Multidimensional Systems and Signal Processing, 16, 7–47, 2005

State Space Realization of a Three-dimensional Image Set with Application to Noise Reduction of Fluorescent Microscopy Images of Cells

RAIMUND J. OBER* ober@utdallas.edu
Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, USA; Center for Immunology, University of Texas Southwestern Medical Center, Dallas, TX, USA

XUMING LAI
Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, USA; Center for Immunology, University of Texas Southwestern Medical Center, Dallas, TX, USA

ZHIPING LIN
School of Electrical and Electronic Engineering, Nanyang Technological University, Block S2, Nanyang Avenue, Singapore 639798, Republic of Singapore

E. SALLY WARD
Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, USA; Center for Immunology, University of Texas Southwestern Medical Center, Dallas, TX, USA

Received July 9, 2003; Revised April 20, 2003; Accepted June 11, 2004; First online version published in December 2004

Abstract. A method is presented to calculate state space realizations of a three-dimensional image set. It is based on interpreting the image set as the impulse response of a 3D separable system. As an application it is shown how this method, combined with approximation steps, including balanced model reduction, can be used to suppress noise in three-dimensional image sets. The approach was motivated by a practical problem in the analysis of three-dimensional fluorescent microscopy image data of fluorescently labelled cells. The method is illustrated by an analysis of simulated data and experimental data. The proposed approach can also be applied to a two-dimensional image in a straightforward way.

Key Words: multi-dimensional state space realization, separable *n*-D system, image processing, noise suppression, balanced realization, fluorescent microscopy

1. Introduction

Fluorescence microscopy is widely used as a technique to reveal the location of proteins in cells. Modern fluorescent labeling techniques allow specific cellular proteins to be tagged by fluorescent markers. Imaging the cells with an epi-fluorescent microscope reveals the location of the markers and therefore the location of the specific proteins in the cells. To obtain information for the whole cell two-dimensional images (frames) are taken at different focal planes. The resulting set of images therefore forms a three-dimensional image set of the specimen.

*Corresponding Author

One of the problems of this methodology is that the signal level is typically very low even when highly sensitive detectors are used [1]. In addition, the signal, i.e. the photons emitted by the fluorescent tags, is itself a random process. There are also various noise sources in the system ranging from scattered photons to readout noise in the CCD camera [1]. This means that the acquired images have low signal-to-noise ratios. Due to the optical properties of the microscope significant blurring can occur because of the out of focus components of the sample [2], [3]. This blurring can be removed by the use of deconvolution algorithms. However, the presence of noise in the image sets is a notorious problem in the application of deconvolution algorithms (see e.g., [4], [5], [6]. Therefore noise reduction in an image set is of significant practical importance.

Many advanced signal processing techniques call for the use of state space models. Therefore the question arises whether it is possible to also obtain a state space representation for a three-dimensional image set as described above. In [7], [8], [9], [10], [11] state space realizations are used for multi-dimensional filter design. Here we will establish that it is indeed possible to represent a three-dimensional image set as the impulse response of a three-dimensional (3D) separable system.

Balanced model reduction is an effective method to reduce the order of the system represented by a balanced state space realization. It has also several desirable properties such as an error bound and preservation of the stability of the original system (see e.g. [12], [13], [14] for a review of some of the basic properties of balanced realizations). The singular value decomposition of a data matrix is used in the design of a two-dimensional filter [15] and is also employed in noise reduction for speech enhancement (see [16] and references therein). In [17], [18] both singular value decomposition and state space realizations are used in the design of a two-dimensional separable-denominator digital filter. We show how an approximation of a three-dimensional image set based on singular value decompositions of the corresponding data matrices, combined with a balanced model reduction of the one-dimensional components of the image set, can lead to a significant reduction in the noise components of the image set.

The paper is organized as follows. In Section 2, we introduce some notation. In Section 3, we present an algorithm to decompose a three-dimensional image set and express it as the product of three one-dimensional vector-/matrix-valued components. Then we discuss the use of a balanced realization algorithm to produce state space realizations of the one-dimensional components of the three-dimensional image set. In Section 4, we evaluate the algorithms with both simulated data and experimental data. We draw conclusions in Section 5.

2. Notation

Shift operation. Let $L = [L_{t_1 t_2}]_{\substack{1 \leq t_1 \leq T_1 \\ 1 \leq t_2 \leq T_2}}$ be a block matrix, where $L_{t_1 t_2} \in \mathbb{R}^{p \times q}$ for some $p, q \geq 1$ for all $1 \leq t_1 \leq T_1$, $1 \leq t_2 \leq T_2$. Then let $L^{\uparrow}$ be the $(T_1 - 1)p \times T_2 q$ matrix given by dropping the first block row of L, i.e. $L^{\uparrow} = [\hat{L}_{t_1 t_2}]_{\substack{1 \le t_1 \le T_1 - 1 \\ 1 \le t_2 \le T_2}}$ with $\hat{L}_{t_1 t_2} = L_{(t_1+1)t_2}$ for all $1 \le t_1 \le T_1 - 1$, $1 \le t_2 \le T_2$, and let $L^{\downarrow}$ be the $(T_1 - 1)p \times T_2 q$ matrix given by dropping the last block row of L, $L^{\downarrow} = [\hat{L}_{t_1 t_2}]_{\substack{1 \le t_1 \le T_1 - 1 \\ 1 \le t_2 \le T_2}}$ with $\hat{L}_{t_1 t_2} = L_{t_1 t_2}$ for all $1 \le t_1 \le T_1 - 1$, $1 \le t_2 \le T_2$, and let L^{F_r} be the $p \times T_2 q$ matrix given by the first block row of L, i.e. $L^{F_r} = [\hat{L}_{1 t_2}]_{1 \le t_2 \le T_2}$ with $\hat{L}_{1 t_2} = L_{1 t_2}$ for all $1 \le t_2 \le T_2$, and let L^{L_r} be the $p \times T_2 q$ matrix given by the last block row of L, i.e. $L^{L_r} = [\hat{L}_{1 t_2}]_{1 \le t_2 \le T_2}$ with $\hat{L}_{1 t_2} = L_{T_1 t_2}$ for all $1 \le t_2 \le T_2$. Then we have that

$$L = \begin{bmatrix} L^{F_r} \\ L^{\uparrow} \end{bmatrix} = \begin{bmatrix} L^{\downarrow} \\ L^{L_r} \end{bmatrix}.$$

Let $\overleftarrow{L}$ be the $T_1 p \times (T_2 - 1)q$ matrix given by dropping the first block column of L, i.e. $\overleftarrow{L} = [\hat{L}_{t_1 t_2}]_{\substack{1 \le t_1 \le T_1 \\ 1 \le t_2 \le T_2 - 1}}$ with $\hat{L}_{t_1 t_2} = L_{t_1 (t_2+1)}$ for all $1 \le t_1 \le T_1$, $1 \le t_2 \le T_2 - 1$, and let $\overrightarrow{L}$ be the $T_1 p \times (T_2 - 1)q$ matrix given by dropping the last block column of L, i.e. $\overrightarrow{L} = [\hat{L}_{t_1 t_2}]_{\substack{1 \le t_1 \le T_1 \\ 1 \le t_2 \le T_2 - 1}}$ with $\hat{L}_{t_1 t_2} = L_{t_1 t_2}$ for all $1 \le t_1 \le T_1$, $1 \le t_2 \le T_2 - 1$, and let L^{F_c} be the $T_1 p \times q$ matrix given by the first block column of L, i.e. $L^{F_c} = [\hat{L}_{t_1 1}]_{1 \le t_1 \le T_1}$ with $\hat{L}_{t_1 1} = L_{t_1 1}$ for all $1 \le t_1 \le T_1$, and let L^{L_c} be the $T_1 p \times q$ matrix given by the last block column of L, i.e. $L^{L_c} = [\hat{L}_{t_1 1}]_{1 \le t_1 \le T_1}$ with $\hat{L}_{t_1 1} = L_{t_1 T_2}$ for all $1 \le t_1 \le T_1$. Then we have that $L = [L^{F_c}\ \overleftarrow{L}] = [\overrightarrow{L}\ L^{L_c}]$.

Root-mean-square error. For positive integers N_1, N_2, N_3 and two three-dimensional arrays $P(k_1, k_2, k_3)$, $\hat{P}(k_1, k_2, k_3) \in \mathbb{R}$, $k_i = 1, 2, \ldots, N_i$, $i = 1, 2, 3$, the root-mean-square error (RMSE) is given by

$$\epsilon = \left(\frac{1}{N_1 N_2 N_3} \sum_{k_1=1}^{N_1} \sum_{k_2=1}^{N_2} \sum_{k_3=1}^{N_3} (P(k_1, k_2, k_3) - \hat{P}(k_1, k_2, k_3))^2 \right)^{1/2}.$$

Identity matrix. I_d stands for the identity matrix of size $d \times d$. If the size is clear from the context the subscript d is often dropped.

Diagonal matrix. If $\mu_1, \ldots, \mu_K$ are scalars or matrices we denote by $diag(\mu_1, \ldots, \mu_K)$ the (block-)diagonal matrix with diagonal entries $\mu_1, \ldots, \mu_K$.

Singular value decomposition. Let H be an $M \times N$ matrix. A singular value decomposition of H is defined by the factorization $H = U \Sigma V$, where U, V are matrices consisting of orthogonal columns and orthogonal rows, respectively, i.e. $U^* U = I$ and $VV^* = I$, and their sizes are $M \times K$ and $K \times N$, respectively, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_K) > 0$. Note that σ_i, $i = 1, 2, \ldots, K$, are the singular values.

Frobenius norm. The Frobenius norm of a matrix A is defined by $||A||_F = \sqrt{Tr(A^* A)}$, where the superscript $*$ denotes the complex conjugate transpose and Tr(.) denotes the trace of the matrix.

3. State Space Realizations of a Three-dimensional Image Set

The basis of the approach taken here is given by the following realization result, in which it is shown that a finite multi-dimensional data set can be decomposed and expressed as the product of the impulse responses of multiple one-dimensional multi-input multi-output systems. A similar approach was used in [17], [18] to calculate the realizations of two-dimensional data sets. In contrast to our finite multi-dimensional data situation the theoretical results in [17], [18] primarily address the realizations of infinite two-dimensional data sets.

Theorem 1. *For positive integers* $N_1, \ldots, N_D$ *let* $P(k_1, k_2, \ldots, k_D) \in \mathbb{R}$, $k_i = 1, \ldots, N_i$, $i = 1, \ldots, D$, *be a finite D-dimensional array. Then there exist system matrices* (A_i, B_i, C_i), $i = 1, \ldots, D$, *such that*

$$P(k_1, k_2, \ldots, k_D) = P_1(k_1)P_2(k_2)\cdots P_D(k_D),$$

where $P_i(k_i) := C_i A_i^{k_i-1} B_i$ *with* $C_i \in \mathbb{R}^{p_i \times n_i}$, $A_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times m_i}$, $k_i = 1, \ldots, N_i$, $i = 1, \ldots, D$.

Proof: For positive integers $N_1, \ldots, N_r$ let $Q(k_1, k_2, \ldots, k_r)$, $k_i = 1, \ldots, N_i$, $i = 1, \ldots, r$, be an array with block entries. We now arrange the block entries of Q in the following way as

$$Q_r := \begin{bmatrix} Q(1,1,\ldots,1,1) & Q(1,1,\ldots,1,2) & \cdots & Q(1,1,1,\ldots,1,N_r) & Q(1,1,\ldots,1,2,1) & \cdots \\ Q(2,1,\ldots,1,1) & Q(2,1,\ldots,1,2) & \cdots & Q(2,1,1,\ldots,1,N_r) & Q(2,1,\ldots,1,2,1) & \cdots \\ Q(3,1,\ldots,1,1) & Q(3,1,\ldots,1,2) & \cdots & Q(3,1,1,\ldots,1,N_r) & Q(3,1,\ldots,1,2,1) & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \\ Q(N_1,1,\ldots,1,1) & Q(N_1,1,\ldots,1,2) & \cdots & Q(N_1,1,1,\ldots,1,N_r) & Q(N_1,1,\ldots,1,2,1) & \cdots \end{bmatrix.$$

$$\begin{matrix} \cdots & Q(1,1,\ldots,1,2,N_r) & \cdots & Q(1,N_2,\ldots,N_{r-2},N_{r-1},N_r) \\ \cdots & Q(2,1,\ldots,1,2,N_r) & \cdots & Q(2,N_2,\ldots,N_{r-2},N_{r-1},N_r) \\ \cdots & Q(3,1,\ldots,1,2,N_r) & \cdots & Q(3,N_2,\ldots,N_{r-2},N_{r-1},N_r) \\ \vdots & & \vdots & \\ \cdots & Q(N_1,1,\ldots,1,2,N_r) & \cdots & Q(N_1,N_2,\ldots,N_{r-2},N_{r-1},N_r) \end{matrix}\Bigg]. \quad (1)$$

Note that the arrangement of the entries is such that all indices from the second onwards are identical in all rows, whereas the first index increases from row to row.

