonal matrices with real positive diagonal, $\mathbf{V}$ is a unitary matrix, and $\mathbf{U}$ satisfies $\mathbf{U}^*\mathbf{U} = \mathbf{I}$.
2. If $\sigma_n(\mathbf{A}) > \sigma_{n+1}\begin{pmatrix} \mathbf{A} & \sqrt{\alpha}\mathbf{B} \end{pmatrix}$, then the solution of (MTLS) is given by
$$\begin{pmatrix} \Delta\mathbf{A} & \Delta\mathbf{B} \end{pmatrix} = \mathbf{U}_2\Sigma_2(\mathbf{V}_{12}^T, \mathbf{V}_{22}^T)\mathbf{T}^{-1},$$
$$\mathbf{x} = -\frac{1}{\sqrt{\alpha}}\mathbf{V}_{12}\mathbf{V}_{22}^{-1},$$
where $\mathbf{T} = \left(\begin{smallmatrix} \mathbf{I}_n & \mathbf{O} \\ \mathbf{O} & \sqrt{\alpha}\mathbf{I}_k \end{smallmatrix}\right)$

3. Block circulant matrices and the DFT. The aim of this section is to give a short summary of results on DFT defined on block circulant matrices and block vectors that are used in the paper. Subsection 3.1 (but not subsection 3.2) is based on [13].

A block circulant matrix is a matrix of the form

$$\mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1}) \stackrel{\triangle}{=} \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{N-1} \\ \mathbf{A}_{N-1} & \mathbf{A}_0 & \cdots & \mathbf{A}_{N-2} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_0 \end{pmatrix},$$

where each submatrix $\mathbf{A}_j$ is a $k \times l$ matrix. The dimensions k and l will be clear from the context and therefore are not part of the notation.

3.1. The DFT of block circulant matrices. From the definition of block circulant matrices we have the following facts.

LEMMA 3.1. *Let* $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1} \in \mathbb{F}^{k\times l}$ *and* $\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{N-1} \in \mathbb{F}^{l\times m}$. *Then*

1. $\mathcal{C}^T(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1})$ *and* $\mathcal{C}^*(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1})$ *are also a block circulant matrix, where*

$$\mathcal{C}^T(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1}) = \mathcal{C}(\mathbf{A}_0^T, \mathbf{A}_{N-1}^T, \ldots, \mathbf{A}_1^T),$$
$$\mathcal{C}^*(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1}) = \mathcal{C}(\mathbf{A}_0^*, \mathbf{A}_{N-1}^*, \ldots, \mathbf{A}_1^*);$$

2. *The product* $\mathcal{C}(\mathbf{A}_0, \ldots, \mathbf{A}_{N-1})\mathcal{C}(\mathbf{B}_0, \ldots, \mathbf{B}_{N-1})$ *is a block circulant matrix* $\mathcal{C}(\mathbf{C}_0, \ldots, \mathbf{C}_{N-1})$, *where*

$$\mathbf{C}_j = \sum_{i=0}^{N-1} \mathbf{A}_j \mathbf{B}_{[j-i]_N}, \quad 0 \le j \le N-1. \tag{2}$$

We now define the DFT and its inverse, which are the main mathematical tools used in the paper.

DEFINITION 3.1. *Let* $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1})$. *Then the DFT of* $\mathbf{A}$ *denoted by* $\mathbf{F}(\mathbf{A})$ *is the block circulant matrix of the same dimensions given by*

$$\mathbf{F}(\mathbf{A}) = \mathcal{C}(\mathbf{F}_0(\mathbf{A}), \mathbf{F}_1(\mathbf{A}), \ldots, \mathbf{F}_{N-1}(\mathbf{A})),$$

where $\mathbf{F}_j(\mathbf{A}),\ 0 \le j \le N-1$ *is defined by*

$$\mathbf{F}_j(\mathbf{A}) \stackrel{\triangle}{=} \sum_{i=0}^{N-1} \omega^{ij} \mathbf{A}_i, \quad 0 \le j \le N-1,$$

where $\omega = e^{-\frac{2\pi \mathbf{i}}{N}}$. *The matrix* $\mathbf{F}_j(\mathbf{A})$ *is called the jth DFT component.*

DEFINITION 3.2. *Let* $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1})$. *Then the inverse DFT (IDFT), denoted by* $\mathbf{F}^{-1}$, *is the BC matrix*

$$\mathbf{F}^{-1}(\mathbf{A}) = (\mathbf{F}_0^{-1}(\mathbf{A}), \mathbf{F}_1^{-1}(\mathbf{A}), \ldots, \mathbf{F}_{N-1}^{-1}(\mathbf{A})),$$

where the jth block is given by

$$\mathbf{F}_j^{-1}(\mathbf{A}) \stackrel{\triangle}{=} \frac{1}{N} \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{A}_i, \quad 0 \le j \le N-1.$$

In particular, we have

$$\mathbf{F}_0^{-1}(\mathbf{A}) = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{A}_i. \tag{3}$$

$\mathbf{F}^{-1}$ is indeed an inverse of $\mathbf{F}$ in the sense that for every $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \ldots, \mathbf{A}_{N-1})$

$$\mathbf{F}^{-1}(\mathbf{F}(\mathbf{A})) = \mathbf{F}(\mathbf{F}^{-1}(\mathbf{A})) = \mathbf{A}. \tag{4}$$

The proof of this fact relies heavily on the useful identity

$$\sum_{j=0}^{N-1} \omega^{jp} = \begin{cases} 0, & p = 1, 2, \ldots, N-1, \\ N, & p = 0. \end{cases} \tag{5}$$

The following properties of $\mathbf{F}_j$ are generalizations to the block circulant case of well-known properties of the DFT for vectors.

LEMMA 3.2. *Let* $\mathbf{A}, \mathbf{B}$*, and* $\mathbf{C}$ *be block circulant matrices* $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1})$, $\mathbf{B} = \mathcal{C}(\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{N-1})$*, and* $\mathbf{C} = \mathcal{C}(\mathbf{C}_0, \mathbf{C}_1, \ldots, \mathbf{C}_{N-1})$*, where* $\mathbf{A}_j, \mathbf{C}_j \in \mathbb{F}^{k\times l}, \mathbf{B}_j \in \mathbb{F}^{l\times m}$, $0 \le j \le N-1$*. Then for every* $0 \le j \le N-1$ *the following holds:*

1. $(\mathbf{F}_j(\mathbf{A}))^* = \mathbf{F}_j(\mathbf{A}^*)$.
2. $\mathbf{F}_j(\mathbf{A}+\mathbf{C}) = \mathbf{F}_j(\mathbf{A}) + \mathbf{F}_j(\mathbf{C})$.
3. $\mathbf{F}_j(\mathbf{A}\mathbf{B}) = \mathbf{F}_j(\mathbf{A})\mathbf{F}_j(\mathbf{B})$.

An important special case of block circulant matrices are *elementary block circulant* matrices, which are matrices of the form

$$\mathcal{M}(\mathbf{A}_0, \mathbf{A}_1) \stackrel{\triangle}{=} \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_1) = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{A}_0 & \cdots & \mathbf{A}_1 \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_1 & \cdots & \mathbf{A}_0 \end{pmatrix}.$$

In this case there are also only two different DFT components:

$$\begin{aligned} \mathbf{F}_0(\mathcal{M}(\mathbf{A}_0, \mathbf{A}_1)) &= \mathbf{A}_0 + (N-1)\mathbf{A}_1, \\ \mathbf{F}_j(\mathcal{M}(\mathbf{A}_0, \mathbf{A}_1)) &= \mathbf{A}_0 - \mathbf{A}_1, \ \ 1 \le j \le N-1. \end{aligned}$$

It is also easy to see that there are only two different inverse DFT components:

$$\begin{aligned} \mathbf{F}_0^{-1}(\mathcal{M}(\mathbf{A}_0, \mathbf{A}_1)) &= \frac{1}{N}(\mathbf{A}_0 + (N-1)\mathbf{A}_1), \\ \mathbf{F}_j^{-1}(\mathcal{M}(\mathbf{A}_0, \mathbf{A}_1)) &= \frac{1}{N}(\mathbf{A}_0 - \mathbf{A}_1), \ \ 1 \le j \le N-1. \end{aligned}$$

3.2. The DFT of block vectors.

DEFINITION 3.3. *Let* $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{N-1} \in \mathbb{F}^p$*; then the* DFT of the block vector $\mathbf{y} = (\mathbf{y}_0^T, \mathbf{y}_1^T, \ldots, \mathbf{y}_{N-1}^T)^T \in \mathbb{F}^{Np}$ *is the vector* $\mathbf{f}(\mathbf{y}) = (\mathbf{f}_0(\mathbf{y})^T, \mathbf{f}_1(\mathbf{y})^T, \ldots, \mathbf{f}_{N-1}(\mathbf{y})^T)^T$ *whose jth block (subvector) is given by*

$$\mathbf{f}_j(\mathbf{y}) \stackrel{\triangle}{=} \sum_{i=0}^{N-1} \omega^{-ij}\mathbf{y}_i.$$

Remark. Notice that the definition of the DFT for block vectors is slightly different from the definition of the DFT for block circulant matrices (ω^{-ij} instead of ω^{ij}). Although the difference seems negligible, it is crucial to define the DFT for block vectors in that manner; otherwise, some critical properties will be lost (cf. Lemma 3.4).

