

Tensor SVD and distributed control

Ram V. Iyer

Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409-1042.

ABSTRACT

The (approximate) diagonalization of symmetric matrices has been studied in the past in the context of distributed control of an array of collocated smart actuators and sensors. For distributed control using a two dimensional array of actuators and sensors, it is more natural to describe the system transfer function as a complex tensor rather than a complex matrix. In this paper, we study the problem of approximately diagonalizing a transfer function tensor for a locally spatially invariant system, and study its application along with the technique of recursive orthogonal transforms to achieve distributed control for a smart structure.

Keywords: Distributed Control, Spatial Invariance, Recursive Orthogonal Transforms, Tensor Singular Value Decomposition, Smart Structures, Two Dimensional Array, Collocated Actuators and Sensors, Active Noise Suppression

1. INTRODUCTION

The area of active noise suppression and interior acoustics control involves distributed actuation and sensing, usually with piezoelectric devices.¹⁻⁴ In these applications the actuators and sensors are collocated and distributed. If these actuator-sensor units are also provided with processing power, then it is possible to envisage a truly distributed control system. In the following, we will refer to such actuator-sensor-processor units as *controller units*. Though a general distributed parameter system would have to be treated using tools of partial differential equations, considerable simplification can be achieved if one assumes that the controlled system is *spatially invariant*. In the literature, there are two different notions of spatial invariance, one global and the other local. The global notion of spatial invariance for a linear system described by a partial differential equation on an infinite non-compact spatial domain, is akin to reducing the problem to one on a compact domain with periodic boundary conditions.⁵ Mathematically, this reduction procedure amounts to “factoring” the domain using a symmetry group. The factored system (with full actuation and sensing) which is a partial differential equation, but now described on a spatially compact domain, could still exhibit extensive coupling. The notion of local spatial invariance⁶ is for a system such as those considered in active noise suppression, defined on a compact spatial domain, with finite number of collocated actuators and sensors laid out in an array or matrix fashion. Our notion of local spatial invariance is slightly more general and includes that of Kantor.⁶ Consider the transfer function between any pair of actuator inputs and sensor outputs on the array of collocated actuators and sensors. If this transfer function depends only on the spatial distance between the actuator and the sensor considered, then the controlled system is said to be *locally spatially invariant*.

Recently, Chou, Flamm, Guthart and Ueberschaer⁷; Bamieh, Paganini and Dahleh⁵; and Kantor and Krishnaprasad^{6, 8, 9} have considered the problem of distributed control of spatially invariant systems from slightly different points of view. Bamieh, Paganini and Dahleh consider the distributed system to arise from a linear (parabolic) partial differential equation that is described on an infinite domain with spatial periodicity, and full control and sensing - that is control authority and sensing capability can be exercised at each point of the spatial domain. The (global) spatial invariance that they consider leads naturally to the idea of block diagonalization via the Fourier transform applied to the spatial variable. However, inside each block the system could still be extensively coupled. For spatially localized control, they arrive at a condition that is based on a Hamiltonian function having a rational realization. The spatial invariance concept used by Bamieh, Paganini and Dahleh has been considered by several authors in the past (see references in⁵).

On the other hand, Chou, Flamm, Guthart and Ueberschaer⁷ and Kantor, Krishnaprasad^{6, 8, 9} consider systems with what we term local spatial invariance. The term *local spatial invariance* implies that the transfer function of a system on a compact domain with finite number of controller units has a special form – specifically,

the transfer function between the i -th input and j -th output only depends on the physical distance between the inputs and outputs. Chou et al.⁷ consider a finite rectangular array of collocated actuators and sensors, and implicitly make the assumption of local spatial invariance given in this paper. They consider the problem of transforming the given system into a sparser form (not necessarily diagonal form) using wavelet transforms in the spatial domain. They observe that for Calderon-Zygmund operators, the coefficients obtained after a wavelet transformation have certain decay properties. Thus by observing the decay of Green's function appropriate for the problem, they estimate the number of wavelet coefficients needed to implement a local controller. One of the interesting aspects of their work, is that they realize that a two dimensional array of controller units require a tensor transfer function description. They use a tensor product of 1-D wavelet transforms to obtain the wavelet transform of the entire tensor. Even though the transformed tensor has a sparse characteristic which enables the design of a controller that uses local information, the implementation of the wavelet transform might need global data rather than local data.

Kantor and Krishnaprasad consider a similar problem to Chou et al., that is, a distributed system on a finite domain with the controller units arranged in a linear or circular fashion^{6,9} or in the form of a two-dimensional array.⁸ They investigate the problem of implementation of orthogonal wavelet transforms *using local information* on the input and output signals so as to achieve approximate diagonalization of the transfer function. Their conclusion is that by systematically applying a sequence of multiplications by orthogonal matrices and permutations - each of which can be accomplished using local data - one can indeed achieve approximate diagonalization. This sequence of operations (that they term *recursive orthogonal transforms*) can be performed locally with each actuator only using information from nearby sensors. Practical implementation was shown to be possible by showing that there is a finite sequence of orthogonal transformations and permutations that achieves exact diagonalization. Once the transfer function is approximately diagonalized, it is clear that only localized controllers at each actuator is required to control the system, and decentralization has been achieved for linear arrays.