Let $Q_r = LR$ be a minimum rank decomposition of Q_r, i.e. $\operatorname{rank}(Q_r) = \operatorname{rank}(L) = \operatorname{rank}(R)$. The left factor $L = \begin{bmatrix} L(1) \\ L(2) \\ \vdots \\ L(N_1) \end{bmatrix}$ is a finite block column vector with N_1 entries. Therefore it has a realization (A, B, C), i.e. $L(k_1) = CA^{k_1-1}B$, $k_1 = 1, \ldots, N_1$ [19]. The right factor R is a block row vector, i.e. $R = [R(1),\ R(2), \ldots, R(N_2N_3 \cdots N_r)]$.

We apply the just described approach to P to obtain the factors L and R. Denote the left factor L by P_1 and its realization by (A_1, B_1, C_1), i.e. $P_1(k_1) = C_1A_1^{k_1-1}B_1$, $k_1 = 1, \ldots, N_1$. The right factor R can in a natural way be associated with a block array P_{1R} with indices $k_2, \ldots, k_D$, i.e.

$$P_{1R} := \begin{bmatrix} P_{1R}(1,1,\ldots,1) & P_{1R}(1,1,\ldots,2) & \cdots & P_{1R}(1,N_3,\ldots,N_D) \\ P_{1R}(2,1,\ldots,1) & P_{1R}(2,1,\ldots,2) & \cdots & P_{1R}(2,N_3,\ldots,N_D) \\ P_{1R}(3,1,\ldots,1) & P_{1R}(3,1,\ldots,2) & \cdots & P_{1R}(3,N_3,\ldots,N_D) \\ \vdots & \vdots & \vdots & \vdots \\ P_{1R}(N_2,1,\ldots,1) & P_{1R}(N_2,1,\ldots,2) & \cdots & P_{1R}(N_2,N_3,\ldots,N_D) \end{bmatrix} \tag{2}$$

$$:= \begin{bmatrix} R(1) & R(2) & R(3) & \cdots & R(N_3N_4\cdots N_D) \\ R(N_3N_4\cdots N_D+1) & R(N_3N_4\cdots N_D+2) & R(N_3N_4\cdots N_D+3) & \cdots & R(2N_3N_4\cdots N_D) \\ R(2N_3N_4\cdots N_D+1) & R(2N_3N_4\cdots N_D+2) & R(2N_3N_4\cdots N_D+3) & \cdots & R(3N_3N_4\cdots N_D) \\ \vdots & \vdots & \vdots & & \vdots \\ R((N_2-1)N_3N_4\cdots N_D+1) & R((N_2-1)N_3N_4\cdots N_D+2) & R((N_2-1)N_3N_4\cdots N_D+3) & \cdots & R(N_2N_3N_4\cdots N_D) \end{bmatrix}.$$

We therefore have obtained a decomposition of P as $P(k_1, k_2, \ldots, k_D) = P_1(k_1)P_{1R}$ $(k_2, k_3, \ldots, k_D)$, $k_i = 1, \ldots, N_i$, $i = 1, \ldots, D$. Now the above decomposition can be applied to P_{1R}. If we continue recursively we have

$$P(k_1, k_2, \ldots, k_D) = P_1(k_1)P_2(k_2) \cdots P_D(k_D),$$

where $P_i(k_i) := C_iA_i^{k_i-1}B_i$, $k_i = 1, \ldots, N_i$, $i = 1, \ldots, D$.

Example 1. To illustrate the main ideas in Theorem 1 and to explain how a product of three vector-/matrix-valued sequences can represent an array that cannot be represented by three scalar-valued sequences, we will show how to decompose the elements of a three-dimensional array $P(k_1, k_2, k_3)$, $k_i = 1, 2, 3$, i.e.

$$P(k_1, 1, k_3) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad P(k_1, 2, k_3) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$
$$P(k_1, 3, k_3) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad k_1, k_3 = 1, 2, 3,$$

into three vector-/matrix-valued sequences. We use such a small array to simplify the calculation.

The first step in the construction introduced in Theorem 1 is to rearrange the elements of P to form a two-dimensional array Q_3 (Equation (1)), i.e.

$$Q_3 = \begin{bmatrix} P(1,1,1) & P(1,1,2) & P(1,1,3) & P(1,2,1) & P(1,2,2) & \cdots & P(1,3,3) \\ P(2,1,1) & P(2,1,2) & P(2,1,3) & P(2,2,1) & P(2,2,2) & \cdots & P(2,3,3) \\ P(3,1,1) & P(3,1,2) & P(3,1,3) & P(3,2,1) & P(3,2,2) & \cdots & P(3,3,3) \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Then a minimum rank decomposition of Q_3 is performed. The results of the decomposition depend on the method of the decomposition. Here, a very simple decomposition can be obtained as

$$Q_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Next, we assign $P_1 := \begin{bmatrix} P_1(1) \\ \hline P_1(2) \\ \hline P_1(3) \end{bmatrix} := \begin{bmatrix} 0 & 1 \\ \hline 1 & 0 \\ \hline 0 & 1 \end{bmatrix}$,

where the block column vector P_1 is the first one-dimensional components of P, and $R := [R(1)|R(2)|\cdots|R(9)] := \left[\begin{array}{c|c|c|c|c|c|c|c|c} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$. We then rearrange the elements in R to form a block array P_{1R}, which is naturally associated with the indices k_2 and k_3, based on Equation (2), i.e.

$$P_{1R} := \left[\begin{array}{c|c|c} \text{P1R (1,1)} & \text{P1R (1,2)} & \text{P1R (1,3)} \\ \hline \text{P1R (2,1)} & \text{P1R (2,2)} & \text{P1R (2,3)} \\ \hline \text{P1R (3,1)} & \text{P1R (3,2)} & \text{P1R (3,3)} \end{array}\right] := \left[\begin{array}{c|c|c} \text{R(1)} & \text{R(2)} & \text{R(3)} \\ \hline \text{R(4)} & \text{R(5)} & \text{R(6)} \\ \hline \text{R(7)} & \text{R(8)} & \text{R(9)} \end{array}\right]$$

$$= \left[\begin{array}{c|c|c} 0 & 1 & 0 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 1 \\ 0 & 1 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}\right].$$

Again, we find a simple minimum rank decomposition of P_{1R}, i.e.,

$$P_{1R} = \left[\begin{array}{cc} 0 & 1 \\ 0 & 0 \\ \hline 1 & 0 \\ 0 & 1 \\ \hline 0 & 1 \\ 0 & 0 \end{array}\right] \left[\begin{array}{c|c|c} 1 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right],$$

and assign $P_2 := \left[\begin{array}{c} P_2(1) \\ \hline P_2(2) \\ \hline P_2(3) \end{array}\right] := \left[\begin{array}{cc} 0 & 1 \\ 0 & 0 \\ \hline 1 & 0 \\ 0 & 1 \\ \hline 0 & 1 \\ 0 & 0 \end{array}\right]$, and $P_3 := [P_3(1)|\ P_3(2)|\ P_3(3)]$ $:= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, where the block column vector P_2 and the block row vector P_3 are the second and third components of the array P. In this case, we see that P_1 and P_3 are vector-valued sequences, while P_2 is a matrix-valued sequence. Thus, we have obtained a decomposition of P such that

$$P(k_1, k_2, k_3) = P_1(k_1)P_2(k_2)P_3(k_3), \quad k_i = 1, 2, 3, \ \ i = 1, 2, 3.$$

The next step involves finding state space realizations of P_1, P_2 and P_3. Such realizations are given by

$$A_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C_1 = [1, \ \ 0, \ \ 0],$$

$$A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and

$$A_3 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad C_3 \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Therefore we have

$$P_1(k_1) = C_1 A_1^{k_1-1} B_1, \quad k_1 = 1, 2, 3,$$
$$P_2(k_2) = C_2 A_2^{k_2-1} B_2, \quad k_2 = 1, 2, 3$$

and

$$P_3(k_3) = C_3 A_3^{k_3-1} B_3, \quad k_3 = 1, 2, 3.$$

It is easy to verify that

$$P(k_1, k_2, k_3) = C_1 A_1^{k_1-1} B_1 C_2 A_2^{k_2-1} B_2 C_3 A_3^{k_3-1} B_3, \quad k_1, k_2, k_3 = 1, 2, 3. \qquad \blacksquare$$

The proof of Theorem 1 gives a high level algorithm as to how to obtain a realization for a finite D-dimensional array of data points. What is missing in this high level algorithm is a particular method to obtain the factorizations of the data matrix and a realization algorithm. While a number of choices exist we will use here approaches based on the singular value decomposition and balanced realizations. The advantage of using these methods is that model reduction and noise reduction can easily be incorporated into the algorithm.

3.1. Decomposition of Three-Dimensional Image Sets

In what follows we present the first part of the algorithm, i.e. the factorization part, in some detail. In the proof of the previous theorem a full rank decomposition $Q_r = LR$ of a rearranged data matrix Q was performed. In our specific implementation we will use a singular value decomposition to carry out this factorization. If it is not required to have a state space realization that completely matches the data, but if noise reduction is desired instead, an approximate factorization can be used that is also based on the singular value decomposition. Singular value decompositions of two-dimensional data matrices are also employed in noise reduction for speech enhancement (see[16] and references therein) and the design of a two-dimensional filter ([15]).

Algorithm 1. *For positive integers* N_1, N_2 *and* N_3 *let* $P(k_1, k_2, k_3)$, $k_i = 1, 2, \ldots, N_i$, $i = 1, 2, 3$, *represent a three-dimensional image set.*

1. *Arrange the entries of* P *in a matrix* Q_3 *as*

$$Q_3 = \left[\begin{matrix} P(1,1,1) & P(1,1,2) & \cdots & P(1,2,1) & P(1,2,2) \\ P(2,1,1) & P(2,1,2) & \cdots & P(2,2,1) & P(2,2,2) \\ P(3,1,1) & P(3,1,2) & \cdots & P(3,2,1) & P(3,2,2) \\ \vdots & \vdots & & \vdots & \vdots \\ P(N_1,1,1) & P(N_1,1,2) & \cdots & P(N_1,2,1) & P(N_1,2,2) \end{matrix}\right.$$

$$\left.\begin{matrix} \cdots & P(1,3,1) & P(1,3,2) & \cdots & P(1,N_2,N_3) \\ \cdots & P(2,3,1) & P(2,3,2) & \cdots & P(2,N_2,N_3) \\ \cdots & P(3,3,1) & P(3,3,2) & \cdots & P(3,N_2,N_3) \\ \vdots & \vdots & & \vdots & \\ \cdots & P(N_1,3,1) & P(N_1,3,2) & \cdots & P(N_1,N_2,N_3) \end{matrix}\right].$$

2. *Decompose* Q_3 *via the singular value decomposition as* $Q_3 = U_1\Sigma_1^{1/2}\Sigma_1^{1/2}V_1$.

3. *Partition* $\Sigma_1 = \mathrm{diag}(\hat{\Sigma}_1, \hat{\hat{\Sigma}}_1)$, $U_1 = [\hat{U}_1, \hat{\hat{U}}_1]$, *and* $V_1 = \begin{bmatrix} \hat{V}_1 \\ \hat{\hat{V}}_1 \end{bmatrix}$ *conformally, where*

$\hat{\Sigma}_1 \in \mathbb{R}^{l_1 \times l_1}$, $\hat{\hat{\Sigma}}_1 \in \mathbb{R}^{r_1 \times r_1}$, $\hat{U}_1 \in \mathbb{R}^{N_1 \times l_1}$, $\hat{\hat{U}}_1 \in \mathbb{R}^{N_1 \times r_1}$, $\hat{V}_1 \in \mathbb{R}^{l_1 \times N_2 N_3}$, $\hat{\hat{V}}_1 \in \mathbb{R}^{r_1 \times N_2 N_3}$ *and* l_1, r_1 *are positive integers.*

Let $L_3^{r_1} := \begin{bmatrix} L_3^{r_1}(1) \\ L_3^{r_1}(2) \\ L_3^{r_1}(3) \\ \vdots \\ L_3^{r_1}(N_1) \end{bmatrix} := \hat{U}_1\hat{\Sigma}_1^{1/2}$, and $R_3^{r_1} := [R_3^{r_1}(1), \ldots, R_3^{r_1}(N_2N_3)] : \; = \hat{\Sigma}_1^{1/2}\hat{V}_1$,

where $L_3^{r_1}(k_1) \in \mathbb{R}^{1 \times l_1}, k_1 = 1, \ldots, N_1, R_3^{r_1}(i) \in \mathbb{R}^{l_1 \times 1}$, $i = 1, \ldots, N_2N_3$, *and* r_1 *denotes the number of singular values (including multiplicities) that are dropped in the approximation, i.e. the number of diagonal entries of* $\hat{\hat{\Sigma}}_1$. *Then* $Q_3 \approx \hat{Q}_3^{r_1} := L_3^{r_1}R_3^{r_1}$ *is an approximate factorization. The factorization is exact for* $r_1 = 0$, *i.e.* $Q_3 = \hat{Q}_3^0 = L_3^0 R_3^0$.