The inverse DFT of $\mathbf{f}$ is defined by

$$\mathbf{f}_j^{-1}(\mathbf{y}) = \frac{1}{N}\sum_{i=0}^{N-1}\omega^{ij}\mathbf{y}_i.$$

For every $0 \le j \le N-1$,

$$\mathbf{f}_j^{-1}(\mathbf{f}(\mathbf{y})) = \mathbf{f}_j(\mathbf{f}^{-1}(\mathbf{y})) = \mathbf{y}_j.$$

The following properties of the DFT and the inverse DFT will be useful later on.

LEMMA 3.3. *Let* $\mathbf{y} = (\mathbf{y}_0^T, \mathbf{y}_1^T, \ldots, \mathbf{y}_{N-1}^T)^T$ *and* $\mathbf{z} = (\mathbf{z}_0^T, \mathbf{z}_1^T, \ldots, \mathbf{z}_{N-1}^T)^T$*, where* $\mathbf{y}_0, \ldots, \mathbf{y}_{N-1} \in \mathbb{F}^m$ *and* $\mathbf{z}_0, \ldots, \mathbf{z}_{N-1} \in \mathbb{F}^m$*. Then we have the following:*

1. (Norm preservation): $\|\mathbf{f}(\mathbf{y})\|^2 = N\|\mathbf{y}\|^2$.

2. (Linearity): *For every two scalars* $\alpha, \beta \in \mathbb{F}, \mathbf{f}(\alpha\mathbf{v} + \beta\mathbf{z}) = \alpha\mathbf{f}(\mathbf{v}) + \beta\mathbf{f}(\mathbf{z})$.

Proof. 1.

$$\begin{aligned}
\|\mathbf{f}(\mathbf{y})\|^2 &= \sum_{j=0}^{N-1}\|\mathbf{f}_j(\mathbf{y})\|^2 = \sum_{j=0}^{N-1}\left\|\sum_{i=0}^{N-1}\omega^{-ij}\mathbf{y}_i\right\|^2 = \sum_{j=0}^{N-1}\left(\sum_{k=0}^{N-1}\omega^{kj}\mathbf{y}_k^T\right)\left(\sum_{i=0}^{N-1}\omega^{-ij}\mathbf{y}_i\right)\\
&= \sum_{j=0}^{N-1}\sum_{i=0}^{N-1}\sum_{k=0}^{N-1}\omega^{j(k-i)}\mathbf{y}_k^T\mathbf{y}_i\\
&= \sum_{i=0}^{N-1}\sum_{k=0}^{N-1}\left(\sum_{j=0}^{N-1}\omega^{j(i-k)}\right)\mathbf{y}_k^T\mathbf{y}_i \overset{(5)}{=} \sum_{i=0}^{N-1}N\|\mathbf{y}_i\|^2 = N\|\mathbf{y}\|^2.
\end{aligned}$$

2. It directly follows from the definition of the DFT for block vectors. ∎

Lemma 3.4 shows a connection between the DFT of block circulant matrices and the DFT of block vectors; this connection is one of the key ingredients in the analysis of the total least squares for block circulant systems.

LEMMA 3.4. *Let* $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1})$ *and* $\mathbf{y} = (\mathbf{y}_0^T, \mathbf{y}_1^T, \ldots, \mathbf{y}_{N-1}^T)^T$*, where* $\mathbf{y}_0, \ldots, \mathbf{y}_{N-1} \in \mathbb{F}^m$ *and* $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1} \in \mathbb{F}^{m\times n}$*. Then* $\mathbf{f}_j(\mathbf{A}\mathbf{y}) = \mathbf{F}_j(\mathbf{A})\mathbf{f}_j(\mathbf{y})$ *for every* $0 \le j \le N-1$.

Proof. For every $0 \le j \le N-1$,

$$\begin{aligned}
\mathbf{f}_j(\mathbf{A}\mathbf{y}) &= \sum_{i=0}^{N-1}\omega^{-ij}(\mathbf{A}\mathbf{y})_i = \sum_{i=0}^{N-1}\omega^{-ij}\left(\sum_{k=0}^{N-1}\mathbf{A}_{[k-i]_N}\mathbf{y}_k\right) = \sum_{i=0}^{N-1}\sum_{k=0}^{N-1}\omega^{-ij}\mathbf{A}_{[k-i]_N}\mathbf{y}_k\\
&= \sum_{k=0}^{N-1}\sum_{i=0}^{N-1}\omega^{-ij}\mathbf{A}_{[k-i]_N}\mathbf{y}_k\\
&= \sum_{k=0}^{N-1}\omega^{-kj}\left(\sum_{i=0}^{N-1}\omega^{(k-i)j}\mathbf{A}_{[k-i]_N}\right)\mathbf{y}_k = \sum_{k=0}^{N-1}\omega^{-kj}\mathbf{F}_j(\mathbf{A})\mathbf{y}_k\\
&= \mathbf{F}_j(\mathbf{A})\sum_{k=0}^{N-1}\omega^{-kj}\mathbf{y}_k = \mathbf{F}_j(\mathbf{A})\mathbf{f}_j(\mathbf{y}). \qquad \blacksquare
\end{aligned}$$

4. STLS in the case of block circulant matrices.

4.1. The algorithm. Suppose that $\mathbf{A}$ has a block circulant structure, i.e., $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1})$, where $\mathbf{A}_i \in \mathbb{F}^{m\times n}$, and we wish to find a perturbation matrix $\Delta\mathbf{A}$, which also has a block circulant structure. We assume that the system is overdetermined, i.e., $m > n$. The STLS problem for block circulant matrices can be written as

$$\begin{array}{lll} & \min_{\Delta\mathbf{A}_0,\ldots,\Delta\mathbf{A}_{N-1},\Delta\mathbf{b},\mathbf{x}} & \|\Delta\mathbf{A}\|^2 + \|\Delta\mathbf{b}\|^2 \\ (6) & \text{subject to} & (\mathbf{A} - \Delta\mathbf{A})\mathbf{x} = \mathbf{b} - \Delta\mathbf{b}, \\ & & \Delta\mathbf{A} = \mathcal{C}(\Delta\mathbf{A}_0, \Delta\mathbf{A}_1, \ldots, \Delta\mathbf{A}_{N-1}). \end{array}$$

In order to solve this problem, we will first apply the DFT on both sides of the consistency equation

$$(\mathbf{A} - \Delta\mathbf{A})\mathbf{x} = \mathbf{b} - \Delta\mathbf{b}. \tag{7}$$

It will be useful to treat $\mathbf{b} \in \mathbb{F}^{Nm}$ and $\Delta\mathbf{b} \in \mathbb{F}^{Nm}$ as block vectors, i.e., $(\mathbf{b}_0^T, \mathbf{b}_1^T, \ldots, \mathbf{b}_{N-1}^T)^T$ and $\Delta\mathbf{b} = (\Delta\mathbf{b}_0^T, \ldots, \Delta\mathbf{b}_{N-1}^T)^T$, where $\mathbf{b}_j, \Delta\mathbf{b}_j \in \mathbb{F}^m, 0 \le j \le N-1$. By applying the block vector DFT, $\mathbf{f}$, on both sides of (7) we obtain

$$\mathbf{f}((\mathbf{A} - \Delta\mathbf{A})\mathbf{x}) = \mathbf{f}(\mathbf{b} - \Delta\mathbf{b}).$$