In this paper, we combine the lines of investigation of Chou et al., and Kantor, Krishnaprasad. One of the problems that was not addressed by the latter authors is that for two dimensional arrays,⁸ the approximate diagonalization might mean that data from non-neighbouring controller units might be required at an intermediate computation step. This problem arises due to the labeling of a two-dimensional array using only one index. An early recognition of the role tensors play in electrical networks can be found in Kron.¹⁰ However, as far as we are aware, there does not seem to be a systematic study of tensors in the control theory framework apart from its mention in Chou et al. In this paper, we investigate transfer function tensors that arise in distributed control systems and the use of the *tensor singular value decomposition* (TSVD)¹¹ for designing distributed controllers that only rely on local sensor information. The interesting connection with Chou et al. is that while they arrive at a *sparse but not diagonal structure* for the wavelet transformed transfer function *by choosing the appropriate transform*, we argue using the results of TSVD that diagonalization is not possible and a sparse structure is perhaps the best result one can achieve! As the orthogonal transforms that result in the TSVD can be approximated by recursive orthogonal transforms, we complete the link between Chou et al. and Kantor, Krishnaprasad.

2. TRANSFER FUNCTION MATRICES AND TENSORS IN DISTRIBUTED CONTROL

In this section, we examine symmetry properties enjoyed by transfer function matrices and tensors arising from the property of local spatial invariance and their implication for numerical computation. As mentioned in the Introduction, Kantor and Krishnaprasad consider the controller units to be arranged in a linear or circular array,^{6,9} or in the case of a two dimensional array, numbered using only one index.⁸ In both of these cases the transfer function matrix $G_i^j(s) = \frac{Y^j(s)}{U^i(s)}$ (where U^i is the i -th input and Y^j is the j -th output) is a (1,1) tensor. Here $s \in \mathbb{C}$ denotes the Laplace transform variable. However, for more general topologies where the controller units are distributed in two or more dimensions (see Frecker¹² and references therein), a better numbering system that corresponds to the number of spatial dimensions is needed. This is because no matter how one numbers the actuated elements using one variable, one can always find numerically neighboring elements that are physically separate!. Please see Figure 2 at this time. To overcome this problem, it is natural to introduce two variables for actuated elements distributed in a plane and three variables for actuated elements distributed on the surface of a

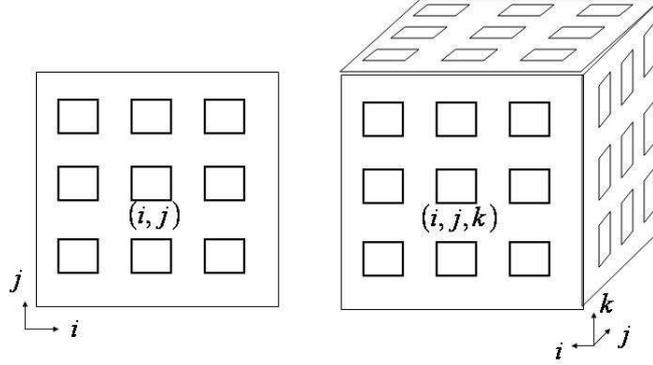


Figure 1. A distributed network of collocated smart actuators and sensors

three dimensional structure (see Figure 1). Such topologies arise in active noise control, for example see Savran, Atalla and Hall.³ With the inputs and outputs numbered $U^{ij}(s)$ and $Y^{ij}(s)$, we have a correlation between the labels and the physical layout. Then the transfer function is given by

$$G_{ij}^{kl}(s) = \frac{Y^{kl}(s)}{U^{ij}(s)}; \quad i, k, \in \{1, \dots, m\}; \quad j, l, \in \{1, \dots, n\}, \quad (1)$$

and is thus a complex $(2, 2)$ tensor. A three dimensional arrangement of actuated elements would lead to a $(3, 3)$ tensor. The interesting aspect of local spatial invariance is that it leads to transfer function tensors with various types of symmetry.

2.1. Symmetric, Persymmetric and Circulant Matrices and $(1,1)$ Tensors

First, we define the concepts of symmetry and persymmetry for $(1,1)$ tensors $G_i^j(s)$; $i, j \in \{1, \dots, N\}$. As these tensors can be represented by square matrices $G(s)$ whose (i, j) -th entry is the transfer function $G_i^j(s)$, we can simply borrow the concepts of symmetry and persymmetry from matrix theory.^{13,14} The concept of symmetry for $(1,1)$ tensors is then: $G_i^j(s) = G_j^i(s)$; $i, j \in \{1, \dots, N\}$. Persymmetry can be defined to be: $G_i^j(s) = G_{(N+1-j)}^{(N+1-i)}(s)$; $i, j \in \{1, \dots, N\}$. The condition of symmetry can be simply expressed as:

$$G(s) = G^T(s), \quad (2)$$

where the right-hand-side denotes the transpose of $G(s)$, while the condition of persymmetry can be expressed as:

$$G(s) = J G^T(s) J; \quad \text{where } J = [e_n \ e_{n-1} \ \dots \ e_1]; \quad \text{with } e_k = \underbrace{[0 \ \dots \ 0]_{k-1}}_{k-1} \uparrow_k \underbrace{[0 \ \dots \ 0]_{n-k}}_{n-k}. \quad (3)$$

A centrosymmetric matrix $G(s)$ is one that satisfies¹⁴:

$$G(s) = J G(s) J. \quad (4)$$

The following lemma is immediate.