Let the block column vector $P_1^{r_1}$ *be* $P_1^{r_1} := \begin{bmatrix} P_1^{r_1}(1) \\ P_1^{r_1}(2) \\ P_1^{r_1}(3) \\ \vdots \\ P_1^{r_1}(N_1) \end{bmatrix} := L_3^{r_1}$, *where* $P_1^{r_1}(k_1) \in \mathbb{R}^{1\times l_1}$, $k_1 = 1, 2, \ldots, N_1$.

4. *Rearrange the elements of* $R_3^{r_1}$ *to form* Q_2 *as*

$$Q_2 := \begin{bmatrix} Q_2(1,1) & Q_2(1,2) & Q_2(1,3) & \cdots & Q_2(1,N_3) \\ Q_2(2,1) & Q_2(2,2) & Q_2(2,3) & \cdots & Q_2(2,N_3) \\ Q_2(3,1) & Q_2(3,2) & Q_2(3,3) & \cdots & Q_2(3,N_3) \\ \vdots & \vdots & \vdots & & \vdots \\ Q_2(N_2,1) & Q_2(N_2,2) & Q_2(N_2,3) & \cdots & Q_2(N_2,N_3) \end{bmatrix}$$

$$:= \begin{bmatrix} R_3^{r_1}(1) & R_3^{r_1}(2) & R_3^{r_1}(3) & \cdots & R_3^{r_1}(N_3) \\ R_3^{r_1}(N_3+1) & R_3^{r_1}(N_3+2) & R_3^{r_1}(N_3+3) & \cdots & R_3^{r_1}(2N_3) \\ R_3^{r_1}(2N_3+1) & R_3^{r_1}(2N_3+2) & R_3^{r_1}(2N_3+3) & \cdots & R_3^{r_1}(3N_3) \\ \vdots & \vdots & \vdots & & \vdots \\ R_3^{r_1}((N_2-1)N_3+1) & R_3^{r_1}((N_2-1)N_3+2) & R_3^{r_1}((N_2-1)N_3+3) & \cdots & R_3^{r_1}(N_2N_3) \end{bmatrix}.$$

5. *Decompose* Q_2 *via the singular value decomposition as* $Q_2 = U_2\Sigma_2^{1/2}\Sigma_2^{1/2}V_2$.

6. *Partition* $\Sigma_2 = diag(\hat{\Sigma}_2, \hat{\hat{\Sigma}}_2)$, $U_2 = [\hat{U}_2, \hat{U}_2]$, *and* $V_2 = \begin{bmatrix} \hat{V}_2 \\ \hat{\hat{V}}_2 \end{bmatrix}$ *conformally, where* $\hat{\Sigma}_2 \in \mathbb{R}^{l_2\times l_2}$, $\hat{\hat{\Sigma}}_2 \in \mathbb{R}^{r_2\times r_2}$, $\hat{U}_2 \in \mathbb{R}^{N_2 l_1\times l_2}$ (l_1 *is the number of the retained singular values in step 3*), $\hat{\hat{U}}_2 \in \mathbb{R}^{N_2 l_1\times r_2}$, $\hat{V}_2 \in \mathbb{R}^{l_2\times N_3}$, $\hat{\hat{V}}_2 \in \mathbb{R}^{r_2\times N_3}$ *and* l_2, r_2 *are positive integers.*

Let $L_2^{r_2} := \begin{bmatrix} L_2^{r_2}(1) \\ L_2^{r_2}(2) \\ L_2^{r_2}(3) \\ \vdots \\ L_2^{r_2}(N_2) \end{bmatrix} := \hat{U}_2\hat{\Sigma}_2^{1/2}$, *and* $R_2^{r_2} := [R_2^{r_2}(1), \ldots, R_2^{r_2}(N_3)] := \hat{\Sigma}_2^{1/2}\hat{V}_2$, *where* $L_2^{r_2}(k_2) \in \mathbb{R}^{l_1\times l_2}$, $k_2 = 1, \ldots, N_2$, $R_2^{r_2}(k_3) \in \mathbb{R}^{l_2\times 1}$, $k_3 = 1, \ldots, N_3$, *and* r_2 *denotes the number of singular values (including multiplicities) that are dropped in the approximation, i.e. the number of diagonal entries of* $\hat{\hat{\Sigma}}_2$. *Then* $Q_2 \approx \hat{Q}_2^{r_2} := L_2^{r_2}R_2^{r_2}$ *is an approximate factorization. The factorization is exact for* $r_2 = 0$, *i.e.* $Q_2 = \hat{Q}_2^0 = L_2^0 R_2^0$.

Let the block column vector $P_2^{r_2}$ *be* $P_2^{r_2} := \begin{bmatrix} P_2^{r_2}(1) \\ P_2^{r_2}(2) \\ P_2^{r_2}(3) \\ \vdots \\ P_2^{r_2}(N_2) \end{bmatrix} := L_2^{r_2}$ *and the block row vector* $P_3^{r_2}$ *be* $P_3^{r_2} := [P_3^{r_2}(1),\ P_3^{r_2}(2),\ \ldots,\ P_3^{r_2}(N_3)] := R_2^{r_2}$, *where* $P_2^{r_2}(k_2) \in \mathbb{R}^{l_1 \times l_2}$, $k_2 = 1,2,\ldots,N_2$, *and* $P_3^{r_2}(k_3) \in \mathbb{R}^{l_2 \times 1}$, $k_3 = 1,2,\ldots,N_3$.

In Algorithm 1 an approximate decomposition of the image data points is obtained

$$\begin{aligned} P(k_1,k_2,k_3) &\approx P^{r_1,r_2}(k_1,k_2,k_3) \\ &:= P_1^{r_1}(k_1)P_2^{r_2}(k_2)P_3^{r_2}(k_3), \quad k_i = 1,2,\ldots,N_i,\ i = 1,2,3, \end{aligned}$$

where r_1 and r_2 are the numbers of the dropped singular values in step 3 and step 6. If the image set P is noisy small singular values will be corrupted by noise. Since those small singular values are dropped during the approximation the noise level of the estimate significantly reduces.

If no approximation is carried out, i.e. if $r_1 = r_2 = 0$, we have the exact factorization

$$\begin{aligned} &P(k_1,k_2,k_3) = P^{0,0}(k_1,k_2,k_3) = P_1^0(k_1)P_2^0(k_2)P_3^0(k_3), \\ &k_i = 1,2,\ldots,N_i,\ i = 1,2,3. \end{aligned}$$

It is important to note that the dimensions of the three components $P_1^{r_1}(k_1)$, $P_2^{r_2}(k_2)$ and $P_3^{r_2}(k_3)$ typically differ significantly and depend on the numbers of the retained singular values in step 3 and step 6. In general, $P_1^{r_1}(k_1)$ and $P_3^{r_2}(k_3)$ are vector-valued, whereas $P_2^{r_2}(k_2)$ is matrix-valued.

3.2. Balanced State Space Realizations of Finite One-dimensional Sequences

In the previous subsection the three-dimensional image set was decomposed into vector-/matrix-valued one-dimensional components. In this subsection a state space realization will be calculated for each of these components. Since each component is a finite sequence the existence of a state space realization is guaranteed by fundamental results in realization theory (see e.g. [19]). Among the choices of potential realizations we will use balanced realizations [20], since they are also highly suitable for purposes of model reduction. The particular realization algorithm that we use is a modification of Kung's algorithm (see [21], [22] for a detailed discussion of Kung's algorithm) that is based on singular value decompositions of the Hankel matrices associated with the one-dimensional components. The proposed modifications guarantee that we obtain exact state space realizations for any finite one-dimensional sequences, and approximate state space realizations can be easily obtained from the exact state space realizations.

Algorithm 2. *For positive integer N let $P(i) \in \mathbb{R}^{p\times m}$, $i = 1, 2, \ldots, N$, be a finite one-dimensional sequence.*

1. *Construct the $(N+1)p \times (N+1)m$ Hankel matrix*

$$H = \begin{bmatrix} P(1) & P(2) & \cdots & P(N-1) & P(N) & 0 \\ P(2) & P(3) & \cdots & P(N) & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ P(N) & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 0 \end{bmatrix},$$

where 0 denotes a block of zeros of size $p \times m$.

2. *Let $H = U\Sigma V$ be a singular value decomposition.*

3. *Partition $\Sigma = \mathrm{diag}(\Sigma_1, \Sigma_2)$, $\Sigma_1 \in \mathbb{R}^{n\times n}$, $\Sigma_2 \in \mathbb{R}^{s\times s}$, $U = [U_1, U_2]$, $U_1 \in \mathbb{R}^{(N+1)p\times n}$, $U_2 \in \mathbb{R}^{(N+1)p\times s}$, and $V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$, $V_1 \in \mathbb{R}^{n\times(N+1)m}$, $V_2 \in \mathbb{R}^{s\times(N+1)m}$, conformally. We also allow for the partition in which the second components are empty, i.e. $s = 0$.*

4. *Let $C^s \in \mathbb{R}^{p\times n}$ be the first p rows of $U_1\Sigma_1^{1/2}$.*

5. *Let $B^s \in \mathbb{R}^{n\times m}$ be the first m columns of $\Sigma_1^{1/2}V_1$.*

6. *Let $U_1 = \begin{bmatrix} \bar{U}^1_{11} \\ \vdots \\ \bar{U}^1_{N1} \\ \bar{U}^1_{(N+1)1} \end{bmatrix}$, where the block rows $\bar{U}^1_{t_1 1} \in \mathbb{R}^{p\times n}$ for all $t_1 = 1, \ldots, N+1$,*

and $U_1^{\uparrow} = \begin{bmatrix} \bar{U}^1_{21} \\ \vdots \\ \bar{U}^1_{(N+1)1} \end{bmatrix}$ and $U_1^{\downarrow} = \begin{bmatrix} \bar{U}^1_{11} \\ \vdots \\ \bar{U}^1_{N1} \end{bmatrix}$ (for a formal definition of the shift operation

see Section 2). Then let A^s be the solution of the following least-squares problem

$$\|(U_1\Sigma_1^{1/2})^{\downarrow}A^s - (U_1\Sigma_1^{1/2})^{\uparrow}\|_F = \inf_{X\in\mathbb{R}^{n\times n}} \|(U_1\Sigma_1^{1/2})^{\downarrow}X - (U_1\Sigma_1^{1/2})^{\uparrow}\|_F,$$

and $A^s \in \mathbb{R}^{n\times n}$ can be calculated as

$$A^s = \Sigma_1^{-1/2}U_1^{\downarrow *}U_1^{\uparrow}\Sigma_1^{1/2}.$$

In the following proposition we collect a number of properties concerning the realization obtained from the previous algorithm. While similar results ([11], [21], [23]) have appeared in the literature the specific combination of some of the results presented here appears to be new. In addition, we also discuss the situation concerning an exact state space realization (when $s = 0$) and results on the relationship between the exact state space realization and an approximate realization (when $s \neq 0$).