Using property 2 of Lemma 3.3 we have

$$\mathbf{f}((\mathbf{A} - \Delta\mathbf{A})\mathbf{x}) = \mathbf{f}(\mathbf{b}) - \mathbf{f}(\Delta\mathbf{b}),$$

which is equivalent to the following system of N equations:

$$\mathbf{f}_j((\mathbf{A} - \Delta\mathbf{A})\mathbf{x}) = \mathbf{f}_j(\mathbf{b}) - \mathbf{f}_j(\Delta\mathbf{b}) \quad \forall\ 0 \le j \le N-1.$$

Finally, using Lemma 3.4 we have that (7) is equivalent to the following N "small" linear systems (in the unknown variables being $\mathbf{f}_0(\mathbf{x}), \ldots, \mathbf{f}_{N-1}(\mathbf{x})$):

$$(\mathbf{F}_j(\mathbf{A}) - \mathbf{F}_j(\Delta\mathbf{A}))\mathbf{f}_j(\mathbf{x}) = \mathbf{f}_j(\mathbf{b}) - \mathbf{f}_j(\Delta\mathbf{b}) \quad \forall\ 0 \le j \le N-1.$$

The objective function $\|\Delta\mathbf{A}\|^2 + \|\Delta\mathbf{b}\|^2$ can also be expressed solely by its DFT components. Indeed, from Lemma 3.3 we have

$$\|\Delta\mathbf{b}\|^2 = \frac{1}{N}\|\mathbf{f}(\Delta\mathbf{b})\|^2 = \frac{1}{N}\left(\sum_{j=0}^{N-1}\|\mathbf{f}_j(\Delta\mathbf{b})\|^2\right). \tag{8}$$

Also, by the definition of Frobenius norm we have that $\|\Delta\mathbf{A}\|^2 = \mathrm{Tr}(\Delta\mathbf{A}^*\Delta\mathbf{A})$. Since $\Delta\mathbf{A}$ is block circulant, then by Lemma 3.1 we have that $\Delta\mathbf{A}^*\Delta\mathbf{A}$ is also block circulant, and thus we can write

$$\Delta\mathbf{A}^*\Delta\mathbf{A} = \mathcal{C}(\mathbf{R}_0, \mathbf{R}_1, \ldots, \mathbf{R}_{N-1}) = \begin{pmatrix} \mathbf{R}_0 & \mathbf{R}_1 & \cdots & \mathbf{R}_{N-1} \\ \mathbf{R}_{N-1} & \mathbf{R}_0 & \cdots & \mathbf{R}_{N-2} \\ \vdots & \vdots & & \vdots \\ \mathbf{R}_1 & \mathbf{R}_2 & \cdots & \mathbf{R}_0 \end{pmatrix}$$

for some $\mathbf{R}_0, \mathbf{R}_1, \ldots \mathbf{R}_{N-1} \in \mathbb{F}^{n\times n}$, and therefore

$$\|\Delta\mathbf{A}\|^2 = \mathrm{Tr}(\Delta\mathbf{A}^*\Delta\mathbf{A}) = N\mathrm{Tr}(\mathbf{R}_0).$$

Moreover,

$$\begin{aligned}\|\Delta\mathbf{A}\|^2 &= N\mathrm{Tr}(\mathbf{R}_0) \overset{(4)}{=} N\mathrm{Tr}(\mathbf{F}_0^{-1}(\mathbf{F}(\Delta\mathbf{A}^*\Delta\mathbf{A})))\\ &= \sum_{j=0}^{N-1}\mathrm{Tr}(\mathbf{F}_j(\Delta\mathbf{A}^*\Delta\mathbf{A})) \quad \text{(by (3))}\\ &= \sum_{j=0}^{N-1}\mathrm{Tr}(\mathbf{F}_j(\Delta\mathbf{A})^*\mathbf{F}_j(\Delta\mathbf{A})) = \sum_{j=0}^{N-1}\|\mathbf{F}_j(\Delta\mathbf{A})\|^2 \text{(by Lemma 3.1, property 2).} \end{aligned} \tag{9}$$

We thus obtained that (6) is reduced to

$$\begin{array}{ll} \min_{\Delta\mathbf{A}_0,\ldots,\Delta\mathbf{A}_{N-1},\Delta\mathbf{b},\mathbf{x}} & \sum_{j=0}^{N-1}\left(\|\mathbf{F}_j(\Delta\mathbf{A})\|^2 + \frac{1}{N}\|\mathbf{f}_j(\Delta\mathbf{b})\|^2\right)\\ \text{subject to} & (\mathbf{F}_j(\mathbf{A}) - \mathbf{F}_j(\Delta\mathbf{A}))\mathbf{f}_j(\mathbf{x}) = \mathbf{f}_j(\mathbf{b}) - \mathbf{f}_j(\Delta\mathbf{b}), \quad 0 \le j \le N-1,\\ & \Delta\mathbf{A} = \mathcal{C}(\Delta\mathbf{A}_0, \Delta\mathbf{A}_1, \ldots, \Delta\mathbf{A}_{N-1}). \end{array} \tag{10}$$

Making the change of variables

$$\mathbf{G}_j = \mathbf{F}_j(\Delta\mathbf{A}) = \sum_{i=0}^{N-1}\omega^{ij}\Delta\mathbf{A}_i, \quad 0 \le j \le N-1,$$

$$\mathbf{c}_j = \mathbf{f}_j(\Delta\mathbf{b}) = \sum_{i=0}^{N-1}\omega^{-ij}\Delta\mathbf{b}_i, \quad 0 \le j \le N-1,$$

$$\mathbf{z}_j = \mathbf{f}_j(\mathbf{x}) = \sum_{i=0}^{N-1}\omega^{-ij}\mathbf{x}_i, \quad 0 \le j \le N-1,$$

we obtain the following equivalent minimization problem:

$$\begin{array}{ll} \min_{\mathbf{G}_0,\ldots,\mathbf{G}_{N-1},\mathbf{c}_0,\ldots,\mathbf{c}_{N-1},\mathbf{z}_0,\ldots,\mathbf{z}_{N-1}} & \sum_{j=0}^{N-1}\left(\|\mathbf{G}_j\|^2 + \frac{1}{N}\|\mathbf{c}_j\|^2\right)\\ \text{subject to} & (\mathbf{F}_j(\mathbf{A}) - \mathbf{G}_j)\mathbf{z}_j = \mathbf{f}_j(\mathbf{b}) - \mathbf{c}_j, \quad 0 \le j \le N-1. \end{array} \tag{11}$$

Since (10) is separable in the variables

$$(\mathbf{G}_0, \mathbf{c}_0, \mathbf{z}_0), (\mathbf{G}_1, \mathbf{c}_1, \mathbf{z}_1), \ldots, (\mathbf{G}_{N-1}, \mathbf{c}_{N-1}, \mathbf{z}_{N-1})$$

we actually need to solve N small TLS problems and then use the inverse DFT in order to find the values of $(\Delta\mathbf{A}, \Delta\mathbf{b}, \mathbf{x})$.

We summarize the above by presenting the block circulant TLS (BCTLS) algorithm for solving the STLS problem (6). The algorithm is essentially as simple as the classical SVD-based algorithm since its main effort consists of solving N small unstructured TLS problems.

Algorithm BCTLS.

Input: $\mathbf{A}, \mathbf{b}$, where $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1}) \in \mathbb{F}^{Nm\times Nn}$ is a BC matrix and $\mathbf{b} = (\mathbf{b}_0^T, \mathbf{b}_1^T, \ldots, \mathbf{b}_{N-1}^T)^T$ such that $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1} \in \mathbb{F}^{m\times n}$ and $\mathbf{b}_0, \ldots, \mathbf{b}_{N-1} \in \mathbb{F}^m$.
Output: $\Delta\mathbf{A}, \Delta\mathbf{b}, \mathbf{x}$, where $\Delta\mathbf{A} \in \mathbb{F}^{Nm\times Nn}$ is a block circulant matrix, $\Delta\mathbf{b} \in \mathbb{F}^{Nm}$, and $\mathbf{x} \in \mathbb{F}^{Nn}$ is the STLS solution.