LEMMA 2.1. *If a matrix satisfies any two of symmetry, persymmetry and centrosymmetry, then it also satisfies the third.*

Proof. The proof is straightforward and uses the fact that $J = J^{-1}$. \square

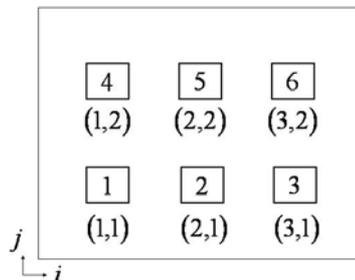


Figure 2. A 3×2 network of collocated smart actuators and sensors

The set of eigenvectors for a persymmetric matrix has an interesting structure,¹⁴ and furthermore, a persymmetric matrix has a canonical decomposition depending on whether is odd or even dimensional. An investigation along these lines needs to be done for persymmetric tensors. A particular case of a persymmetric (but not symmetric) matrix is a circulant matrix.¹³ It is very easy to show that one can diagonalize any circulant matrix via the Discrete Fourier Transform (DFT),^{15,16} as the columns of the normalized Discrete Fourier Transform (denoted by Q) is identical to the eigenvector space of any circulant matrix. In other words, we have for each $s \in \mathbb{C}$: $G(s) = Q \Sigma(s) Q^*$, where Q^* is the conjugate-transpose of Q and represents the normalized Inverse Discrete Fourier Transform (IDFT) matrix. The following equations show how a circulant transfer function can be diagonalized by applying the IDFT to the output and DFT to the input signals:

$$\begin{aligned} Y(s) &= G(s) U(s) \\ G(s) &= Q \Sigma(s) Q^* \\ \tilde{Y}(s) &= \Sigma(s) \tilde{U}(s) \end{aligned} \tag{5}$$

$$\tilde{Y}(s) = Q^* Y(s) \implies \tilde{y}(t) = Q^* y(t) \tag{6}$$

$$U(s) = Q \tilde{U}(s) \implies u(t) = Q^* \tilde{u}(t) \tag{7}$$

Due to the special eigenspace structure of all circulant matrices, one can simply apply the IDFT to the output signals $y(t)$ of the sensors; send this to the local controllers designed using the diagonal system (5); then take the output $\tilde{u}(t)$ of a controllers and construct the signal $u(t)$ by the DFT. A similar observation can be found in Kantor,⁶ who further shows that we can approximate Q by a sequence of local orthogonal transformations and permutations (in other words, a recursive orthogonal transform), and in essence, achieve approximate diagonalization. This result easily carries over to a two dimensional array in the form of a torus.

2.2. Symmetric and Persymmetric (2,2) Tensors

Now, consider for simplicity a $m \times n$ network (with $m = 3$ and $n = 2$) of controller units as shown in Figure 2. We have shown two different types of indexing schemes for this network in Figure 2. The first is a single indexing scheme where each of the controller units is given a unique single number from the set $\{1, \dots, mn\}$, while the second is a double indexing scheme (i, j) where $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$. In the single indexing scheme, one obtains a transfer function matrix $G_i^j(s) = \frac{Y^j(s)}{U^i(s)}$; $i, j \in \{1, \dots, mn\}$, where U^i is the i -th input and Y^j is the j -th output. On the other hand, with the double indexing scheme, one obtains a transfer function tensor $G_{ij}^{kl}(s) = \frac{Y^{kl}(s)}{U^{ij}(s)}$ where $i, k, \in \{1, \dots, m\}$ and $j, l \in \{1, \dots, n\}$.

The property of local spatial invariance can be expressed more easily in the matrix numbering scheme:

$$G_{ij}^{kl}(s) = g_{|k-i|, |l-j|}(s) \tag{8}$$

Observe that this property implies that the transfer function tensor is *symmetric and per-symmetric*:

$$G_{ij}^{kl}(s) = G_{kl}^{ij}(s) \quad (\text{Symmetry}) \quad (9)$$

$$G_{ij}^{kl}(s) = G_{(m+1-k)(n+1-l)}^{(m+1-i)(n+1-j)}(s) \quad (\text{Persymmetry}). \quad (10)$$

However, the single indexing schemes need not always lead to symmetry and persymmetry. For instance, the indexing scheme shown in Figure 2 leads to the following transfer function when local spatial invariance is taken into account:

$$G(s) = \left[\begin{array}{ccc|ccc} g_0(s) & g_1(s) & g_2(s) & g_1(s) & g_3(s) & g_4(s) \\ g_1(s) & g_0(s) & g_1(s) & g_3(s) & g_1(s) & g_3(s) \\ g_2(s) & g_1(s) & g_0(s) & g_4(s) & g_3(s) & g_1(s) \\ \hline g_1(s) & g_3(s) & g_4(s) & g_0(s) & g_1(s) & g_2(s) \\ g_3(s) & g_1(s) & g_3(s) & g_1(s) & g_0(s) & g_1(s) \\ g_4(s) & g_3(s) & g_1(s) & g_2(s) & g_1(s) & g_0(s) \end{array} \right]$$