Proposition 1. *For positive integer N let $P(i) \in \mathbb{R}^{p\times m}$, $i = 1, 2, \ldots, N$, be a one-dimensional sequence. Then*

1. *if$(A^s,\ B^s,\ C^s)$ is a state space realization of P calculated via Algorithm 2, then,*

$$A^s = \Sigma_1^{-1/2} U_1^{\downarrow *} U_1^{\uparrow} \Sigma_1^{1/2}$$

is the solution of the following least-squares problem, i.e.

$$||(U_1\Sigma_1^{1/2})^{\downarrow} A^s - (U_1\Sigma_1^{1/2})^{\uparrow}||_F = \inf_{X\in\mathbb{R}^{n\times n}} ||(U_1\Sigma_1^{1/2})^{\downarrow} X - (U_1\Sigma_1^{1/2})^{\uparrow}||_F.$$

2. *we have that*

$$A^s = \Sigma_1^{-1/2} U_1^{\downarrow *} U_1^{\uparrow} \Sigma_1^{1/2} = \Sigma_1^{1/2} \overleftarrow{V}_1 \overrightarrow{V}_1^{*} \Sigma_1^{-1/2} \tag{3}$$

and A^s is also the least-squares solution such that

$$||A^s(\Sigma_1^{1/2}\overrightarrow{V}_1) - \Sigma_1^{1/2}\overleftarrow{V}_1||_F = \inf_{X\in\mathbb{R}^{n\times n}} ||X(\Sigma_1^{1/2}\overrightarrow{V}_1) - \Sigma_1^{1/2}\overleftarrow{V}_1||_F.$$

3. *if $(A^0,\ B^0,\ C^0)$ is a state space realization of P calculated via Algorithm 2 when $s = 0$ and, based on the sizes of the matrices used in Algorithm 2, we rewrite $U = \begin{bmatrix} \bar{U}_{11} \\ \vdots \\ \bar{U}_{N1} \\ \bar{U}_{(N+1)1} \end{bmatrix}$, where the block rows $\bar{U}_{t_1 1} \in \mathbb{R}^{p\times(n+s)}$* for all $t_1 = 1, \ldots, N+1$, *$V = [\bar{V}_{11}\ \bar{V}_{12}\ \ldots\ \bar{V}_{1(N+1)}]$, where the block columns $\bar{V}_{1t_2} \in \mathbb{R}^{(n+s)\times m}$ for all $t_2 = 1, \ldots, N+1$, and have $U^{\uparrow} = \begin{bmatrix} \bar{U}_{21} \\ \vdots \\ \bar{U}_{(N+1)1} \end{bmatrix}$, $U^{\downarrow} = \begin{bmatrix} \bar{U}_{11} \\ \vdots \\ \bar{U}_{N1} \end{bmatrix}$, $\overleftarrow{V} = [\bar{V}_{12}\ \ldots\ \bar{V}_{1(N+1)}]$*

and $\overrightarrow{V} = [\bar{V}_{11}\ \ldots\ \bar{V}_{1N}]$ (for a definition of the shift operation see Section 2), then (a)*$A^0 = \Sigma^{-1/2} U^{\downarrow *} U^{\uparrow} \Sigma^{1/2}$ is the exact solution of the linear equation*

$$A^0(\Sigma^{1/2}\overrightarrow{V}) = \Sigma^{1/2}\overleftarrow{V}.$$

(b) $A^0 = \Sigma^{-1/2} U^{\downarrow *} U^{\uparrow} \Sigma^{1/2}$ *is the exact solution of the linear equation*

$$(U\Sigma^{1/2})^{\downarrow} A^0 = (U\Sigma^{1/2})^{\uparrow}.$$

(c) *the controllability matrix and the observability matrix are*

$$\begin{bmatrix} C^0 \\ C^0 A^0 \\ C^0(A^0)^2 \\ \vdots \\ C^0(A^0)^N \end{bmatrix} = U\Sigma^{1/2} and \begin{bmatrix} B^0 & A^0B^0 & (A^0)^2B^0 & \ldots & (A^0)^N B^0 \end{bmatrix} = \Sigma^{1/2} V,$$

respectively.

(d) $(A^0,\ B^0,\ C^0)$ *is a minimal and stable realization and* $(A^0)^i = 0,\ i \geq N.$

(e) *the system* $(A^0,\ B^0,\ C^0)$ *is a realization of the sequence P, i.e.*

$$P(k) = C^0(A^0)^{k-1}B^0, \quad k = 1, 2, 3, \ldots, N,$$

and .

$$C^0(A^0)^{k-1}B^0 = 0, \ k > N$$

(f) $(A^0,\ B^0,\ C^0)$ *is a balanced realization, i.e.*

$$A^0 P_g (A^0)^* - P_g + B^0(B^0)^* = 0,$$
$$(A^0)^* Q_g A^0 - Q_g + (C^0)^* C^0 = 0,$$

where $P_g = Q_g = \Sigma.$

4. *if the exact realization* $(A^0,\ B^0,\ C^0)$, *i.e.* $s = 0$, *and the corresponding Gramians are decomposed conformally as*

$$A^0 = \begin{bmatrix} A_{11}^0 & A_{12}^0 \\ A_{21}^0 & A_{22}^0 \end{bmatrix}, \quad B^0 = \begin{bmatrix} B_1^0 \\ B_2^0 \end{bmatrix}, \quad C^0 = [C_1^0 \quad C_2^0], \text{ and } \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}$$

such that

$$C^0 A^0 = [C_1^0 \ \ C_2^0] \begin{bmatrix} A_{11}^0 & A_{12}^0 \\ A_{21}^0 & A_{22}^0 \end{bmatrix} = [C_1^0 A_{11}^0 + C_2^0 A_{21}^0, C_1^0 A_{12}^0 + C_2^0 A_{22}^0],$$

and

$$A^0 B^0 = \begin{bmatrix} A_{11}^0 & A_{12}^0 \\ A_{21}^0 & A_{22}^0 \end{bmatrix} \begin{bmatrix} B_1^0 \\ B_2^0 \end{bmatrix} = \begin{bmatrix} A_{11}^0 B_1^0 + A_{12}^0 B_2^0 \\ A_{21}^0 B_1^0 + A_{22}^0 B_2^0 \end{bmatrix},$$

with $\Sigma_2 \in \mathbb{R}^{s_a \times s_a}$, *for some integer* $s_a \geq 1$, *and* $(A^{s_a},\ B^{s_a},\ C^{s_a})$ *is the realization obtained via Algorithm* 2 *for the same* s_a, *then*
(a) *we have*

$$A^{s_a} = A^0_{11}, \quad B^{s_a} = B^0_1, \quad \text{and} \quad C^{s_a} = C^0_1.$$

(b) *the realization* $(A^{s_a},\ B^{s_a},\ C^{s_a})$ *is minimal and asymptotically stable and we have*

$$\begin{aligned} &A^{s_a}\Sigma_1(A^{s_a})^* - \Sigma_1 + B^{s_a}(B^{s_a})^* \leq 0, \\ &(A^{s_a})^*\Sigma_1 A^{s_a} - \Sigma_1 + (C^{s_a})^* C^{s_a} \leq 0. \end{aligned}$$

Proof: 1. Let $H = U\Sigma V$ be a singular value decomposition. Based on the sizes of the matrices given in Algorithm 2 we have that $\Sigma \in \mathbb{R}^{(n+s)\times(n+s)}$, $U \in \mathbb{R}^{(N+1)p\times(n+s)}$ and $V \in \mathbb{R}^{(n+s)\times(N+1)m}$. Let U and V be partitioned as $U = \begin{bmatrix} U^{\downarrow} \\ U^{L_r} \end{bmatrix}$ and $V = [\overrightarrow{V} \quad V^{L_c}]$ (for a definition of the shift operation see Section 2), where $U^{L_r} \in \mathbb{R}^{p\times(n+s)}$ and $V^{L_c} \in \mathbb{R}^{(n+s)\times m}$. Then

$$\left[\begin{array}{ccccc|c} P(1) & P(2) & \cdots & P(N-1) & P(N) & 0 \\ P(2) & P(3) & \cdots & P(N) & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ P(N) & 0 & \cdots & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & \cdots & 0 & 0 \end{array}\right] = H$$

$$= \left[\begin{array}{c} U^{\downarrow} \\ \hline U^{L_r} \end{array}\right] \Sigma [\overrightarrow{V} \mid V^{L_c}] = \left[\begin{array}{c|c} U^{\downarrow}\Sigma\overrightarrow{V} & U^{\downarrow}\Sigma V^{L_c} \\ \hline U^{L_r}\Sigma\overrightarrow{V} & U^{L_r}\Sigma V^{L_c} \end{array}\right].$$

Therefore, we have

$$0 = [U^{L_r}\Sigma\overrightarrow{V} \mid U^{L_r}\Sigma V^{L_c}] = U^{L_r}\Sigma V, \quad 0 = \left[\begin{array}{c} U^{\downarrow}\Sigma V^{L_c} \\ \hline U^{L_r}\Sigma V^{L_c} \end{array}\right] = U\Sigma V^{L_c},$$

where 0 denotes a block of zeros of appropriate sizes. Hence,

$$\begin{aligned} U^{L_r} =& U^{L_r}(\Sigma V V^* \Sigma^{-1}) = (U^{L_r}\Sigma V) V^* \Sigma^{-1} = 0, \\ V^{L_c} =& (\Sigma^{-1} U^* U \Sigma) V^{L_c} = \Sigma^{-1} U^* (U \Sigma V^{L_c}) = 0. \end{aligned}$$

Consequently,

$$H = \left[\begin{array}{ccccc|c} P(1) & P(2) & \cdots & P(N-1) & P(N) & 0 \\ P(2) & P(3) & \cdots & P(N) & 0 & 0 \\ \vdots & & & & \vdots & \vdots \\ P(N) & 0 & \cdots & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & \cdots & 0 & 0 \end{array}\right] = \left[\begin{array}{c|c} U^{\downarrow}\Sigma\overrightarrow{V} & 0 \\ \hline 0 & 0 \end{array}\right],$$

and

$$U^{\downarrow *}U^{\downarrow} = U^{\downarrow *}U^{\downarrow} + (U^{L_r})^* U^{L_r} = [U^{\downarrow *}\ (U^{L_r})^*]\begin{bmatrix} U^{\downarrow} \\ U^{L_r} \end{bmatrix} = U^* U = I.$$

Similarly we have that $\overrightarrow{V}\overrightarrow{V}^* = I$. Considering the partition used earlier we have that $U^{\downarrow} = [U_1^{\downarrow},\ U_2^{\downarrow}]$, where $U_1^{\downarrow}$ is a $Np \times n$ matrix and $U_2^{\downarrow}$ is a $Np \times s$ matrix. Therefore, we have that

$$\begin{bmatrix} U_1^{\downarrow *}U_1^{\downarrow} & U_1^{\downarrow *}U_2^{\downarrow} \\ U_2^{\downarrow *}U_1^{\downarrow} & U_2^{\downarrow *}U_2^{\downarrow} \end{bmatrix} = [U_1^{\downarrow}\ U_2^{\downarrow}]^*[U_1^{\downarrow}\ U_2^{\downarrow}] = U^{\downarrow *}U^{\downarrow} = I.$$

As a result, we have $U_1^{\downarrow *}U_1^{\downarrow} = I$, $U_2^{\downarrow *}U_2^{\downarrow} = I$, $U_1^{\downarrow *}U_2^{\downarrow} = 0$ and $U_2^{\downarrow *}U_1^{\downarrow} = 0$.

Let $X_o \in \mathbb{R}^{n\times n}$ be the solution to a least-squares problem, i.e.

$$\inf_{X\in\mathbb{R}^{n\times n}} ||EX - F||_F = ||EX_o - F||_F.$$

If E has full column rank then X_o can be calculated as [24] $X_o = (E^*E)^{-1}E^*F$. Let $\hat{X}$ be the least-squares solution such that

$$||(U_1\Sigma_1^{1/2})^{\downarrow}\hat{X} - (U_1\Sigma_1^{1/2})^{\uparrow}||_F = \inf_{X\in\mathbb{R}^{n\times n}} ||(U_1\Sigma_1^{1/2})^{\downarrow}X - (U_1\Sigma_1^{1/2})^{\uparrow}||_F,$$

where $(U_1\Sigma_1^{1/2})^{\uparrow} = U_1^{\uparrow}\Sigma_1^{1/2}$ and $(U_1\Sigma_1^{1/2})^{\downarrow} = U_1^{\downarrow}\Sigma_1^{1/2}$. From the previous discussion we know $U_1^{\downarrow}$ consists of orthogonal columns and Σ_1 has full rank. As a result, $(U_1\Sigma_1^{1/2})^{\downarrow}$ has full column rank. Therefore, we have

$$\begin{aligned} \hat{X} &= ((U_1\Sigma_1^{1/2})^{\downarrow *}(U_1\Sigma_1^{1/2})^{\downarrow})^{-1}(U_1\Sigma_1^{1/2})^{\downarrow *}(U_1\Sigma_1^{1/2})^{\uparrow} \\ &= (\Sigma_1^{1/2}U_1^{\downarrow *}U_1^{\downarrow}\Sigma_1^{1/2})^{-1}\Sigma_1^{1/2}U_1^{\downarrow *}U_1^{\uparrow}\Sigma_1^{1/2} \\ &= \Sigma_1^{-1}\Sigma_1^{1/2}U_1^{\downarrow *}U_1^{\uparrow}\Sigma_1^{1/2} = \Sigma_1^{-1/2}U_1^{\downarrow *}U_1^{\uparrow}\Sigma_1^{1/2} \\ &= A^s. \end{aligned}$$

2. Let us decompose the matrices U and V as $U = \begin{bmatrix} U^{F_r} \\ U^{\uparrow} \end{bmatrix}$ and $V = [V^{F_c}\ \overleftarrow{V}]$, where $U^{F_r} \in \mathbb{R}^{p\times(n+s)}$ is the first block row of U and $V^{F_c} \in \mathbb{R}^{(n+s)\times m}$ is the first block column of V (for a definition of the shift operation see Section 2). We have

$$H = \left[\begin{array}{cccccc|c} P(1) & P(2) & \cdots & P(N-1) & P(N) & 0 \\ P(2) & P(3) & \cdots & P(N) & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ P(N-1) & P(N) & \cdots & \cdots & 0 & 0 \\ P(N) & 0 & \cdots & \cdots & 0 & 0 \\ \hline 0 & 0 & \cdots & \cdots & 0 & 0 \end{array}\right]$$

$$= U\Sigma V = \left[\begin{array}{c} U^{\downarrow} \\ \hline U^{L_r} \end{array}\right] \Sigma [V^{F_c}\ |\overleftarrow{V}] = \left[\begin{array}{c|c} U^{\downarrow}\Sigma V^{F_c} & U^{\downarrow}\Sigma\overleftarrow{V} \\ \hline U^{L_r}\Sigma V^{F_c} & U^{L_r}\Sigma\overleftarrow{V} \end{array}\right]. \tag{4}$$