1. Calculate the N DFT components of $\mathbf{A}$,

$$\mathbf{F}_j(\mathbf{A}) = \sum_{i=0}^{N-1}\omega^{ij}\mathbf{A}_i, \qquad 0 \le j \le N-1, \tag{12}$$

and the N DFT components of $\mathbf{b}$,

$$\mathbf{f}_j(\mathbf{b}) = \sum_{i=0}^{N-1} \omega^{-ij}\mathbf{b}_i, \quad 0 \le j \le N-1.$$

2. For every $0 \le j \le N-1$, call the TLS algorithm with input $\left(\mathbf{F}_j(\mathbf{A}), \mathbf{f}_j(\mathbf{b}), \frac{1}{N}\right)$ and obtain an output $(\mathbf{G}_j, \mathbf{c}_j, \mathbf{z}_j)$.
3. Denote $\mathbf{G} = \mathcal{C}(\mathbf{G}_0, \mathbf{G}_1, \ldots, \mathbf{G}_{N-1}), \mathbf{c} = (\mathbf{c}_0^T, \mathbf{c}_1^T, \ldots, \mathbf{c}_{N-1}^T)^T$ and $\mathbf{z} = (\mathbf{z}_0^T, \mathbf{z}_1^T, \ldots, \mathbf{z}_{N-1}^T)^T$. The output of the BCTLS algorithm is computed by applying the inverse DFT of $\mathbf{G}, \mathbf{c}$, and $\mathbf{z}$. The obtained solution of the STLS problem (6) is then $\Delta\mathbf{A} = \mathcal{C}(\Delta\mathbf{A}_0, \Delta\mathbf{A}_1, \ldots, \Delta\mathbf{A}_{N-1})$, $\Delta\mathbf{b} = (\Delta\mathbf{b}_0^T, \Delta\mathbf{b}_1^T, \ldots, \Delta\mathbf{b}_{N-1}^T)^T$, and $\mathbf{x} = (\mathbf{x}_0^T, \mathbf{x}_1^T, \ldots, \mathbf{x}_{N-1}^T)$, where

$$\Delta\mathbf{A}_j = \mathbf{F}_j^{-1}(\mathbf{G}) = \frac{1}{N}\sum_{i=0}^{N-1} \omega^{-ij}\mathbf{G}_i, \quad 0 \le j \le N-1, \tag{13}$$

$$\Delta\mathbf{b}_j = \mathbf{f}_j^{-1}(\mathbf{c}) = \frac{1}{N}\sum_{i=0}^{N-1} \omega^{ij}\mathbf{c}_i, \quad 0 \le j \le N-1,$$

$$\mathbf{x}_j = \mathbf{f}_j^{-1}(\mathbf{z}) = \frac{1}{N}\sum_{i=0}^{N-1} \omega^{ij}\mathbf{z}_i, \quad 0 \le j \le N-1. \tag{14}$$

Remark. Step 2 of Algorithm BCTLS requires N executions of algorithm TLS, and hence the following condition must be satisfied (see step 2 of algorithm TLS):

$$\sigma_n(\mathbf{F}_j(\mathbf{A})) > \sigma_{n+1}\left(\mathbf{F}_j(\mathbf{A}), \frac{1}{\sqrt{N}}\mathbf{f}_j(\mathbf{b})\right) \quad \forall 0 \le j \le N-1. \tag{15}$$

We claim that condition (15) implies that the matrix $\mathbf{A}$ is full column rank, which is the same as $\mathbf{A}^*\mathbf{A}$ being nonsingular. Indeed, condition (15) implies in particular that $\sigma_n(\mathbf{F}_j(\mathbf{A})) > 0$ for every $0 \le j \le N-1$, which is the same as saying that $\mathbf{F}_j(\mathbf{A})^*\mathbf{F}_j(\mathbf{A})$ is nonsingular for every $0 \le j \le N-1$. By Lemma 3.2 we obtain that $\mathbf{F}_j(\mathbf{A}^*\mathbf{A})$ is nonsingular for every $0 \le j \le N-1$. By [13, Theorem 3.1], the eigenvalues of $\mathbf{A}^*\mathbf{A}$ are the Nn eigenvalues of the N matrices $\mathbf{F}_j(\mathbf{A}^*\mathbf{A})$. The latter matrices are nonsingular, and hence have only nonzero eigenvalues. Therefore, all the eigenvalues of $\mathbf{A}^*\mathbf{A}$ are different from zero, and, as a result, $\mathbf{A}^*\mathbf{A}$ is nonsingular. ∎

From (14) it seems as if $\mathbf{x}$ is a complex vector even in the real case $\mathbb{F} = \mathbb{R}$. However, we claim that in fact it is a real vector. This is proved in the following theorem.

THEOREM 4.1. *Let* $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1})$ *and* $\mathbf{b} = (\mathbf{b}_0^T, \mathbf{b}_1^T, \ldots, \mathbf{b}_{N-1}^T)^T$, *where* $\mathbf{A}_0, \ldots, \mathbf{A}_{N-1} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{b}_0, \ldots, \mathbf{b}_{N-1} \in \mathbb{R}^m$. *Then Algorithm BCTLS with input* $\mathbf{A}$ *and* $\mathbf{b}$ *produces a real solution:* $\Delta\mathbf{A} \in \mathbb{R}^{Nm\times Nn}, \Delta\mathbf{b} \in \mathbb{R}^{Nm}$, *and* $\mathbf{x} \in \mathbb{R}^{Nn}$.

Proof. Throughout the proof we use the same notation used in the description of the BCTLS algorithm. The componentwise complex conjugate of a vector or a matrix is denoted by $\overline{(\cdot)}$. The proof is based on the following four claims:

(i) $\mathbf{F}_{N-j}(\mathbf{A}) = \overline{\mathbf{F}_j(\mathbf{A})}$ and $\mathbf{f}_{N-j}(\mathbf{b}) = \overline{\mathbf{f}_j(\mathbf{b})}$ for every $0 \le j \le N-1$.
(ii) $\mathbf{G}_{N-j} = \overline{\mathbf{G}_j}$ and $\mathbf{c}_{N-j} = \overline{\mathbf{c}_j}$ for every $0 \le j \le N-1$.
(iii) $\Delta\mathbf{A}$ is a real-valued matrix, and $\Delta\mathbf{b}$ is a real-valued vector.
(iv) $\mathbf{x}$ is a real valued vector.

Proof of (i). For every $0 \le j \le N-1$,

$$\mathbf{F}_{N-j}(\mathbf{A}) = \sum_{i=0}^{N-1} \omega^{i(N-j)} \mathbf{A}_i = \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{A}_i = \sum_{i=0}^{N-1} \overline{\omega^{ij}} \mathbf{A}_i = \overline{\sum_{i=0}^{N-1} \omega^{ij} \mathbf{A}_i} = \overline{\mathbf{F}_j(\mathbf{A})}.$$

The proof that $\mathbf{f}_{N-j}(\mathbf{b}) = \overline{\mathbf{f}_j(\mathbf{b})}$ is almost identical.

Proof of (ii). For every $0 \le j \le N-1$, let the SVD of $\left(\mathbf{F}_j(\mathbf{A}), \frac{1}{N}\mathbf{f}_j(\mathbf{b})\right)$ be given by

$$\left(\mathbf{F}_j(\mathbf{A}), \frac{1}{N}\mathbf{f}_j(\mathbf{b})\right) = \mathbf{U}^j \Sigma^j (\mathbf{V}^j)^*. \tag{16}$$

Taking the complex conjugate of both sides of (16) and using fact (i), we obtain that the SVD of $\left(\mathbf{F}_{N-j}(\mathbf{A}), \frac{1}{N}\mathbf{f}_{N-j}(\mathbf{b})\right)$ is given by

$$\left(\mathbf{F}_{N-j}(\mathbf{A}), \frac{1}{N}\mathbf{f}_{N-j}(\mathbf{b})\right) = \overline{\mathbf{U}}^j \Sigma^j (\overline{\mathbf{V}^j})^*. \tag{17}$$

Since $(\mathbf{G}_j, \mathbf{c}_j, \mathbf{z}_j)$ is the output of the TLS algorithm with input $\left(\mathbf{F}_j(\mathbf{A}), \mathbf{f}_j(\mathbf{b}), \frac{1}{N}\right)$, we have that

$$(\mathbf{G}_j, \mathbf{c}_j) = \sigma_{n+1} \mathbf{u}_{n+1}^j (\mathbf{v}_{n+1}^j)^T \mathbf{T}, \quad (\mathbf{G}_{N-j}, \mathbf{c}_{N-j}) = \sigma_{n+1} \mathbf{u}_{n+1}^{N-j} (\mathbf{v}_{n+1}^{N-j})^T \mathbf{T}, \tag{18}$$

where

$$\mathbf{T} = (\operatorname{diag} \underbrace{1, 1, \ldots, 1}_{n \text{ times}}, \sqrt{1/N}).$$