Notice that the above matrix can be written in the block diagonal form:

$$G(s) = \left[\begin{array}{cc} G_0(s) & G_1(s) \\ G_1(s) & G_0(s) \end{array} \right]$$

which is a *block symmetric and persymmetric* matrix, where each of the blocks $\{G_k; k = 0, 1\}$ is itself a 3×3 symmetric and persymmetric matrix! Furthermore, there are several indexing schemes that lead to a similar description of the transfer function matrix – for instance those considered in Figures 3(a) - 3(c). However, the indexing system considered in Figure 3(d) only leads to a symmetric but not a persymmetric matrix:

$$G(s) = \left[\begin{array}{ccc|ccc} g_0(s) & g_1(s) & g_3(s) & g_4(s) & g_2(s) & g_1(s) \\ g_1(s) & g_0(s) & g_1(s) & g_2(s) & g_4(s) & g_3(s) \\ g_3(s) & g_1(s) & g_0(s) & g_1(s) & g_3(s) & g_1(s) \\ \hline g_4(s) & g_2(s) & g_1(s) & g_0(s) & g_1(s) & g_3(s) \\ g_2(s) & g_4(s) & g_3(s) & g_1(s) & g_0(s) & g_1(s) \\ g_1(s) & g_3(s) & g_1(s) & g_3(s) & g_1(s) & g_0(s) \end{array} \right]$$

In this case, the block description:

$$G(s) = \left[\begin{array}{cc} G_0(s) & G_1(s) \\ G_1(s) & G_0(s) \end{array} \right]$$

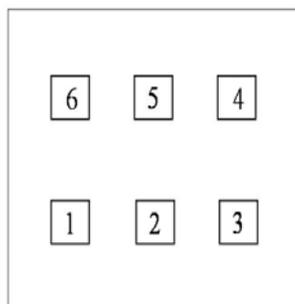
which is a *block symmetric and persymmetric* matrix, where the block $G_0(s)$ is a 3×3 symmetric and persymmetric matrix, but the block $G_1(s)$ is only symmetric.

A similar situation arises for a 3×3 array – consider Figure 4. Invoking local spatial invariance, we can say that the transfer function for the array given in Figure 4(a) (using the single indexing scheme) has the form:

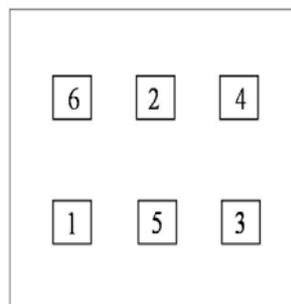
$$G(s) = \left[\begin{array}{ccc|ccc|ccc} g_0(s) & g_1(s) & g_2(s) & g_1(s) & g_3(s) & g_4(s) & g_2(s) & g_4(s) & g_5(s) \\ g_1(s) & g_0(s) & g_1(s) & g_3(s) & g_1(s) & g_3(s) & g_4(s) & g_2(s) & g_4(s) \\ g_2(s) & g_1(s) & g_0(s) & g_4(s) & g_3(s) & g_1(s) & g_5(s) & g_4(s) & g_2(s) \\ \hline g_1(s) & g_3(s) & g_4(s) & g_0(s) & g_1(s) & g_2(s) & g_1(s) & g_3(s) & g_4(s) \\ g_3(s) & g_1(s) & g_3(s) & g_1(s) & g_0(s) & g_1(s) & g_3(s) & g_1(s) & g_3(s) \\ g_4(s) & g_3(s) & g_1(s) & g_2(s) & g_1(s) & g_0(s) & g_4(s) & g_3(s) & g_1(s) \\ \hline g_2(s) & g_4(s) & g_5(s) & g_1(s) & g_3(s) & g_4(s) & g_0(s) & g_1(s) & g_2(s) \\ g_4(s) & g_2(s) & g_4(s) & g_3(s) & g_1(s) & g_3(s) & g_1(s) & g_0(s) & g_1(s) \\ g_5(s) & g_4(s) & g_2(s) & g_4(s) & g_3(s) & g_1(s) & g_2(s) & g_1(s) & g_0(s) \end{array} \right] \quad (11)$$

Again, notice that the above matrix can be written in the block diagonal form:

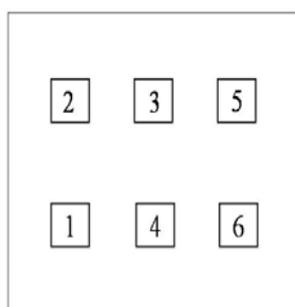
$$G(s) = \left[\begin{array}{ccc} G_0(s) & G_1(s) & G_2(s) \\ G_1(s) & G_0(s) & G_1(s) \\ G_2(s) & G_1(s) & G_0(s) \end{array} \right]$$



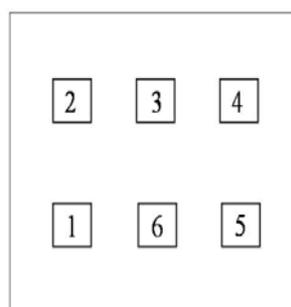
(a) Array indexing scheme 2



(b) Array indexing scheme 3



(c) Array indexing scheme 4



(d) Array indexing scheme 5

Figure 3. Possible different array indexing schemes for a 3×2 distributed control system

which is a *block symmetric and persymmetric* matrix, where each of the blocks $\{G_k; k = 0, 1, 2\}$ is itself a 3×3 symmetric and persymmetric matrix. There are a number of indexing schemes that result in such a transfer function matrix. However, the indexing scheme in Figure 4(b) results in the transfer function form:

$$G(s) = \begin{bmatrix} g_0(s) & g_1(s) & g_2(s) & g_4(s) & g_5(s) & g_4(s) & g_2(s) & g_1(s) & g_3(s) \\ g_1(s) & g_0(s) & g_1(s) & g_3(s) & g_4(s) & g_2(s) & g_4(s) & g_3(s) & g_1(s) \\ g_2(s) & g_1(s) & g_0(s) & g_1(s) & g_2(s) & g_4(s) & g_5(s) & g_4(s) & g_3(s) \\ \hline g_4(s) & g_3(s) & g_1(s) & g_0(s) & g_1(s) & g_3(s) & g_4(s) & g_2(s) & g_1(s) \\ g_5(s) & g_4(s) & g_2(s) & g_1(s) & g_0(s) & g_1(s) & g_2(s) & g_4(s) & g_3(s) \\ g_4(s) & g_2(s) & g_4(s) & g_3(s) & g_1(s) & g_0(s) & g_1(s) & g_3(s) & g_1(s) \\ \hline g_2(s) & g_4(s) & g_5(s) & g_4(s) & g_2(s) & g_1(s) & g_0(s) & g_1(s) & g_3(s) \\ g_1(s) & g_3(s) & g_4(s) & g_2(s) & g_4(s) & g_3(s) & g_1(s) & g_0(s) & g_1(s) \\ g_3(s) & g_1(s) & g_3(s) & g_1(s) & g_3(s) & g_1(s) & g_3(s) & g_1(s) & g_0(s) \end{bmatrix}$$

This matrix can be written in the block diagonal form:

$$G(s) = \begin{bmatrix} G_0(s) & G_1(s) & G_2(s) \\ G_1^T(s) & G_0(s) & G_3(s) \\ G_2^T(s) & G_3(s) & G_0(s) \end{bmatrix},$$

where the block $G_0(s)$ is symmetric and persymmetric; the block $G_3(s)$ is symmetric but not persymmetric; and the other blocks are neither. Overall, the transfer function matrix is only symmetric and not persymmetric. It is apparent that the indexing scheme has destroyed or perhaps hidden the special form that can be seen in the form (11).

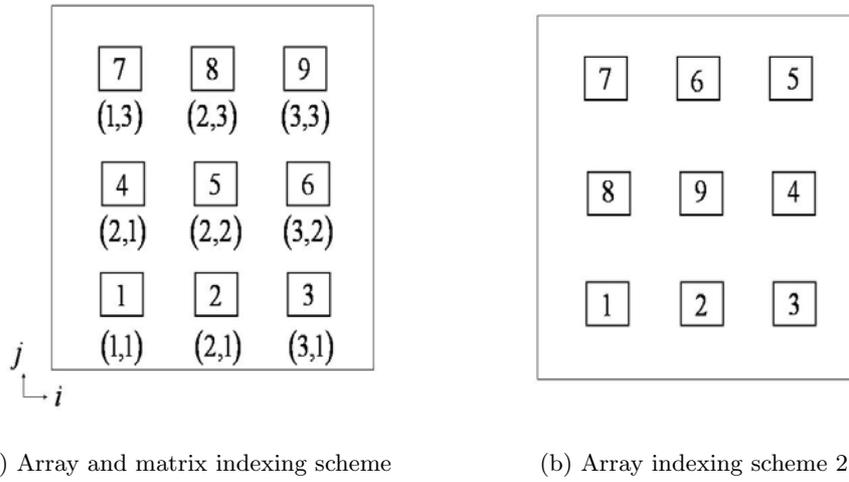


Figure 4. Two different array indexing schemes for a 3×3 distributed control system

It is clear that the indexing scheme is an artifice, and has the capacity to make the numerical computation of the orthogonal transforms perhaps more difficult. More importantly, if one uses recursive orthogonal transforms on the resulting transfer function matrix with the goal of approximate diagonalization, then the resulting form could have non-zero entries along the main diagonal⁸ which defeats the purpose of local computation. For instance, in the indexing scheme seen in Figure 4(a), the controller units with indices 3 and 4 are not physically near. To overcome this problem, let us consider the tensor transfer function description of the system in Figure 4(a):

$$G_{ij}^{kl} = \begin{cases} g_0(s) & \text{if } i = k; j = l \\ g_1(s) & \text{if } |i - k| + |j - l| = 1 \\ g_2(s) & \text{if } |i - k| + |j - l| = 2; \text{ with } i = k \text{ or } j = l \\ g_3(s) & \text{if } |i - k| + |j - l| = 2; \text{ with } i \neq k \text{ and } j \neq l \\ g_4(s) & \text{if } |i - k| + |j - l| = 3; \\ g_5(s) & \text{if } |i - k| + |j - l| = 4; \end{cases} \quad (12)$$

It is clear that no matter how one chose to index the system as a two dimensional array, one always gets the same transfer function matrix! Furthermore, this transfer function matrix is both symmetric (as $G_{ij}^{kl}(s) = G_{kl}^{ij}(s)$) and persymmetric (as $G_{ij}^{kl}(s) = G_{(4-k)(4-l)}^{(4-i)(4-j)}(s)$).