Similarly we get

$$H = \left[\begin{array}{ccccc|c} P(1) & P(2) & \cdots & P(N-1) & P(N) & 0 \\ \hline P(2) & P(3) & \cdots & P(N) & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ P(N-1) & P(N) & \cdots & \cdots & 0 & 0 \\ P(N) & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 0 \end{array}\right]$$

$$= U\Sigma V = \left[\begin{array}{c} U^{F_r} \\ \hline U^{\uparrow} \end{array}\right] \Sigma [\overrightarrow{V}\ |V^{L_c}] = \left[\begin{array}{c|c} U^{F_r}\Sigma\overrightarrow{V} & U^{F_r}\Sigma V^{L_c} \\ \hline U^{\uparrow}\Sigma\overrightarrow{V} & U^{\uparrow}\Sigma V^{L_c} \end{array}\right]. \tag{5}$$

Note that the (1,2) block of H in Equation (4) equals the (2,1) block of H in Equation (5). Therefore,

$$U^{\downarrow}\Sigma\overleftarrow{V} = U^{\uparrow}\Sigma\overrightarrow{V}. \tag{6}$$

We substitute $\Sigma = \mathrm{diag}(\Sigma_1, \Sigma_2)$, $U^{\downarrow} = [U_1^{\downarrow}\ U_2^{\downarrow}]$, $U^{\uparrow} = [U_1^{\uparrow}\ U_2^{\uparrow}]$, $\overleftarrow{V} = \begin{bmatrix} \overleftarrow{V}_1 \\ \overleftarrow{V}_2 \end{bmatrix}$ and $\overrightarrow{V} = \begin{bmatrix} \overrightarrow{V}_1 \\ \overrightarrow{V}_2 \end{bmatrix}$ in Equation (6), where, from the partition discussed earlier and the notation defined in Section 2, the $Np \times s$ matrix $U_2^{\uparrow}$ is given by dropping the first block row of U_2, the $n \times Nm$ matrix $\overleftarrow{V}_1$ is given by dropping the first block column of V_1, the $n \times Nm$ matrix $\overrightarrow{V}_1$ is given by dropping the last block column of V_1, the $s \times Nm$ matrix $\overleftarrow{V}_2$ is given by dropping the first block column of V_2, and the $s \times Nm$ matrix $\overrightarrow{V}_2$ is given by dropping the last block column of V_2. We have that

$$U_1^{\downarrow}\Sigma_1\overleftarrow{V}_1 + U_2^{\downarrow}\Sigma_2\overleftarrow{V}_2 = U_1^{\uparrow}\Sigma_1\overrightarrow{V}_1 + U_2^{\uparrow}\Sigma_2\overrightarrow{V}_2. \tag{7}$$

Using $U_1^{\downarrow *}U_1^{\downarrow} = I$, $U_1^{\downarrow *}U_2^{\downarrow} = 0$ and multiplying $\Sigma_1^{-1/2}U_1^{\downarrow *}$ on both sides of Equation (7) we get

$$\Sigma_1^{1/2}\overleftarrow{V}_1 = \Sigma_1^{-1/2}U_1^{\downarrow *}U_1^{\uparrow}\Sigma_1\overrightarrow{V}_1 + \Sigma_1^{-1/2}U_1^{\downarrow *}U_2^{\uparrow}\Sigma_2\overrightarrow{V}_2. \tag{8}$$

Using the fact that $\overrightarrow{V}_1\overrightarrow{V}_1^{*} = I$, $\overrightarrow{V}_2\overrightarrow{V}_1^{*} = 0$ and multiplying both sides of Equation (8) with $\overrightarrow{V}_1^{*}\Sigma_1^{-1/2}$ we get

$$\Sigma_1^{1/2}\overleftarrow{V}_1\overrightarrow{V}_1^{*}\Sigma_1^{-1/2} = \Sigma_1^{-1/2}U_1^{\downarrow *}U_1^{\uparrow}\Sigma_1^{1/2} = A^s.$$

The solution to the least-squares problem, i.e.

$$||X_o(\Sigma_1^{1/2}\overrightarrow{V}_1) - \Sigma_1^{1/2}\overleftarrow{V}_1||_F = \inf_{X\in\mathbb{R}^{n\times n}} ||X(\Sigma_1^{1/2}\overrightarrow{V}_1) - \Sigma_1^{1/2}\overleftarrow{V}_1||_F,$$

is given by

$$\begin{aligned} X_o &= \Sigma_1^{1/2}\overleftarrow{V}_1(\Sigma_1^{1/2}\overrightarrow{V}_1)^{*}((\Sigma_1^{1/2}\overrightarrow{V}_1)(\Sigma_1^{1/2}\overrightarrow{V}_1)^{*})^{-1} \\ &= \Sigma_1^{1/2}\overleftarrow{V}_1(\Sigma_1^{1/2}\overrightarrow{V}_1)^{*}(\Sigma_1^{1/2}\overrightarrow{V}_1\overrightarrow{V}_1^{*}\Sigma_1^{1/2})^{-1} \\ &= \Sigma_1^{1/2}\overleftarrow{V}_1\overrightarrow{V}_1^{*}\Sigma_1^{1/2}\Sigma_1^{-1} = \Sigma_1^{1/2}\overleftarrow{V}_1\overrightarrow{V}_1^{*}\Sigma_1^{-1/2} = A^s. \end{aligned}$$

3(a). When $s = 0$ we have $\Sigma_1 = \Sigma$, $U_1 = U$ and $V_1 = V$. With direct substitution, Equation (3) becomes

$$A^0 = \Sigma^{-1/2}U^{\downarrow *}U^{\uparrow}\Sigma^{1/2}.$$

Using that $U^{\downarrow *}U^{\downarrow} = I$ we have from Equation (6) that

$$\begin{aligned} \Sigma^{1/2}\overleftarrow{V} &= \Sigma^{-1/2}U^{\downarrow *}(U^{\downarrow}\Sigma\overleftarrow{V}) = \Sigma^{-1/2}U^{\downarrow *}(U^{\uparrow}\Sigma\overrightarrow{V}) \\ &= (\Sigma^{-1/2}U^{\downarrow *}U^{\uparrow}\Sigma^{1/2})\Sigma^{1/2}\overrightarrow{V} = A^0\Sigma^{1/2}\overrightarrow{V}. \end{aligned} \tag{9}$$

Therefore, $A^0 = \Sigma^{-1/2}U^{\downarrow *}U^{\uparrow}\Sigma^{1/2}$ is the exact solution of $\Sigma^{1/2}\overleftarrow{V} = A^0\Sigma^{1/2}\overrightarrow{V}$.

3(b). Using Equation (9) and the fact that $\overrightarrow{V}\overrightarrow{V}^{*} = I$ we have that

$$A^0 = (A^0\Sigma^{1/2}\overrightarrow{V})\overrightarrow{V}^{*}\Sigma^{-1/2} = \Sigma^{1/2}\overleftarrow{V}\overrightarrow{V}^{*}\Sigma^{-1/2}. \tag{10}$$

Then making use of both Equation (6) and Equation (10) we have

$$\begin{aligned} U^{\uparrow}\Sigma^{1/2} &= (U^{\uparrow}\Sigma\overrightarrow{V})\overrightarrow{V}^{*}\Sigma^{-1/2} = (U^{\downarrow}\Sigma\overleftarrow{V})\overrightarrow{V}^{*}\Sigma^{-1/2} \\ &= U^{\downarrow}\Sigma^{1/2}(\Sigma^{1/2}\overleftarrow{V}\overrightarrow{V}^{*}\Sigma^{-1/2}) = U^{\downarrow}\Sigma^{1/2}A^{0}. \end{aligned}$$

Therefore A^0 is also the exact solution of the equation $(U\Sigma^{1/2})^{\downarrow}A^0 = (U\Sigma^{1/2})^{\uparrow}$.

3(c). Let us rewrite $U\Sigma^{1/2} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N+1} \end{bmatrix}$, where $w_i \in \mathbb{R}^{p\times m}$, $i = 1, 2, \dots, N+1$. Since C^0 is the first p rows of $U\Sigma^{1/2}$ we have $w_1 = C^0$. Because A^0 is the exact solution of $\begin{bmatrix} w_2 \\ w_3 \\ \vdots \\ w_{N+1} \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} A^0$ we have $w_i = w_{i-1}A^0$, $i = 2, 3, \dots, N+1$. As a result,

$$U\Sigma^{1/2} = \begin{bmatrix} C^0 \\ C^0A^0 \\ \vdots \\ C^0(A^0)^N \end{bmatrix}.$$

Since A^0 is also the solution of $\Sigma^{1/2}\overleftarrow{V} = A^0\Sigma^{1/2}\overrightarrow{V}$ we similarly have that

$$\Sigma^{1/2}V = [B^0 \; A^0B^0 \; \cdots \; (A^0)^NB^0].$$

3(d). We have

$$\begin{bmatrix} U^{\downarrow}\Sigma \\ U^{L_r}\Sigma \end{bmatrix} = U\Sigma^{1/2} = \left[\begin{array}{c} C^0 \\ C^0A^0 \\ \vdots \\ \hline C^0(A^0)^N \end{array}\right],$$

and it has been shown that $U^{L_r} = 0$ in the proof of Proposition 2 part 1. As a result, $C^0(A^0)^N = 0$. Similarly, we have that $(A^0)^NB^0 = 0$. Therefore,

$$\begin{aligned} \operatorname{rank}([B^0 \;\; A^0B^0 \;\; \dots \;\; (A^0)^{N-1}B^0]) &= \operatorname{rank}(\Sigma^{1/2}\overrightarrow{V}) = \operatorname{rank}(\Sigma) \\ = \operatorname{rank}(U^{\downarrow}\Sigma^{1/2}) &= \operatorname{rank}\left(\begin{bmatrix} C^0 \\ C^0A^0 \\ \vdots \\ C^0(A^0)^{N-1} \end{bmatrix}\right). \end{aligned}$$

Since Σ has full rank $(A^0,\ B^0)$ is controllable and $(C^0,\ A^0)$ is observable. Hence the realization $(A^0,\ B^0,\ C^0)$ is minimal.

Since $C^0(A^0)^N = 0$ and $(A^0)^N B^0 = 0$ we also have $C^0(A^0)^i = 0,\ i \geq N$, and $(A^0)^i B^0 = 0,\ i \geq N$. As a result, we have $(A^0)^N[B^0\ A^0B^0\ \cdots\ (A^0)^{N-1}B^0] = 0$. Since $[B^0\ A^0B^0\ \cdots\ (A^0)^{N-1}B^0]$ has full row rank we have $(A^0)^N = 0$. Consequently, we also have

$$(A^0)^i = 0,\ i \geq N.$$

Therefore, the realization $(A^0,\ B^0,\ C^0)$ is asymptotically stable.

3(e). When $s = 0$

$$\begin{aligned}
H &= \begin{bmatrix} P(1) & P(2) & \cdots & P(N-1) & P(N) & 0 \\ P(2) & P(3) & \cdots & P(N) & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ P(N) & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 0 \end{bmatrix} = U\Sigma^{1/2}\Sigma^{1/2}V \\
&= \begin{bmatrix} C^0 \\ C^0A^0 \\ \vdots \\ C^0(A^0)^{N-1} \\ C^0(A^0)^N \end{bmatrix} [B^0\ A^0B^0\ \cdots\ (A^0)^{N-1}B^0\ (A^0)^N B^0] \\
&= \begin{bmatrix} C^0B^0 & C^0A^0B^0 & C^0(A^0)^2B^0 & \cdots & C^0(A^0)^{N-1}B^0 & 0 \\ C^0A^0B^0 & C^0(A^0)^2B^0 & C^0(A^0)^3B^0 & \cdots & 0 & 0 \\ C^0(A^0)^2B^0 & C^0(A^0)^3B^0 & C^0(A^0)^4B^0 & \cdots & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ C^0(A^0)^{N-1}B^0 & 0 & \cdots & \cdots & 0 & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}.
\end{aligned}$$

As a result, the system is a realization of the sequence, i.e.

$$P(k) = C^0(A^0)^{k-1}B^0, \quad k = 1, 2, 3, \ldots, N.$$

From Proposition 1 part 3(d) we know $(A^0)^i = 0,\ i \geq N$. Therefore,

$$C^0(A^0)^{k-1}B^0 = 0, \quad k > N.$$

3(f). The observability Gramian P_g and the controllability Gramian Q_g can be calculated as

$$\begin{aligned}
P_g &= \sum_{k=1}^{\infty}((A^0)^{k-1}B^0)((A^0)^{k-1}B^0)^* = \sum_{k=1}^{N}((A^0)^{k-1}B^0)((A^0)^{k-1}B^0)^* \\
&= \Sigma^{1/2}V(\Sigma^{1/2}V)^* = \Sigma > 0, \\
Q_g &= \sum_{k=1}^{\infty}(C^0(A^0)^{k-1})^*(C^0(A^0)^{k-1}) = \sum_{k=1}^{N}(C^0(A^0)^{k-1})^*(C^0(A^0)^{k-1}) \\
&= (U\Sigma^{1/2})^*U\Sigma^{1/2} = \Sigma > 0.
\end{aligned}$$

Therefore, $P_g = Q_g$ and the Gramians are diagonal, which shows that the system is balanced. That the Lyapunov equations hold is a standard result.