$\mathbf{u}_{n+1}^j$ and $\mathbf{v}_{n+1}^j$ are the $(n+1)$th columns of the matrices $\mathbf{U}^j$ and $\mathbf{V}^j$ respectively. $\mathbf{u}_{n+1}^{N-j}$ and $\mathbf{v}_{n+1}^{N-j}$ are the $(n+1)$th columns of the matrices $\mathbf{U}^{N-j}$ and $\mathbf{V}^{N-j}$, respectively. From (16) and (17) it follows that

$$\mathbf{u}_{n+1}^{N-j} = \overline{\mathbf{u}_{n+1}^j}, \quad \mathbf{v}_{n+1}^{N-j} = \overline{\mathbf{v}_{n+1}^j}. \tag{19}$$

Substituting (19) into (18) we deduce that $\mathbf{G}_{N-j} = \overline{\mathbf{G}_j}$ and $\mathbf{c}_{N-j} = \overline{\mathbf{c}_j}$.

Proof of (iii). First notice that $(\mathbf{G}_0, \mathbf{c}_0, \mathbf{z}_0)$ is the output of the TLS algorithm with *real* input $(\mathbf{F}_0(\mathbf{A}_0), \mathbf{f}_0(\mathbf{b}), \frac{1}{N})$, and therefore $\mathbf{G}_0$ and $\mathbf{c}_0$ are real-valued. If N is odd, then $\Delta\mathbf{A}_j$ can be written as

$$\begin{aligned}
\Delta\mathbf{A}_j &= \frac{1}{N} \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{G}_i \\
&= \frac{1}{N}\mathbf{G}_0 + \frac{1}{N} \sum_{i=1}^{\frac{N-1}{2}} (\omega^{-ij}\mathbf{G}_i + \omega^{-(N-i)j}\mathbf{G}_{N-i}) \\
&\overset{\text{fact (ii)}}{=} \frac{1}{N}\mathbf{G}_0 + \frac{1}{N} \sum_{i=1}^{\frac{N-1}{2}} (\omega^{-ij}\mathbf{G}_i + \overline{\omega^{-ij}\mathbf{G}_i}),
\end{aligned}$$

which is a real-valued matrix. The proof for the case where N is even and for the vector $\Delta\mathbf{b}$ are almost identical.

Proof of (iv). Since $\Delta\mathbf{A}$ is a real-valued matrix and $\Delta\mathbf{b}$ is a real-valued matrix and $\mathbf{x}$ is the unique solution of the real system $(\mathbf{A}-\Delta\mathbf{A})\mathbf{x}=\mathbf{b}-\Delta\mathbf{b}$, we conclude that $\mathbf{x}$ must be real-valued. □

Computational complexity of algorithm BCTLS. There are two kinds of calculations performed by Algorithm BCTLS:

1. *DFT and IDFT calculations.* Steps 1 and 3 require the calculation of the DFT of a block circulant matrix and a block vector (step 1) and the IDFT of a block circulant matrix and a block vector (step 3). The calculation of the DFT or the IDFT of a block circulant matrix given in (12) and (13) both require $O(N^2mn)$ operations. However, this can be reduced to $O(N\log(N)mn)$ if one uses fast Fourier transform (FFT) [2]. Indeed, the computation of the sequence $\mathbf{F}_0(\mathbf{A}),\ldots,\mathbf{F}_{N-1}(\mathbf{A})$ involves mn DFT calculations of sequences of scalars, one for each component of $\mathbf{A}_0,\ldots,\mathbf{A}_{N-1}$. Each of the mn DFT calculations can be done in $O(N\log(N))$ operations by the classical FFT. Similarly, the computational effort required in order to calculate the DFT or the IDFT of a block vector is $O(N\log(N)m)$. Therefore, the overall computational effort of the DFT computations is $O(N\log(N)mn)$.
2. *SVD calculations.* The algorithm requires N SVD calculations in step 2. The SVD of an $m\times n$ matrix can be calculated in $O(m^2n)$ operations (see, e.g., [6]). Thus, step 2 requires $O(Nm^2n)$ operations.

As a consequence, the overall computational complexity of the algorithm is $O(Nm^2n+N\log(N)mn)$ operations. Note that the computational effort of the unstructured TLS problem with a matrix of the same size $Nm\times Nn$ is $O(N^3m^2n)$ operations, which, for large N, is significantly higher.

4.2. Illustration of Algorithm BCTLS. In this section we illustrate the BCTLS algorithm through an example with $N=3, m=3$, and $n=2$. Suppose we have a "correct" linear system

$$\mathbf{A}^{(c)}\mathbf{x}^{(c)}=\mathbf{b}^{(c)},$$

where the "correct" solution is $\mathbf{x}^{(c)}=(1,1,1,1,1,1)^T$, $A^{(c)}$ is the 9×6 matrix defined by $\mathbf{A}^{(c)}=\mathcal{C}(\mathbf{A}_0^{(c)},\mathbf{A}_1^{(c)},\mathbf{A}_2^{(c)})$, where

$$\mathbf{A}_0^{(c)}=\begin{pmatrix}1&1\\1&1\\1&0\end{pmatrix},\quad \mathbf{A}_1^{(c)}=\begin{pmatrix}1&1\\0&0\\1&0\end{pmatrix},\quad \mathbf{A}_2^{(c)}=\begin{pmatrix}1&1\\1&0\\0&1\end{pmatrix},$$

and $\mathbf{b}^{(c)}$ is the "correct" observed vector, i.e., $\mathbf{b}^{(c)}=\mathbf{A}^{(c)}\mathbf{x}=(6,3,3,6,3,3,6,3,3)^T$. Assume now that each component of $\mathbf{A}_0^{(c)},\mathbf{A}_1^{(c)},\mathbf{A}_2^{(c)}$, and $\mathbf{b}^{(c)}$ is corrupted by a Gaussian additive noise with zero expectation and standard variation 0.3. In one realization of the noise, this gave rise to the "observed matrix" $\mathbf{A}=\mathcal{C}(\mathbf{A}_0,\mathbf{A}_1,\mathbf{A}_2)$,

$$\mathbf{A}_0=\begin{pmatrix}1.529&0.584\\0.989&0.839\\1.094&-0.091\end{pmatrix},\quad \mathbf{A}_1=\begin{pmatrix}1.038&0.935\\0.177&-0.140\\0.681&-0.148\end{pmatrix},\quad \mathbf{A}_2=\begin{pmatrix}1.074&0.132\\1.287&0.224\\0.092&1.195\end{pmatrix},$$

and the "observed vector,"

$$\mathbf{b}=(5.934,2.925,2.941,5.656,2.989,3.043,6.434,3.114,3.163)^T.$$

Ignoring the special structure of $\mathbf{A}$ and applying the (unstructured) TLS algorithm, the TLS solution is

$$\mathbf{x}_{TLS} = (0.6832, 1.0906, 0.8109, 1.3365, 0.9744, 1.1405)^T.$$

To solve the STLS problem we need to compute the three DFT components of $\mathbf{A}$ and $\mathbf{b}$:

$$\mathbf{F}_0(\mathbf{A}) = \mathbf{A}_0 + \mathbf{A}_1 + \mathbf{A}_2 = \begin{pmatrix} 3.6424 & 2.6517 \\ 2.4546 & 0.9236 \\ 1.8682 & 0.9557 \end{pmatrix},$$

$$\mathbf{F}_1(\mathbf{A}) = \mathbf{A}_0 + \omega\mathbf{A}_1 + \omega^2\mathbf{A}_2 = \begin{pmatrix} 0.4728 - 0.0307\mathbf{i} & -0.4499 - 0.1702\mathbf{i} \\ 0.2566 - 0.9609\mathbf{i} & 0.7974 - 0.3166\mathbf{i} \\ 0.7077 + 0.5107\mathbf{i} & -0.6149 - 1.1646\mathbf{i} \end{pmatrix},$$