3. TENSOR SINGULAR VALUE DECOMPOSITION

Our description of Tensor Singular Value Decomposition (TSVD) is from Lathauwer, De Moor and Vandewalle¹¹ who call it Higher-Order Singular Value Decomposition (HOSVD). As we are primarily interested in (2,2)

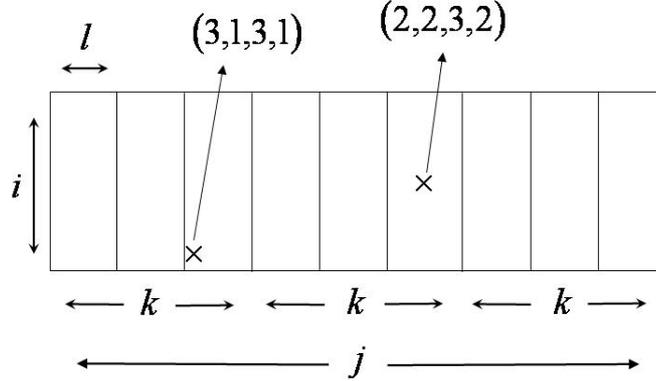


Figure 5. The 1 unfolding of the transformation tensor for a 3×3 array.

tensors, we will specialize their theory to this case. For $(1, 1)$ tensors, the description will match the usual matrix singular value decomposition (SVD). As noted by Lathauwer, De Moor and Vandewalle,¹¹ and by Kofidis and Regalia,¹⁷ it is possible to extend the usual notion of SVD from matrices to tensors in several different ways, depending on what property is emphasized. In the case of matrices, the SVD has 4 properties simultaneously: (a) it diagonalizes a given matrix; (b) the right and left transformation matrices are unitary; and (c) it can be obtained as the result of solving the least-squares diagonality criterion $\sum_i |s_{ii}|^2$ (see Golub and Van Loan,¹⁸ page 426); (d) the number of nonzero diagonal elements in the singular value matrix is equal to the rank of the given matrix. However, in the case of tensors each of these properties do not imply the others, and so one can extend the SVD to tensors while emphasizing any one of these particular properties. But as noted by Lathauwer, De Moor and Vandewalle,¹¹ the most natural extension is obtained by emphasizing the property that the transformation matrices be unitary. However, this means that the core singular value tensor need not be diagonal. This ties in with the result that we believe was observed by Chou et al.⁷ in the numerical studies on a rectangular array of controller units.

As we use the notation of Lathauwer, De Moor and Vandewalle, we consider a $(2, 2)$ transfer function tensor as an element of $\mathbb{C}^{m \times n \times m \times n}$ where m and n denote the maximum index values for the i, k and j, l co-ordinates respectively. According to the definition of Tensor Unfolding on page 1255 of Lathauwer, De Moor and Vandewalle,¹¹ we have the following 1-unfolding of the tensor $\mathcal{I}2$ (see Figure 5 for a pictorial explanation of the 1-unfolding with a couple of (i, j, k, l) terms shown):

$$A_{(1)}(s) = \begin{bmatrix} g_0 & g_1 & g_2 & | & g_1 & g_3 & g_4 & | & g_2 & g_4 & g_5 \\ g_1 & g_3 & g_4 & | & g_0 & g_1 & g_2 & | & g_1 & g_3 & g_4 \\ g_2 & g_4 & g_5 & | & g_1 & g_3 & g_4 & | & g_0 & g_1 & g_2 \\ & & & & g_1 & g_0 & g_1 & | & g_3 & g_1 & g_3 & | & g_4 & g_2 & g_4 \\ & & & & g_3 & g_1 & g_3 & | & g_1 & g_0 & g_1 & | & g_3 & g_1 & g_3 \\ & & & & g_4 & g_2 & g_4 & | & g_3 & g_1 & g_3 & | & g_1 & g_0 & g_1 \\ & & & & & & & & g_2 & g_1 & g_0 & | & g_4 & g_3 & g_1 & | & g_5 & g_4 & g_2 \\ & & & & & & & & g_4 & g_3 & g_1 & | & g_2 & g_1 & g_0 & | & g_4 & g_3 & g_1 \\ & & & & & & & & g_5 & g_4 & g_2 & | & g_4 & g_3 & g_1 & | & g_2 & g_1 & g_0 \end{bmatrix} \quad (13)$$

The 1-tensor unfolding is a $\mathbb{C}^{3 \times 9}$ matrix and hence can be decomposed into the SVD:

$$A_{(1)} = U^{(1)} \cdot \Sigma^{(1)} \cdot V^{(1)H} \quad (14)$$

The local spatial invariance of the system leads to the following result:

LEMMA 3.1. *Suppose that the (2,2) transformation tensor $G(s) \in \mathbb{C}^{m \times n \times m \times n}$ arises from a locally spatially invariant system. Then the unitary matrix $U^{(1)} = U^{(3)}$ and $U^{(2)} = U^{(4)}$. Furthermore, if $m = n$, then $U^{(1)} = U^{(2)} = U^{(3)} = U^{(4)}$.*

Proof. The proof is elementary and is based on the observation that a spatially invariant system satisfies the tensor symmetry condition (9). \square

This lemma considerably simplifies the computation of the core singular value tensor whose 1 unfolding is given by¹¹:

$$S_{(1)} = U^{(1)H} \cdot A_{(1)} \cdot \left(U^{(2)} \otimes U^{(3)} \otimes U^{(4)} \right) \quad (15)$$