4(a). We know

$$\begin{bmatrix} [C_1^0\ C_2^0] \\ [C_1^0\ C_2^0]A^0 \\ \vdots \end{bmatrix} = U\Sigma^{1/2} = [U_1\ U_2]\begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}^{1/2} = [U_1\Sigma_1^{1/2}\ U_2\Sigma_2^{1/2}],$$

and

$$\left[\begin{bmatrix} B_1^0 \\ B_2^0 \end{bmatrix} \quad A^0\begin{bmatrix} B_1^0 \\ B_2^0 \end{bmatrix} \quad \cdots\right] = \Sigma^{1/2}V = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}^{1/2}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \Sigma_1^{1/2}V_1 \\ \Sigma_2^{1/2}V_2 \end{bmatrix}.$$

By definition C^{s_a} is the first block row in $U_1\Sigma_1^{1/2}$ and B^{s_a} is the first block column in $\Sigma_1^{1/2}V_1$. Consequently, $C^{s_a} = C_1^0$ and $B^{s_a} = B_1^0$. Since A^0 is the solution of $(U\Sigma^{1/2})^{\uparrow} = (U\Sigma^{1/2})^{\downarrow}A^0$ we have that

$$[(U_1\Sigma_1^{1/2})^{\uparrow} \quad (U_2\Sigma_2^{1/2})^{\uparrow}] = [(U_1\Sigma_1^{1/2})^{\downarrow} \quad (U_2\Sigma_2^{1/2})^{\downarrow}]\begin{bmatrix} A_{11}^0 & A_{12}^0 \\ A_{21}^0 & A_{22}^0 \end{bmatrix}.$$

As a result,

$$(U_1\Sigma_1^{1/2})^{\uparrow} = (U_1\Sigma_1^{1/2})^{\downarrow}A_{11}^0 + (U_2\Sigma_2^{1/2})^{\downarrow}A_{21}^0. \tag{11}$$

Rearranging Equation (11) yields

$$(U_1\Sigma_1^{1/2})^{\downarrow}A_{11}^0 = (U_1\Sigma_1^{1/2})^{\uparrow} - (U_2\Sigma_2^{1/2})^{\downarrow}A_{21}^0.$$

Using that $U_1^{\downarrow *}U_2^{\downarrow} = 0$ we therefore have

$$A_{11}^0 = \Sigma_1^{-1/2}U_1^{\downarrow *}U_1^{\uparrow}\Sigma_1^{1/2} - \Sigma_1^{-1/2}U_1^{\downarrow *}U_2^{\downarrow}\Sigma_2^{1/2}A_{21}^0 = \Sigma_1^{-1/2}U_1^{\downarrow *}U_1^{\uparrow}\Sigma_1^{1/2} = A^{s_a}.$$

4(b). From Proposition 1 part 3(d) and Proposition 1 part 3(f) we know $(A^0,\ B^0,\ C^0)$ is a minimal, stable and balanced realization. Therefore, the realization $(A^{s_a},\ B^{s_a},\ C^{s_a})$ is minimal and stable [25]. From Proposition 1 part 3(f) we have also

$$\begin{aligned} A^0 P_g (A^0)^* - P_g + B^0 (B^0)^* &= 0, \\ (A^0)^* Q_g A^0 - Q_g + (C^0)^* C^0 &= 0. \end{aligned}$$

Considering the partitions of the realization $(A^0,\ B^0,\ C^0)$ we have

$$\begin{aligned} 0 &= A^0 P_g (A^0)^* - P_g + B^0 (B^0)^* \\ &= \begin{bmatrix} A^0_{11} & A^0_{12} \\ A^0_{21} & A^0_{22} \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} A^0_{11} & A^0_{12} \\ A^0_{21} & A^0_{22} \end{bmatrix}^* - \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} + \begin{bmatrix} B^0_1 \\ B^0_2 \end{bmatrix} \begin{bmatrix} B^0_1 \\ B^0_2 \end{bmatrix}^* \\ &= \left[\begin{matrix} A^0_{11}\Sigma_1(A^0_{11})^* + A^0_{12}\Sigma_2(A^0_{12})^* - \Sigma_1 + B^0_1(B^0_1)^* \\ A^0_{21}\Sigma_1(A^0_{11})^* + A^0_{22}\Sigma_2(A^0_{12})^* + B^0_2(B^0_1)^* \end{matrix} \right. \\ &\qquad \left. \begin{matrix} A^0_{11}\Sigma_1(A^0_{21})^* + A^0_{12}\Sigma_2(A^0_{22})^* + B^0_1(B^0_2)^* \\ A^0_{21}\Sigma_1(A^0_{21})^* + A^0_{22}\Sigma_2(A^0_{22})^* - \Sigma_2 + B^0_2(B^0_2)^* \end{matrix} \right]. \end{aligned}$$

Since $A^{s_a} = A^0_{11},\ \ C^{s_a} = C^0_1$ and $B^{s_a} = B^0_1$ and $A^0_{12}\Sigma_2(A^0_{12})^* \geq 0$ we have

$$A^{s_a}\Sigma_1(A^{s_a})^* - \Sigma_1 + B^{s_a}(B^{s_a})^* \leq 0.$$

Similarly, we can show

$$(A^{s_a})^*\Sigma_1 A^{s_a} - \Sigma_1 + (C^{s_a})^* C^{s_a} \leq 0. \qquad \blacksquare$$

Note that a number of modifications of Algorithm 2 are possible, although with many of the modifications the results of the previous proposition no longer hold. For example, the Hankel matrix can be constructed as ([11])

$$\hat{H} = \begin{bmatrix} P(1) & P(2) & \cdots & P(N-1) & P(N) \\ P(2) & P(3) & \cdots & P(N) & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ P(N) & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Let us denote by $(\hat{A^s}, \hat{B^s}, \hat{C^s})$ the system obtained from Algorithm 2, if the Hankel matrix $\hat{H}$ is used instead of the Hankel matrix H.

It follows from the proof of the previous proposition that if $H = U\Sigma V$ is a singular value decomposition of H then $\hat{H} = \hat{U}\Sigma\hat{V}$ is a singular value decomposition of $\hat{H}$, where $\hat{U} := U^{\downarrow}$ and $\hat{V} := \overrightarrow{V}$. Now let us partition these matrices as $\hat{U} = [\hat{U}_1 \ \ \hat{U}_2]$,

$\hat{\Sigma} = \begin{bmatrix} \hat{\Sigma}_1 & 0 \\ 0 & \hat{\Sigma}_2 \end{bmatrix}$, $\hat{V} = \begin{bmatrix} \hat{V}_1 \\ \hat{V}_2 \end{bmatrix}$. Since the first block rows of $\hat{U}_1\Sigma_1^{1/2}$ and $U_1\Sigma_1^{1/2}$ coincide we have that the B matrices agree, i.e. $\hat{B}^s = B^s$. Similarly, since the first block columns of $\Sigma_1^{1/2}\hat{V}_1$ and $\Sigma_1^{1/2}V_1$ coincide we have that $\hat{C}^s = C^s$.

However, in contrast to the B and C matrices the A matrices typically differ. The last block row in $\hat{U}$ is not necessarily zero. Therefore, $\hat{U}^{\downarrow}$ is, in general, not a matrix with orthonormal columns. As a result, although $\hat{A}^s$ is the least-squares solution of the following problem

$$\|(\hat{U}_1\Sigma_1^{1/2})^{\downarrow}\hat{A}^s - (\hat{U}_1\Sigma_1^{1/2})^{\uparrow}\|_F = \inf_{X\in\mathbb{R}^{n\times n}} \|(\hat{U}_1\Sigma_1^{1/2})^{\downarrow}X - (\hat{U}_1\Sigma_1^{1/2})^{\uparrow}\|_F \tag{12}$$

it cannot be calculated by using Equation (3). Consequently, $\hat{A}^s$ is in general not the exact solution of $(\hat{U}_1\Sigma_1^{1/2})^{\downarrow}X = (\hat{U}_1\Sigma_1^{1/2})^{\uparrow}$, even when $s = 0$. This implies that if Algorithm 2 is based on the Hankel matrix $\hat{H}$ instead of H, in general, we cannot expect to obtain an exact realization.

To examine the connection between A^s and $\hat{A}^s$ and applying the shift operations defined in Section 2, we first decompose $\hat{U}_1 = \begin{bmatrix} \hat{U}_1^{\downarrow} \\ \hat{U}_1^{L_r} \end{bmatrix}$, where $\hat{U}_1^{L_r} \in \mathbb{R}^{p\times n}$ is the last block row of $\hat{U}_1$. We also know $\hat{U}_1 = U_1^{\downarrow}$ and $U_1^{\uparrow} = \begin{bmatrix} \hat{U}_1^{\uparrow} \\ 0 \end{bmatrix}$. From Algorithm 2, if $(\hat{U}_1\Sigma_1^{1/2})^{\downarrow}$ has full column rank the least-squares solution $\hat{A}^s$ of Equation (12) is calculated as

$$\begin{aligned}
\hat{A}^s =&((\hat{U}_1\Sigma_1^{1/2})^{\downarrow *}(\hat{U}_1\Sigma_1^{1/2})^{\downarrow})^{-1}(\hat{U}_1\Sigma_1^{1/2})^{\downarrow *}(\hat{U}_1\Sigma_1^{1/2})^{\uparrow} \\
=&(\Sigma_1^{1/2}\hat{U}_1^{\downarrow *}\hat{U}_1^{\downarrow}\Sigma_1^{1/2})^{-1}\Sigma_1^{1/2}\hat{U}_1^{\downarrow *}(\hat{U}_1\Sigma_1^{1/2})^{\uparrow} \\
=&\Sigma_1^{-1/2}(\hat{U}_1^{\downarrow *}\hat{U}_1^{\downarrow})^{-1}\Sigma_1^{-1/2}\Sigma_1^{1/2}\hat{U}_1^{\downarrow *}(\hat{U}_1\Sigma_1^{1/2})^{\uparrow} \\
=&\Sigma_1^{-1/2}(\hat{U}_1^{\downarrow *}\hat{U}_1^{\downarrow})^{-1}\hat{U}_1^{\downarrow *}\hat{U}_1^{\uparrow}\Sigma_1^{1/2}.
\end{aligned} \tag{13}$$

From the previous discussion we know

$$\hat{U}_1^{\downarrow *}\hat{U}_1^{\downarrow} + (\hat{U}_1^{L_r})^*\hat{U}_1^{L_r} = \begin{bmatrix} \hat{U}_1^{\downarrow} \\ \hat{U}_1^{L_r} \end{bmatrix}^* \begin{bmatrix} \hat{U}_1^{\downarrow} \\ \hat{U}_1^{L_r} \end{bmatrix} = U_1^*U_1 = I_n.$$

As a result we have $\hat{U}_1^{\downarrow *}\hat{U}_1^{\downarrow} = I_n - (\hat{U}_1^{L_r})^*\hat{U}_1^{L_r}$. Based on the well-known matrix inversion formula [19]

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1},$$

provided matrices A and C are invertible, we rewrite

$$\begin{aligned}(\hat{U}_1^{\downarrow *}\hat{U}_1^{\downarrow})^{-1} =&(I_n - (\hat{U}_1^{L_r})^{*}\hat{U}_1^{L_r})^{-1} = (I_n + (-\hat{U}_1^{L_r})^{*}I_p\hat{U}_1^{L_r})^{-1}\\ =&I_n + (\hat{U}_1^{L_r})^{*}(I_p - \hat{U}_1^{L_r}(\hat{U}_1^{L_r})^{*})^{-1}\hat{U}_1^{L_r}.\end{aligned} \tag{14}$$