$$\mathbf{F}_2(\mathbf{A}) = \mathbf{A}_0 + \omega^2\mathbf{A}_1 + \omega\mathbf{A}_2 = \begin{pmatrix} 0.4728 + 0.0307\mathbf{i} & -0.4499 + 0.1702\mathbf{i} \\ 0.2566 + 0.9609\mathbf{i} & 0.7974 + 0.3166\mathbf{i} \\ 0.7077 - 0.5107\mathbf{i} & -0.6149 + 1.1646\mathbf{i} \end{pmatrix},$$

$$\begin{aligned}
\mathbf{f}_0(\mathbf{b}) &= \mathbf{b}_0 + \mathbf{b}_1 + \mathbf{b}_2 = (18.0260, 9.0289, 9.1477)^T, \\
\mathbf{f}_1(\mathbf{b}) &= \mathbf{b}_0 + \omega^2\mathbf{b}_1 + \omega\mathbf{b}_2 \\
&= (-0.1106 + 0.6740\mathbf{i}, -0.1266 + 0.1085\mathbf{i}, -0.1621 + 0.1034\mathbf{i})^T, \\
\mathbf{f}_2(\mathbf{b}) &= \mathbf{b}_0 + \omega\mathbf{b}_1 + \omega^2\mathbf{b}_2 \\
&= (-0.1106 - 0.6740\mathbf{i}, -0.1266 - 0.1085\mathbf{i}, -0.1621 - 0.1034\mathbf{i})^T.
\end{aligned}$$

Applying the TLS algorithm on the three sets of inputs $(\mathbf{F}_j(\mathbf{A}), \mathbf{f}_j(\mathbf{b}), 1/3)$ we derive the three DFT components of $\mathbf{x}$:

$$\mathbf{f}_0(\mathbf{x}) = \begin{pmatrix} 2.5428 \\ 3.4395 \end{pmatrix}, \quad \mathbf{f}_1(\mathbf{x}) = \begin{pmatrix} -0.2096 + 0.1416\mathbf{i} \\ -0.1480 - 0.1697\mathbf{i} \end{pmatrix}, \quad \mathbf{f}_2(\mathbf{x}) = \begin{pmatrix} -0.2096 - 0.1416\mathbf{i} \\ -0.1480 + 0.1697\mathbf{i} \end{pmatrix},$$

and the solution of the STLS is given by

$$\mathbf{x}_{STLS} = \mathbf{x} = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{x})) = \frac{1}{3}\begin{pmatrix} \mathbf{f}_0(\mathbf{x}) + \mathbf{f}_1(\mathbf{x}) + \mathbf{f}_2(\mathbf{x}) \\ \mathbf{f}_0(\mathbf{x}) + \omega\mathbf{f}_1(\mathbf{x}) + \omega^2\mathbf{f}_2(\mathbf{x}) \\ \mathbf{f}_0(\mathbf{x}) + \omega^2\mathbf{f}_1(\mathbf{x}) + \omega\mathbf{f}_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 0.7079 \\ 1.0478 \\ 0.8357 \\ 1.2938 \\ 0.9993 \\ 1.0978 \end{pmatrix}.$$

It is interesting to note that in this particular case it so happens that

$$|(\mathbf{x}_{STLS})_i - \mathbf{x}_i^{(c)}| < |(\mathbf{x}_{TLS})_i - \mathbf{x}_i^{(c)}| \quad \forall i. \tag{20}$$

Notice also that, as claimed in Theorem 4.1, $\mathbf{x}_{STLS}$ is a real vector.

5. STLS in the case of elementary block circulant systems. In this section we assume that $\mathbf{A}$ is an elementary block circulant matrix, i.e., $\mathbf{A} = \mathcal{M}(\mathbf{A}_0, \mathbf{A}_1)$, $\mathbf{A}_0$, $\mathbf{A}_1 \in \mathcal{F}^{m\times n}(m > n)$ (see section 3), and we wish to find a perturbation matrix $\Delta\mathbf{A}$, which also has an elementary block circulant structure. In this case, the STLS problem becomes

$$\begin{array}{lll}
& \min_{\Delta\mathbf{A}_0, \Delta, \mathbf{A}_1, \Delta\mathbf{b}, \mathbf{x}} & \|\Delta\mathbf{A}\|^2 + \|\Delta\mathbf{b}\|^2 \\
(21) & \text{subject to} & (\mathbf{A} - \Delta\mathbf{A})\mathbf{x} = \mathbf{b} - \Delta\mathbf{b}, \\
& & \Delta\mathbf{A} = \mathcal{M}(\Delta\mathbf{A}_0, \Delta\mathbf{A}_1).
\end{array}$$

Remark. Although an EBC matrix is a special case of a BC matrix, we cannot apply the BCTLS algorithm to solve (21) since an EBC matrix possesses additional special structure, $\mathbf{A}_j = \mathbf{A}_k \forall j \neq k (j, k \neq 0)$, which is not guaranteed to be produced by the BCTLS algorithm.

As in the case of the block circulant structure, we will apply the DFT on both sides of the consistency equation $(\mathbf{A}-\Delta\mathbf{A})\mathbf{x} = \mathbf{b}-\Delta\mathbf{b}$ and obtain that the consistency equation is equivalent to N "small" linear systems: the linear system

$$(\mathbf{F}_0(\mathbf{A}) - \mathbf{F}_0(\Delta\mathbf{A}))\mathbf{f}_0(\mathbf{x}) = \mathbf{f}_0(\mathbf{b}) - \mathbf{f}_0(\Delta\mathbf{b})$$

and the $N-1$ linear systems

$$(\mathbf{F}_1(\mathbf{A}) - \mathbf{F}_1(\Delta\mathbf{A}))\mathbf{f}_j(\mathbf{x}) = \mathbf{f}_j(\mathbf{b}) - \mathbf{f}_j(\Delta\mathbf{b}), \quad j = 1, 2, \dots, N-1.$$

From (9) and (8) we have that

$$\|\Delta\mathbf{b}\|^2 = \frac{1}{N}\left(\sum_{j=0}^{N-1} \|\mathbf{f}_j(\Delta\mathbf{b})\|^2\right), \quad \|\Delta\mathbf{A}\|^2 = \|\mathbf{F}_0(\Delta\mathbf{A})\|^2 + (N-1)\|\mathbf{F}_1(\Delta\mathbf{A})\|^2.$$

Thus we obtain that in the case of EBC structure, the STLS problem (21) is reduced to

$$\begin{array}{ll} \min_{\Delta\mathbf{A}_0, \Delta\mathbf{A}_1, \Delta\mathbf{b}} & \|\Delta\mathbf{A}_0\|^2 + \frac{1}{N}\|\Delta\mathbf{b}_0\|^2 + (N-1)\|\Delta\mathbf{A}_1\|^2 + \frac{1}{N}\sum_{i=1}^{N-1}\|\Delta\mathbf{b}_i\|^2 \\ \text{subject to} & (\mathbf{F}_0(\mathbf{A}) - \mathbf{F}_0(\Delta\mathbf{A}))\mathbf{f}_0(\mathbf{x}) = \mathbf{f}_0(\mathbf{b}) - \mathbf{f}_0(\Delta\mathbf{b}) \\ & (\mathbf{F}_1(\mathbf{A}) - \mathbf{F}_1(\Delta\mathbf{A}))\mathbf{f}_j(\mathbf{x}) = \mathbf{f}_j(\mathbf{b}) - \mathbf{f}_j(\Delta\mathbf{b}), \quad 1 \le j \le N-1, \\ & \Delta\mathbf{A} = \mathcal{M}(\Delta\mathbf{A}_0, \Delta\mathbf{A}_1). \end{array}$$