3.1. Numerical Results

Similar to Chou et al.,⁷ and Kantor, Krishnaprasad,⁸ we first chose a 3×3 array of controller units, so that the nature of the tensor SVD becomes clear. Following Kantor and Krishnaprasad, the transfer functions at a specific frequency $s = 2\pi f i$, which we simply denote $g_r(s)$; $r = 0, \dots, 5$ in (12) were chosen as follows:

$$\begin{aligned} g_0(s) &= 1 & g_3(s) &= \frac{1}{2d^2} e^{\frac{2\sqrt{2}\pi i}{3}} \\ g_1(s) &= \frac{1}{d^2} e^{\frac{2\pi i}{3}} & g_4(s) &= \frac{1}{5d^2} e^{\frac{2\sqrt{5}\pi i}{3}} \\ g_2(s) &= \frac{1}{4d^2} e^{\frac{4\pi i}{3}} & g_5(s) &= \frac{1}{8d^2} e^{\frac{4\sqrt{2}\pi i}{3}}, \end{aligned} \quad (16)$$

where $d = 2$. This choice led to the following results (only two significant digits are shown for simplicity):

$$U^{(1)} = \begin{bmatrix} -0.5 & 0.7 & 0.5 \\ 0.8 + 0.1i & 0 & 0.7 + 0.1i \\ -0.5 & -0.7 & 0.5 \end{bmatrix}$$

It was observed that $U^{(2)} = U^{(3)} = U^{(1)}$ as expected.

$$S_{(1)} = \left[\begin{array}{ccc|ccc|ccc} 1.1 - 0.4i & 0 & 0.2i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.2 - 0.2i & 0 & 0.2i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.2 + 0.1i & 0 & -0.1 + 0.2i \\ \hline 0 & 1.2 - 0.3i & 0 & 0 & 0 & 0 & 0 & 0.1i & 0 \\ 0 & 0 & 0 & 0 & 1.1 + 0.1i & 0 & 0 & 0 & 0 \\ 0 & 0.1i & 0 & 0 & 0 & 0 & 0 & 0.8 + 0.4i & 0 \\ \hline 0.2i & 0 & 1.2 & 0 & 0 & 0 & 0 & 0 & 0.2i \\ 0 & 0 & 0 & 0.2i & 0 & 0.8 + 0.5i & 0 & 0 & 0 \\ 0 & 0 & -0.1 + 0.1i & 0 & 0 & 0 & -0.1 + 0.2i & 0 & 0.3 + 0.5i \end{array} \right]$$

The other unfoldings $S_{(2)}$ and $S_{(3)}$ are identical to $S_{(1)}$ as expected. The core singular value tensor \mathcal{S} is then constructed from the unfolding $S_{(1)}$. We note here that Chou, Flamm and Guthart⁷ arrived at a similar conclusion that one cannot diagonalize the transfer function tensor in general, through ad hoc means. The following properties were verified for \mathcal{S} :

$$\left. \begin{aligned} &\langle \mathcal{S}(r, :, :, :), \mathcal{S}(s, :, :, :)\rangle = 0 \\ &\langle \mathcal{S}(:, r, :, :), \mathcal{S}(:, s, :, :)\rangle = 0 \\ &\langle \mathcal{S}(:, :, r, :), \mathcal{S}(:, :, s, :)\rangle = 0 \\ &\langle \mathcal{S}(:, :, :, r), \mathcal{S}(:, :, :, s)\rangle = 0 \end{aligned} \right\} \text{ when } r \neq s; \quad r, s \in \{1, 2, 3\},$$

where the product shown is the standard inner-product for tensors, for example:

$$\langle \mathcal{S}(:, r, :, :), \mathcal{S}(:, s, :, :)\rangle = \sum_i \sum_k \sum_l \mathcal{S}(i, r, k, l) \mathcal{S}(i, s, k, l).$$

Furthermore, it was also verified that the n -mode singular values satisfy:

$$\begin{aligned}\|\mathcal{S}(1, :, :, :)\|_F &= [\langle \mathcal{S}(1, :, :, :), \mathcal{S}(1, :, :, :)\rangle]^{\frac{1}{2}} = 1.25 \\ \|\mathcal{S}(2, :, :, :)\|_F &= 1.23 \\ \|\mathcal{S}(3, :, :, :)\|_F &= 1.23 \\ \|\mathcal{S}(4, :, :, :)\|_F &= 1.22,\end{aligned}$$

which confirms the inequalities $\|\mathcal{S}(1, :, :, :)\|_F \geq \|\mathcal{S}(2, :, :, :)\|_F \geq \|\mathcal{S}(3, :, :, :)\|_F \geq \|\mathcal{S}(4, :, :, :)\|_F$ predicted by Theorem 2 of Lathauwer, De Moor and Vandewalle.¹¹

On the other hand, we indexed the 3×3 array as in Figure 4(a), and obtained the matrix transfer function given by (11) with the individual transfer functions again chosen according to (16). On doing a SVD of this matrix, the following singular values were obtained:

$$\Sigma = [1.25, 1.23, 1.23, 1.22, 1.21, 1.10, 0.92, 0.92, 0.65]^T.$$

Observe that the first four singular values are exactly the same as the Frobenius norms $\|\mathcal{S}(r, :, :, :)\|_F$; $r = 1, 2, 3, 4$.