Replacing the term $(\hat{U}_1^{\downarrow *}\hat{U}_1^{\downarrow})^{-1}$ in Equation (13) by Equation (14) yields

$$\begin{aligned}\widehat{A^s} =&\Sigma_1^{-1/2}(I_n + (\hat{U}_1^{L_r})^{*}(I_p - \hat{U}_1^{L_r}(\hat{U}_1^{L_r})^{*})^{-1}\hat{U}_1^{L_r})\hat{U}_1^{\downarrow *}\hat{U}_1^{\uparrow}\Sigma_1^{1/2}\\ =&\Sigma_1^{-1/2}\hat{U}_1^{\downarrow *}\hat{U}_1^{\uparrow}\Sigma_1^{1/2} + \Sigma_1^{-1/2}(\hat{U}_1^{L_r})^{*}(I_p - \hat{U}_1^{L_r}(\hat{U}_1^{L_r})^{*})^{-1}\hat{U}_1^{L_r}\hat{U}_1^{\downarrow *}\hat{U}_1^{\uparrow}\Sigma_1^{1/2}.\end{aligned} \tag{15}$$

We also have

$$A^s =\Sigma_1^{-1/2}U_1^{\downarrow *}U_1^{\uparrow}\Sigma_1^{1/2} = \Sigma_1^{-1/2}\begin{bmatrix}\hat{U}_1^{\downarrow}\\ \hat{U}_1^{L_r}\end{bmatrix}^{*}\begin{bmatrix}\hat{U}_1^{\uparrow}\\ 0\end{bmatrix}\Sigma_1^{1/2} = \Sigma_1^{-1/2}\hat{U}_1^{\downarrow *}\hat{U}_1^{\uparrow}\Sigma_1^{1/2},$$

which is exactly the same as the first term in Equation (15). Therefore we can write

$$\widehat{A^s} =A^s + \Sigma_1^{-1/2}(\hat{U}_1^{L_r})^{*}(I_p - \hat{U}_1^{L_r}(\hat{U}_1^{L_r})^{*})^{-1}\hat{U}_1^{L_r}\hat{U}_1^{\downarrow *}\hat{U}_1^{\uparrow}\Sigma_1^{1/2}. \tag{16}$$

The second term in Equation (16) becomes zero, if $\hat{U}_1^{L_r} = 0$. In that case we have $\widehat{A^s} = A^s$.

If we define $\overline{A^s}$ as the solution to the modified least-squares problem such that

$$\left\|(\hat{U}_1\Sigma_1^{1/2})\overline{A^s} - \begin{bmatrix}(\hat{U}_1\Sigma_1^{1/2})^{\uparrow}\\ 0\end{bmatrix}\right\|_F = \inf_{X\in\mathbb{R}^{n\times n}}\left\|(\hat{U}_1\Sigma_1^{1/2})X - \begin{bmatrix}(\hat{U}_1\Sigma_1^{1/2})^{\uparrow}\\ 0\end{bmatrix}\right\|_F$$

then we have that

$$\begin{aligned}\overline{A^s} =&((\hat{U}_1\Sigma_1^{1/2})^{*}\hat{U}_1\Sigma_1^{1/2})^{-1}(\hat{U}_1\Sigma_1^{1/2})^{*}\begin{bmatrix}(\hat{U}_1\Sigma_1^{1/2})^{\uparrow}\\ 0\end{bmatrix}\\ =& (\Sigma_1^{1/2}\hat{U}_1^{*}\hat{U}_1\Sigma_1^{1/2})^{-1}\Sigma_1^{1/2}\hat{U}_1^{*}\begin{bmatrix}(\hat{U}_1\Sigma_1^{1/2})^{\uparrow}\\ 0\end{bmatrix}\\ =&\Sigma_1^{-1}\Sigma_1^{1/2}[\hat{U}_1^{\downarrow *}\quad(\hat{U}_1^{L_r})^{*}]\begin{bmatrix}(\hat{U}_1\Sigma_1^{1/2})^{\uparrow}\\ 0\end{bmatrix} = \Sigma_1^{-1/2}\hat{U}_1^{\downarrow *}\hat{U}_1^{\uparrow}\Sigma_1^{1/2} = A^s.\end{aligned}$$

This shows that if in Algorithm 2 the Hankel matrix H is replaced by the Hankel matrix $\hat{H}$ then this modified least-squares problem has to be solved in order to obtain the same realization as in Algorithm 2.

Algorithm 2 does provide not only a numerically stable way to obtain an exact realization for a finite one-dimensional sequence. It is well-known that reduced order balanced realizations have a number of interesting properties (see e.g. [12], [13], [14]). A further important property is that reduced order balanced realizations can be used to perform noise suppression on a sequence of noise corrupted data points. Due to the presence of noise in the sequence, the insignificant states of the full order state space realization are corrupted. By excluding those states via balanced model reduction, we effectively reduce the noise level in the sequence. This property has been successfully used in a number of applications including NMR spectroscopy (see e.g. [26], [27]) and surface plasmon resonance analysis of protein–protein interactions [28]. It is this property of noise suppression that we will also employ here in order to suppress noise in a three-dimensional image set.

3.3. *Calculation of State Space Realizations of a Three-dimensional Image Set*

For positive integers N_1, N_2 and N_3 let $P(k_1,k_2,k_3), k_i = 1,2,\ldots,N_i$, $i = 1,2,3$, be a three-dimensional image set and assume that its data points are decomposed as

$$\begin{aligned}&P(k_1,k_2,k_3) \approx P^{r_1,r_2}(k_1,k_2,k_3) = P_1^{r_1}(k_1)P_2^{r_2}(k_2)P_3^{r_2}(k_3),\\ &k_i = 1,2,\ldots,N_i, \quad i = 1,2,3,\end{aligned}$$

via Algorithm 1 with some parameters $r_1 \geq 0$ and $r_2 \geq 0$. For the one-dimensional components $P_1^{r_1}$, $P_2^{r_2}$, and $P_3^{r_2}$ of the image set and for appropriately chosen reduction parameters s_1, s_2, and s_3 approximate realizations $(A_1^{r_1;s_1}, B_1^{r_1;s_1}, C_1^{r_1;s_1})$, $(A_2^{r_2;s_2}, B_2^{r_2;s_2}, C_2^{r_2;s_2})$ and $(A_3^{r_2;s_3}, B_3^{r_2;s_3}, C_3^{r_2;s_3})$ are derived using Algorithm 2. Note that r_1 and r_2 are the numbers of the discarded singular values in step 3 and step 6 of Algorithm 1. s_1, s_2 and s_3 are the numbers of the dropped singular values in step 3 of Algorithm 2, when the state space realizations of the three one-dimensional sequences are calculated. From Proposition 1 it follows that these realizations are minimal and stable. In this way approximations of $P_1^{r_1}$, $P_2^{r_2}$, $P_3^{r_2}$ are obtained by setting

$$\begin{aligned}P_1^{r_1} &\approx P_1^{r_1;s_1}(k_1) := C_1^{r_1;s_1}(A_1^{r_1;s_1})^{k_1-1}B_1^{r_1;s_1}, \quad k_1 = 1,2,\ldots,N_1,\\ P_2^{r_2} &\approx P_2^{r_2;s_2}(k_2) := C_2^{r_2;s_2}(A_2^{r_2;s_2})^{k_2-1}B_2^{r_2;s_2}, \quad k_2 = 1,2,\ldots,N_2,\\ P_3^{r_2} &\approx P_3^{r_2;s_3}(k_3) := C_3^{r_2;s_3}(A_3^{r_2;s_3})^{k_3-1}B_3^{r_2;s_3}, \quad k_3 = 1,2,\ldots,N_3.\end{aligned}$$

It is clear that the numbers of the dropped singular values are closely related to the accuracy of the approximation. Therefore, the estimate of the three-dimensional image set P is denoted as $P^{r_1,r_2;s_1,s_2,s_3}$, and calculated in the following way

$$\begin{aligned}P(k_1,k_2,k_3) \approx&P^{r_1,r_2;s_1,s_2,s_3}(k_1,k_2,k_3)\\ :=&P_1^{r_1;s_1}(k_1)P_2^{r_2;s_2}(k_2)P_3^{r_2;s_3}(k_3)\\ =&C_1^{r_1;s_1}(A_1^{r_1;s_1})^{k_1-1}B_1^{r_1;s_1}C_2^{r_2;s_2}(A_2^{r_2;s_2})^{k_2-1}B_2^{r_2;s_2}C_3^{r_2;s_3}(A_3^{r_2;s_3})^{k_3-1}B_3^{r_2;s_3},\end{aligned}$$

$k_i = 1, 2, \ldots, N_i,\ i = 1, 2, 3$. If in neither of the two algorithms approximations are carried out we have an exact estimate of the image set given by

$$\begin{aligned} P(k_1, k_2, k_3) &= P^{0,0;0,0,0,0}(k_1, k_2, k_3) = P_1^{0;0}(k_1) P_2^{0;0}(k_2) P_3^{0;0}(k_3) \\ &= C_1^{0;0}(A_1^{0;0})^{k_1-1} B_1^{0;0} C_2^{0;0}(A_2^{0;0})^{k_2-1} B_2^{0;0} C_3^{0;0}(A_3^{0;0})^{k_3-1} B_3^{0;0}, \\ &\quad k_i = 1, 2, \ldots, N_i,\ i = 1, 2, 3. \end{aligned}$$

From Proposition 1 part 3(e) we have that

$$P^{0,0;0,0,0,0}(k_1, k_2, k_3) = C_1^{0;0}(A_1^{0;0})^{k_1-1} B_1^{0;0} C_2^{0;0}(A_2^{0;0})^{k_2-1} B_2^{0;0} C_3^{0;0}(A_3^{0;0})^{k_3-1} B_3^{0;0} = 0,$$

if $k_i > N_i$ for some $i = 1, 2, 3$, i.e. the realizations are such that the estimate is zero for indices outside those that describe the given finite data set P. From Proposition 1 part 3(d) and Proposition 1 part 3(f) we have that the realizations $(A_i^{0;0},\ B_i^{0;0},\ C_i^{0;0})$, $i = 1, 2, 3$, are asymptotically stable, minimal and balanced. It also follows from Proposition 1 part 4(a) that the approximate realizations $(A_1^{r_1;s_1},\ B_1^{r_1;s_1},\ C_1^{r_1;s_1})$, $(A_2^{r_2;s_2},\ B_2^{r_2;s_2},\ C_2^{r_2;s_2})$ and $(A_3^{r_2;s_3},\ B_3^{r_2;s_3},\ C_3^{r_2;s_3})$ are easily obtained from the corresponding full order realizations $(A_1^{r_1;0},\ B_1^{r_1;0},\ C_1^{r_1;0})$, $(A_2^{r_2;0},\ B_2^{r_2;0},\ C_2^{r_2;0})$ and $(A_3^{r_2;0},\ B_3^{r_2;0},\ C_3^{r_2;0})$ by performing a balanced approximation step. This is a computationally very simple way to obtain an approximation, if the realization of the full order system is given.

Besides the capabilities of obtaining the exact state space realizations, the proposed method can be used to suppress noise with properly chosen approximation steps in Algorithm 1 and Algorithm 2 as mentioned in Section 3.1 and Section 3.2. The use of these approximation steps will be discussed in more detail in the following section.

For a given three-dimensional image set the assignment of which spatial dimension corresponds to which index in the image array P is a matter of choice. The actual performance of the approximation algorithms can, however, in general not be expected to be independent of the particular choice that is made, in part due to the different sizes of the three one-dimensional components of the image set. We have found that, for example, for microscopy image sets it is often advisable that the middle index k_2 corresponds to the optical axis, if fewer data points are available in this direction than in the others.

4. Results

In this section the previously presented algorithms will be illustrated with two examples. First, a simulated image set will be processed. This is followed by the analysis of a fluorescent microscopy image set.

4.1. *Simulation of Three-dimensional Image Sets*

We simulated an image set of three balls (one with radius 500 nm and two with radius 300 nm), since organelles in a cell often have a similar appearance. The images of balls were generated from an analytical description in the frequency domain [29]

$$H_3(w_1,w_2,w_3)=\frac{-2\pi R\sqrt{w_1^2+w_2^2+w_3^2}\cos(2\pi R\sqrt{w_1^2+w_2^2+w_3^2})+\sin(2\pi R\sqrt{w_1^2+w_2^2+w_3^2})}{\pi^2\sqrt{w_1^2+w_2^2+w_3^2}^3},$$

$w_1, w_2, w_3 \in \mathbb{R}$, where R is the radius of the ball. To avoid Gibbs effects the image set was convolved with a three-dimensional Gaussian filter whose standard deviation is 1 pixel. The simulated noise-free image set consists of 30 images each being a 60 × 60 pixel array. The images are assumed to be 50 *nm* apart along the optical axis, i.e. z-axis. Each pixel is assumed to be 50×50 nm in size. Zero mean Gaussian noise with standard deviation of 200 was added to each pixel of the image set to generate a noisy image set. All frames of the noisy image set are shown in Figure 1. The noise-free image set and the noisy image set are denoted as $P(k_1, k_2, k_3)$ and $P_n(k_1, k_2, k_3)$, $k_1 = 1, 2, \ldots, 60$, $k_2 = 1, 2, \ldots, 30$, $k_3 = 1, 2, \ldots, 60$, respectively. As discussed in Section 3.3, the second components of the image sets correspond to the z-axis/optical axis.