Making the change of variables

$$\begin{aligned} \mathbf{G}_0 &= \mathbf{F}_0(\Delta\mathbf{A}) = \Delta\mathbf{A}_0 + (N-1)\Delta\mathbf{A}_1, \\ \mathbf{G}_1 &= \mathbf{F}_1(\Delta\mathbf{A}) = \Delta\mathbf{A}_0 - \Delta\mathbf{A}_1, \\ \mathbf{c}_j &= \mathbf{f}_j(\Delta\mathbf{b}) = \sum_{i=0}^{N-1} \omega^{-ij}\Delta\mathbf{b}_i, \quad 0 \le j \le N-1, \\ \mathbf{z}_j &= \mathbf{f}_j(\mathbf{x}) = \sum_{i=0}^{N-1} \omega^{-ij}\mathbf{x}_i, \quad 0 \le j \le N-1, \end{aligned}$$

we obtain the following equivalent minimization problem:

$$\begin{array}{ll} \min_{\mathbf{G}_0, \mathbf{G}_1, \mathbf{c}_0, \dots, \mathbf{c}_{N-1}, \mathbf{z}_0, \dots, \mathbf{z}_{N-1}} & \|\mathbf{G}_0\|^2 + \frac{1}{N}\|\mathbf{c}_0\|^2 + (N-1)\|\mathbf{G}_1\|^2 + \frac{1}{N}\sum_{i=1}^{N-1}\|\mathbf{c}_i\|^2 \\ \text{subject to} & (\mathbf{F}_0(\mathbf{A}) - \mathbf{G}_0)\mathbf{z}_0 = \mathbf{f}_0(\mathbf{b}) - \mathbf{c}_0 \\ & (\mathbf{F}_1(\mathbf{A}) - \mathbf{G}_1)\mathbf{z}_j = \mathbf{f}_j(\mathbf{b}) - \mathbf{c}_j, \quad 1 \le j \le N-1, \end{array} \tag{22}$$

which is separable with respect to the groups of variables $(\mathbf{G}_0, \mathbf{c}_0, \mathbf{z}_0)$ and the variables set

$$\{(\mathbf{G}_1, \mathbf{c}_1, \mathbf{z}_1), \dots, (\mathbf{G}_{N-1}, \mathbf{c}_{N-1}, \mathbf{z}_{N-1})\}.$$

Therefore, the solution of the minimization problem (22) is the sum of the two minimization problems

$$\begin{array}{ll} \min_{\mathbf{G}_0, \mathbf{c}_0, \mathbf{z}_0} & \|\mathbf{G}_0\|^2 + \frac{1}{N}\|\mathbf{c}_0\|^2 \\ \text{subject to} & (\mathbf{F}_0(\mathbf{A}) - \mathbf{G}_0)\mathbf{z}_0 = \mathbf{f}_0(\mathbf{b}) - \mathbf{c}_0 \end{array} \tag{23}$$

and

$$\begin{array}{lll} (24) & \min_{\mathbf{G}_1,\mathbf{c}_1,\ldots,\mathbf{c}_{N-1},\mathbf{z}_1,\ldots,\mathbf{z}_{N-1}} & (N-1)\|\mathbf{G}_1\|^2 + \frac{1}{N}\sum_{i=1}^{N-1}\|\mathbf{c}_i\|^2 \\ & \text{subject to} & (\mathbf{F}_1(\mathbf{A})-\mathbf{G}_1)\mathbf{z}_j = \mathbf{f}_j(\mathbf{b}) - \mathbf{c}_j, \quad 1 \le j \le N-1. \end{array}$$

The minimization problem (23) is a TLS problem, and the second problem (24) is an MTLS problem. This gives rise to the following algorithm for solving the STLS problem for EBC matrices.

Algorithm EBCTLS for elementary block circulant matrices.

Input: $\mathbf{A}, \mathbf{b}$, where $\mathbf{A} = \mathcal{M}(\mathbf{A}_0, \mathbf{A}_1) \in \mathbb{F}^{Nm\times Nm}$ is an EBC matrix and $\mathbf{b} = (\mathbf{b}_0^T, \mathbf{b}_1^T, \ldots, \mathbf{b}_{N-1}^T)^T$ such that $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{F}^{m\times n}$ and $\mathbf{b}_0, \ldots, \mathbf{b}_{N-1} \in \mathbb{F}^m$.

Output: $\Delta\mathbf{A}, \Delta\mathbf{b}, \mathbf{x}$, where $\Delta\mathbf{A} \in \mathbb{F}^{Nm\times Nm}$ is an elementary block circulant matrix, $\Delta\mathbf{b} \in \mathbb{F}^{Nm}$, and $\mathbf{x} \in \mathbb{F}^{Nn}$ is the STLS solution.

1. Calculate the two different DFT components of $\mathbf{A}$,

$$\mathbf{F}_0(\mathbf{A}) = \mathbf{A}_0 + (N-1)\mathbf{A}_1, \quad \mathbf{F}_1(\mathbf{A}) = \mathbf{A}_0 - \mathbf{A}_1,$$

   and the N DFT components of $\mathbf{b}$,

$$\mathbf{f}_j(\mathbf{b}) = \sum_{i=0}^{N-1} \omega^{-ij}\mathbf{b}_i, \quad 0 \le j \le N-1.$$

2. Call the TLS algorithm with input $\left(\mathbf{F}_0(\mathbf{A}), \mathbf{f}_0(\mathbf{b}), \frac{1}{N}\right)$ and obtain an output $(\mathbf{G}_0, \mathbf{c}_0, \mathbf{z}_0)$.
3. Call the MTLS algorithm with input $(\mathbf{F}_1(\mathbf{A}), \mathbf{f}_1(\mathbf{b}), \ldots, \mathbf{f}_{N-1}(\mathbf{b}), \frac{1}{N(N-1)})$ and obtain an output $\mathbf{G}_1, \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{N-1}$ and $\mathbf{z}_1, \ldots, \mathbf{z}_{N-1}$.
4. Denote $\mathbf{G} = \mathcal{M}(\mathbf{G}_0, \mathbf{G}_1)$, $\mathbf{c} = (\mathbf{c}_0^T, \mathbf{c}_1^T, \ldots, \mathbf{c}_{N-1}^T)^T$ and $\mathbf{z} = (\mathbf{z}_0^T, \mathbf{z}_1^T, \ldots, \mathbf{z}_{N-1}^T)^T$. The output of the EBCTLS algorithm is computed by applying the inverse DFT of $\mathbf{G}, \mathbf{c}$, and $\mathbf{z}$. The optimal solution of the STLS problem (21) is then $\Delta\mathbf{A} = \mathcal{M}(\Delta\mathbf{A}_0, \Delta\mathbf{A}_1)$, $\Delta\mathbf{b} = (\Delta\mathbf{b}_0^T, \Delta\mathbf{b}_1^T, \ldots, \Delta\mathbf{b}_{N-1}^T)^T$, and $\mathbf{x} = (\mathbf{x}_0^T, \mathbf{x}_1^T, \ldots, \mathbf{x}_{N-1}^T)$, where

$$\begin{aligned} \Delta\mathbf{A}_0 &= \mathbf{F}_0^{-1}(\mathbf{G}) = \frac{1}{N}(\mathbf{G}_0 + (N-1)\mathbf{G}_1), \\ \Delta\mathbf{A}_1 &= \mathbf{F}_1^{-1}(\mathbf{G}) = \frac{1}{N}(\mathbf{G}_0 - \mathbf{G}_1), \\ \Delta\mathbf{b}_j &= \mathbf{f}_j^{-1}(\mathbf{c}) = \frac{1}{N}\sum_{i=0}^{N-1}\omega^{ij}\mathbf{c}_i, \quad 0 \le j \le N-1, \\ \mathbf{x}_j &= \mathbf{f}_j^{-1}(\mathbf{z}) = \frac{1}{N}\sum_{i=0}^{N-1}\omega^{ij}\mathbf{z}_i, \quad 0 \le j \le N-1. \end{aligned}$$

Remarks.

1. Steps 2 and 3 of Algorithm BCTLS require the following conditions to be satisfied (see step 2 of the TLS and MTLS algorithms):

$$\sigma_n(\mathbf{F}_0(\mathbf{A})) > \sigma_{n+1}\left(\mathbf{F}_0(\mathbf{A}), \frac{1}{\sqrt{N}}\mathbf{f}_0(\mathbf{b})\right),$$

$$\sigma_n(\mathbf{F}_1(\mathbf{A})) > \sigma_{n+1}\left(\mathbf{F}_1(\mathbf{A}), \frac{1}{\sqrt{N(N-1)}}\mathbf{f}_1(\mathbf{b}), \ldots, \frac{1}{\sqrt{N(N-1)}}\mathbf{f}_{N-1}(\mathbf{b})\right).$$

2. In the case $\mathbb{F} = \mathbb{R}$, the EBCTLS algorithm generates a real solution $\Delta\mathbf{A}, \Delta\mathbf{b}$, and $\mathbf{x}$. The proof is almost identical to the proof of Theorem 4.1.