The significance of the TSVD for distributed control is that one can ascribe a physical meaning to the unitary matrices $U^{(r)}$; $r = 1, 2, 3, 4$. They correspond to transforms along the i, j, k and l directions of the tensor G_{ij}^{kl} . One cannot ascribe such meaning to the unitary matrices obtained after a SVD of the matrix transfer function. Notice that as the SVD and TSVD computations can be performed off-line to determine the transformations of the input and the output, speed is not a concern here. But once the exact unitary matrices have been computed, one would like to obtain a Recursive, Orthogonal Transform (ROT)^{6,9} as close as possible to the exact unitary matrices, because ROT's only utilize local data. At this point we realize that there is another important reason to use the TSVD. As ROT's only approximately diagonalize the system, it is possible that the system after transformation has significant terms close to the main diagonal.⁸ This can be problematic because neighboring terms index-wise need not correspond to spatial neighbors, when a single index is used. On the other hand, with the tensor description, spatial neighbors correspond uniquely to index neighbors! To find the ROT that is close to a given unitary matrix U , one needs to take a different approach from the one taken by Kantor and Krishnaprasad, who (a) did not compute U before hand and (b) set up a minimization problem for the sum of the square of the absolute values of the diagonal terms. This approach is related to the Jacobi method, but is not applicable to Tensor SVD. We propose to compute the matrices $U^{(n)}$ by (14), and then compute a ROTs $O^{(n)}$ that are "close" to $U^{(n)}$ in some sense. That ROT's are dense in the space of orthogonal matrices was proved by Kantor and Krishnaprasad. We will deal with the computational aspect of $O^{(n)}$ in a future publication.

4. CONCLUSION

In this paper, we have examined the use of the Tensor Singular Value Decomposition for the distributed control of locally spatially invariant linear systems. Collecting together ideas from Chou et al.⁷ who first proposed the idea of diagonalization using wavelet transforms for distributed control; and Kantor, Krishnaprasad,^{6,8,9} who proposed the idea of using Recursive Orthogonal Transforms (ROT's) to achieve approximate diagonalization of the transfer function matrix for a linear array, we have proposed the use of Tensor Singular Value Decomposition along with ROT's, for two or three dimensional arrays of collocated actuators and sensors. This approach uses a multiple indexing scheme that is natural to the layout of the actuators and sensors, and we showed that it is indeed superior to a simple single indexing scheme.

REFERENCES

1. J. Kim, B.-S. Im, and J.-G. Lee, "Active noise suppression of smart panels including piezoelectric devices and absorbing materials," in *Proceedings of SPIE Smart Structures and Materials 2000: Mathematics and Control in Smart Structures*, pp. 94–100, Mar. 2000.

2. B. Balachandran and M. X. Zhao, "Actuator nonlinearities in interior acoustics control," in *Proceedings of SPIE Smart Structures and Materials 2000: Mathematics and Control in Smart Structures*, pp. 101–109, Mar. 2000.
3. C. A. Savran, M. J. Atalla, and S. R. Hall, "Broadband active structural-acoustic control of a fuselage test-bed using collocated piezoelectric sensors and actuators," in *Proceedings of SPIE Smart Structures and Materials 2000: Mathematics and Control in Smart Structures*, pp. 136–147, Mar. 2000.
4. C. Joshi, A. Pappo, D. Upham, and J. Preble, "Progress in the development of SRF cavity tuners based on magnetic smart materials," in *Proceedings of the Particle Accelerator Conference*, 2001. Chicago, June 16–20.
5. B. Bamieh, F. Paganini, and M. Dahleh, "Distributed control of spatially invariant systems," *IEEE Trans. Automatic Control* **47**, pp. 1091–1107, July 2002.
6. G. Kantor, *Approximate Matrix Diagonalization for use in distributed control networks*. PhD thesis, University of Maryland, College Park, MD, 1999. Available at <http://techreports.isr.umd.edu/ARCHIVE/>, Year 1999, Ph.D. thesis section.
7. K. Chou, D. Flamm, G. Guthart, and R. Ueberschaer, "Multiscale approach to the control of smart structures," in *Smart Structures and Materials 1996: Industrial and Commercial Applications of Smart Structures Technologies*, C. R. Crowe, ed., **2721**, pp. 94–105, May 1996.
8. G. Kantor and P. S. Krishnaprasad, "Efficient implementation of controllers for large scale linear systems via wavelet packet transforms," in *Proceedings of the 32nd Conference on Information Sciences and Systems*, pp. 52–56, March 1998. Also published as a technical report, TR98-8, Institute for Systems Research, University of Maryland, College Park.
9. G. Kantor and P. S. Krishnaprasad, "An application of Lie Groups in distributed control," *Systems and Control Letters* **43**, pp. 43–52, 2001.
10. G. Kron, *Tensor Analysis of Networks*, John Wiley and Sons, Inc., 1939.
11. L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal of Matrix Analysis and Applications* **21**(4), pp. 1253–1278, 2000.
12. M. I. Frecker, "Recent advances in optimization of smart structures and actuators," *Journal of Intelligent Material Systems and Structures* **14**, pp. 207–216, 2003.
13. P. A. Roebuck and S. Barnett, "A survey of toeplitz and related matrices," *Int. J. Systems Sci.* **9**(8), pp. 921–934, 1978.
14. A. Cantoni and P. Butler, "Properties of the eigenvectors of persymmetric matrices with applications to communications theory," *IEEE Transactions on Communications* **24**(8), pp. 804–809, 1976.
15. P. Davis, *Circulant Matrices*, Chelsea Pub. Co., 1994.
16. D. Kalman and J. E. White, "Polynomial equations and circulant matrices," *American Mathematical Monthly*, pp. 821–840, November 2001.
17. E. Kofidis and P. Regalia, "Tensor approximation and signal processing applications," in *Structured Matrices in Mathematics, Computer Science and Engineering I*, V. Olshevsky, ed., **280**, pp. 103–133, 2001.
18. G. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, 1996.