6. Computational results. In this section we compare the SVD-based algorithms, BCTLS and EBCTLS, which find the global optimum of the STLS problem with BC and EBC matrices, respectively, to the following three methods.

1. *The least squares* (LS) *method.* Here, we enforce $\Delta\mathbf{A}$ to be zero, and we choose a minimal norm $\Delta\mathbf{b}$. This is of course a very naive algorithm, and it assumes that the nominal value of the matrix is the true value. If the matrix $\mathbf{A}$ has full column rank, then the LS solution is given by $\mathbf{x}_{LS} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ and $\Delta\mathbf{b} = \mathbf{b} - \mathbf{A}\mathbf{x}_{LS}$ (see, e.g., [1]).
2. *The TLS method.* The (unstructured) TLS method disregards the linear structure of the matrix $\mathbf{A}$ and seeks a perturbation matrix $\Delta\mathbf{A}$ and a perturbation vector $\Delta\mathbf{b}$ that minimize $\|\Delta\mathbf{A}\|^2 + \|\Delta\mathbf{b}\|^2$ subject to the consistency equation $\mathbf{b} - \Delta\mathbf{b} \in \mathrm{Range}(\mathbf{A} - \Delta\mathbf{A})$.
3. *The structured total least norm* (STLN) *method.* This method was introduced and studied in [7]. The STLN method (with 2-norm) is an iterative method for solving STLS problems with arbitrary linear structure. In each iteration of the STLN algorithm, a least squares problem is solved. In our problem, the size of the matrix in the least squares problem is $(Nm+2mn)\times(Nn+2mn)$. If $N \ll m, n$, then the complexity *per iteration* of the STLN method is $O(m^3n^3)$, which is computationally very demanding. The algorithm is essentially a Newton-like method applied to a nonconvex function. There is no theoretical proof of convergence and, even when convergence occurs, there is no guarantee that it converges to a *global* optimum.

The first example considers a block circulant structure with $N = 2, m = 28, n = 4$. We assume that there is a "correct" system,

$$\mathbf{b}^{(c)} = \mathbf{A}^{(c)}\mathbf{x}^{(c)},$$

where $\mathbf{A}^{(c)}$ is given by

$$\mathbf{A}^{(c)} = \begin{pmatrix} \mathbf{A}_0^{(c)} & \mathbf{A}_1^{(c)} \\ \mathbf{A}_1^{(c)} & \mathbf{A}_0^{(c)} \end{pmatrix}$$

with $\mathbf{A}_0^{(c)}, \mathbf{A}_1^{(c)} \in \mathbb{R}^{28\times 4}$. Each component of the matrices $\mathbf{A}_0^{(c)}$ and $\mathbf{A}_1^{(c)}$ was chosen to be -1 or 1 with probability 1/2 independently of the other components. Each component of the correct vector $\mathbf{x}^{(c)}$ was a randomly chosen integer number between -10 and 9. The actual data $\mathbf{A}, \mathbf{b}$ contains noise and is thus a perturbation of $\mathbf{A}^{(c)}$ and $\mathbf{b}^{(c)}$ (where $\mathbf{A}$ has the same structure as $\mathbf{A}^{(c)}$). In our experiments each component of $\mathbf{A}$ and $\mathbf{b}$ is the corresponding component of $\mathbf{A}^{(c)}$ and $\mathbf{b}^{(c)}$ plus a normal random variable with zero expectation and standard deviation equal to 0.2. We considered three quantities to describe the results: the relative error of the matrix, the relative error of the solution, and the function value:

$$\begin{aligned} A_{err} &= \|\mathbf{A} - \Delta\mathbf{A} - \mathbf{A}^{(c)}\|/\|\mathbf{A}^{(c)}\|, \\ x_{err} &= \|\mathbf{x} - \mathbf{x}^{(c)}\|/\|\mathbf{x}^{(c)}\|, \\ value &= \|\Delta\mathbf{A}\|^2 + \|\Delta\mathbf{b}\|^2. \end{aligned}$$

The results given in the table below are the average over 200 realizations of the noise affecting the matrix $\mathbf{A}^{(c)}$ and the vector $\mathbf{b}^{(c)}$.

Method	Value
LS	16.4376
TLS	6.2414
STLN	2.7601
BCTLS	2.7601

In this example the STLN algorithm converged to the *global* optimum. The average number of iterations of the STLN algorithm was 14 and ranged between 11 and 17. It also can be seen that both the BCTLS and the STLN solutions were better than the LS solution in all aspects.

In our second example, we considered an EBC matrix with $N = 3$, $m = 16$, and $n = 4$. Hence, $\mathbf{A}^{(c)}$ is given by

$$\mathbf{A}^{(c)} = \begin{pmatrix} \mathbf{A}_0^{(c)} & \mathbf{A}_1^{(c)} & \mathbf{A}_1^{(c)} \\ \mathbf{A}_1^{(c)} & \mathbf{A}_0^{(c)} & \mathbf{A}_1^{(c)} \\ \mathbf{A}_1^{(c)} & \mathbf{A}_1^{(c)} & \mathbf{A}_0^{(c)} \end{pmatrix},$$

where $\mathbf{A}_0^{(c)}, \mathbf{A}_1^{(c)} \in \mathbb{R}^{16\times 4}$ and the components of $\mathbf{A}_0^{(c)}$ and $\mathbf{A}_1^{(c)}$ were randomly chosen to be 0 or 1. Each component of the correct vector $\mathbf{x}^{(c)}$ was a randomly chosen integer number between -10 and 9. The results in the table below are the average over 100 realizations of the noise

Method	Value
LS	523.9572
TLS	34.4647
STLN	12.4633
EBCTLS	6.1789

As can be seen from the above table, the STLN method in this case is suboptimal and does not converge to a global optimum. Moreover, in 17 out of the 100 instances the STLN algorithm did not converge at all, and in all 83 other cases it converged, after hundreds of iterations, but not to a global optimum.

REFERENCES

[1] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore MD, 1996.
[2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, McGraw–Hill, New York, 1990.
[3] H. Murakami, *Discrete wavelet transform based on cyclic convolution*, IEEE Trans. Signal Process., 52 (2004), pp. 165–174.
[4] G. H. Golub and C. F. Van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.
[5] B. De Moor, *Structured total least squares and L_2 approximation problems*, Linear Algebra Appl., 188/189 (1993), pp. 163–207.
[6] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, Frontiers App. Math. 9, SIAM, Philadelphia, 1991.
[7] J. B. Rosen, H. Park, and J. Glick, *Total least norm formulation and solution for structured problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 110–126.
[8] B. De Moor, *Total least squares for affinely structured matrices and the noisy realization problem*, IEEE Trans. Signal Process., 42 (1994), pp. 3104–3113.
[9] T. J. Abatzoglou, J. M. Mendel, and G. A. Harada, *The constrained total least squares technique and its applications to harmonic supperesolution*, IEEE Trans. Signal Process., 39 (1991), pp. 1070–1087.
[10] N. Mastronardi, P. Lemmerling, and S. Van Huffel, *Fast structured total least squares algorithm for solving the basic deconvolution problem*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 533–553.

[11] P. Lemmerling and S. Van Huffel, *Analysis of the structured total least squares problem for Hankel/Toeplitz matrices*, Numer. Algorithms, 27 (2001), pp. 89–114.

[12] S. Van Huffel and J. Vandewalle, *Analysis and properties of the generalized total least squares problem $AX \approx B$ when some or all columns in A are subject to error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 294–315.

[13] A. Beck, Y. C. Eldar, and A. Ben-Tal, *Minimax mean-squared error estimation of multichannel signals.* IEEE Trans. Inform. Theory, submitted.

[14] A. Beck, A. Ben-Tal, and Y. C. Eldar, *Robust mean-squared error estimation of multiple signals in linear systems affected by model and noise uncertainties*, Math. Program., to appear.

[15] A. K. Katsaggelos, K. T. Lay, and N. P. Galatsanos, *A general framework for frequency domain multi-channel signal processing*, IEEE Trans. Image Process., 2 (1993), pp. 417–420.

[16] M. Sznaier, O. Camps, and M. C. Mazzaro, *Finite horizon model reduction of a class of neutrally stable systems with applications to texture synthesis and recognition*, in Proceedings of the 43rd IEEE Conference on Decision and Control, Atlantis, Paradise, Bahamas, 2004.