4.2. *Exact State Space Realizations of Simulated Image Sets*

In this section, we will show how to derive exact state space realizations of both the noisy image set P_n and the noise-free image set P using Algorithms 1 and 2. The first step is to perform the decomposition of the three-dimensional image sets via Algorithm 1, i.e. $P(k_1, k_2, k_3) = P^{0,0}(k_1, k_2, k_3) = P_1^0(k_1)P_2^0(k_2)P_3^0(k_3)$ and $P_n(k_1, k_2, k_3) = P_n^{0,0}(k_1, k_2, k_3) = P_{n;1}^0(k_1)P_{n;2}^0(k_2)P_{n;3}^0(k_3), k_1 = 1, 2, \ldots, 60$, $k_2 = 1, 2, \ldots, 30$, $k_3 = 1, 2, \ldots, 60$, where 0 implies no non-zero singular values are dropped in the algorithm (see Section 3.3 for an explanation of the notation). Key aspects of this step are two singular value decompositions used in steps 3 and 6 of Algorithm 1. The resulting singular values are shown in Figure 2. For the noise-free image set there are 55 and 59 non-zero singular values (we treat all singular values less than 10^{-16} as zeros) from the first and second singular value decompositions, respectively. For the noisy image set there are 60 non-zero singular values for both singular value decompositions, which are the maximum numbers, given the size of the image set. More non-zero singular values are obtained for the noisy image set because of the corruption caused by the additive Gaussian noise. To obtain exact estimates of the image sets we retain all non-zero singular values. A state space realization is then calculated for each of the one-dimensional components P_i^0, $P_{n;i}^0$, $i = 1, 2, 3$, of the image sets P, P_n, respectively, using Algorithm 2 and we obtain the balanced realizations $(A_i^{0;0}, B_i^{0;0}, C_i^{0;0})$ and $(A_{n;i}^{0;0}, B_{n;i}^{0;0}, C_{n;i}^{0;0})$ that are exact realizations of P_i^0 and $P_{n;i}^0$ respectively. Similarly, all non-zero singular values are retained in Algorithm 2.

In Figure 3 cross sections of the simulated image sets and their exact estimates are compared. It is clearly seen that the estimates are very accurate. The root-mean-square errors between the simulated image sets and their estimates are 4.7×10^{-11} in the noise-free case and 6.2×10^{-11} in the noisy case. This shows that up to small numerical errors we have indeed obtained exact realizations of both the noise-free image set and the noisy image set.

The memory requirement for calculating exact state space realizations is heavy. For the noise-free image set we have $P_1^0(k_1) \in \mathbb{R}^{1\times 55}$, $k_1 = 1, 2, \ldots, 60$, $P_2^0(k_2) \in \mathbb{R}^{55\times 59}$, $k_2 = 1, 2, \ldots, 30$, and $P_3^0(k_3) \in \mathbb{R}^{59\times 1}$, $k_3 = 1, 2, \ldots, 60$. Similarly for the

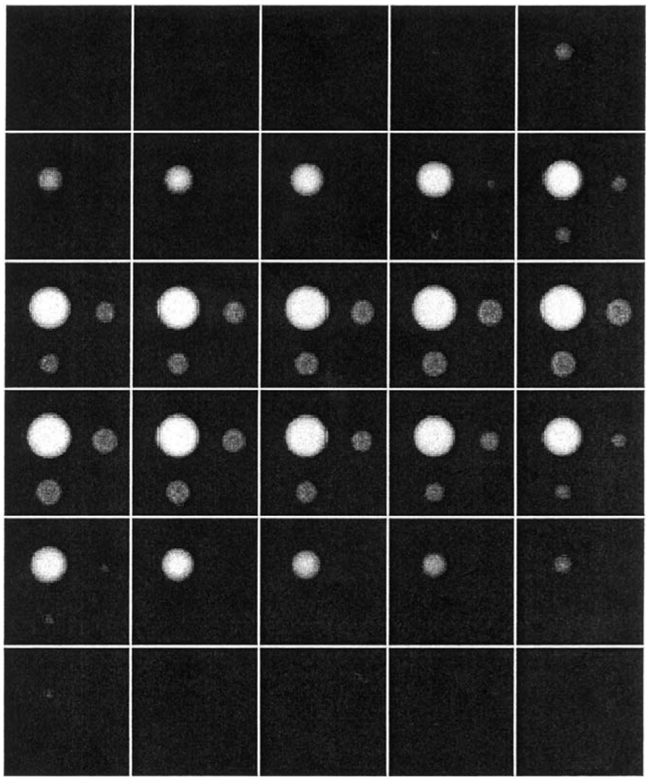

Figure 1. All two-dimensional images of the simulated noisy three-dimensional image set P_n consisting of three balls, one of radius 500 nm and two of radius 300 nm. The simulated image set consists of 30 images each being a 60×60 pixel array. The images are assumed to be 50 nm apart in the optical axis/z-axis. Each pixel is assumed to be 50×50 nm in size. Hence, each frame has a size of $3 \times 3\mu$m. Zero mean Gaussian noise with standard deviation 200 was added to each pixel. The panel is arranged such that the frames are displayed sequentially from left to right and top to bottom.

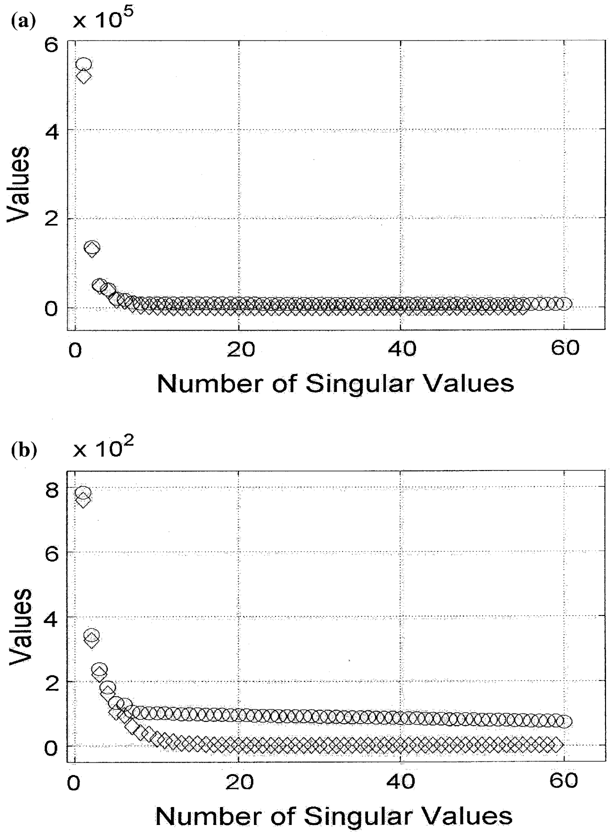

Figure 2. The singular values of the two singular value decompositions (A for the first singular value decomposition, B for the second singular value decomposition assuming all non-zero singular values are retained in the first decomposition) when the simulated noise-free image set P and noisy image set P_n (shown in Figure 1) are decomposed using Algorithm 1. The circles represent the singular values for the noisy simulated image set P_n while the diamonds represent the singular values for the noise-free image set P.

noisy image set we have $P^0_{n;1}(k_1) \in \mathbb{R}^{1\times 60}$, $k_1 = 1, 2, \ldots, 60$, $P^0_{n;2}(k_2) \in \mathbb{R}^{60\times 60}$, $k_2 = 1, 2, \ldots, 30$, and $P^0_{n;3}(k_3) \in \mathbb{R}^{60\times 1}$, $k_3 = 1, 2, \ldots, 60$. Clearly, the second components in both cases have the largest sizes. For the simulated noise-free image set, the size of the matrix $A^{0;0}_2$ is 1233×1233 while that of the matrix $A^{0;0}_{n;2}$ is 1795×1795 for the simulated noisy image set. In both cases, the sizes of the A matrices are bigger than the sizes of the corresponding image sets.

4.3. *Noise Suppression of the Simulated Image Set*

Having established that exact realizations can be obtained for the noise-free image set and the noisy image set we now consider the use of approximate realizations for

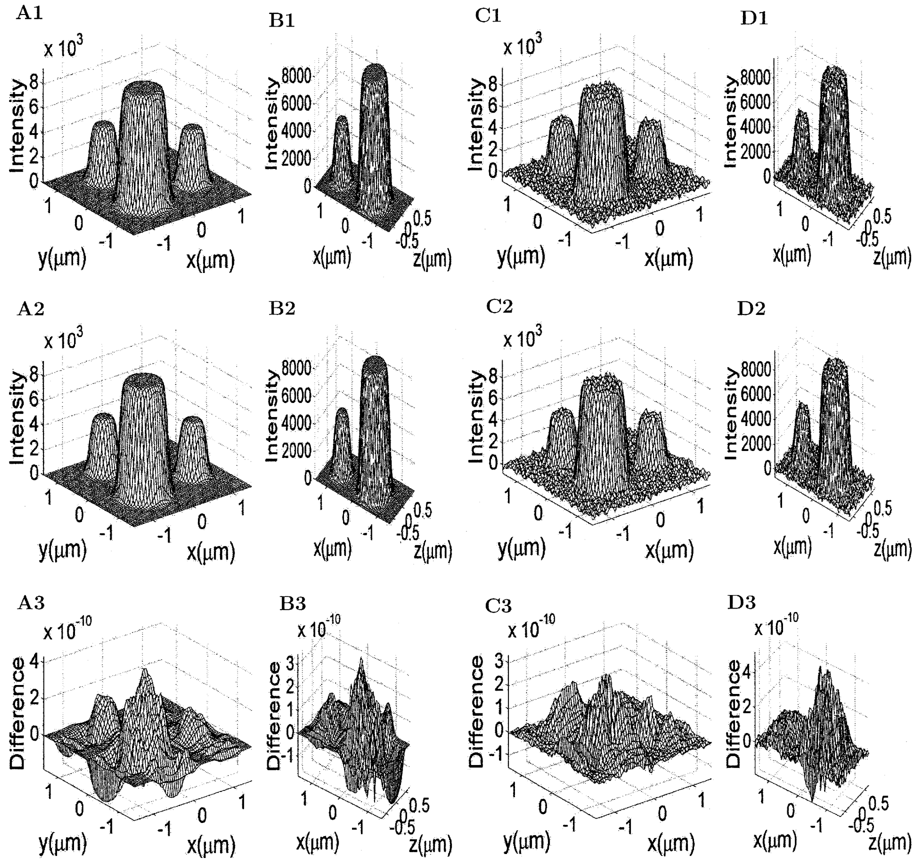

Figure 3. Cross sections of the simulated noise-free image set P and noisy image sets P_n (see Figure 1) and their exact estimates $P^{0,0;0,0,0}$, $P_n^{0,0;0,0,0}$, calculated via exact state space realizations. A1, A2 and A3 show the 16th frame $(P(k_1, 16, k_3), k_1, k_3 = 1, \ldots, 60)$ of the noise-free image set, the 16th frame $(P^{0,0;0,0,0}(k_1, 16, k_3), k_1, k_3 = 1, \ldots, 60)$ of its estimate based on the exact realizations and their difference respectively. B1, B2 and B3 show the cross section $P(22, k_2, k_3), k_2 = 1, \ldots, 30, k_3 = 1, \ldots, 60$, the cross section $P^{0,0;0,0,0}(22, k_2, k_3), k_2 = 1, \ldots, 30, k_3 = 1, \ldots, 60$, and their difference, respectively. C1, C2 and C3 show the 16th frame $(P_n(k_1, 16, k_3), k_1, k_3 = 1, \ldots, 60)$ of the noisy image set, the 16th frame $(P_n^{0,0;0,0,0}(k_1, 16, k_3), k_1, k_3 = 1, \ldots, 60)$ of its estimate based on the exact realizations and their difference respectively. D1, D2 and D3 show the cross section $P_n(22, k_2, k_3), k_2 = 1, \ldots, 30, k_3 = 1, \ldots, 60$, the cross section $P_n^{0,0;0,0,0}(22, k_2, k_3), k_2 = 1, \ldots, 30, k_3 = 1, \ldots, 60$, and their difference, respectively. The coordinate system shown in the plots is such that the y-axis (x-axis, z-axis/optical axis) coincides with the first (third, second) index of the image arrays $P, P_n, P^{0,0;0,0,0}$ and $P_n^{0,0;0,0,0}$, respectively. The origin of this coordinate system is located at the center of the pixel array(s).

noise reduction. In both Algorithms 1 and 2, approximations can be carried out and may result in a reduction of the noise components of the image set P_n. The sizes of the approximations are typically made dependent on the sizes of the retained sin-

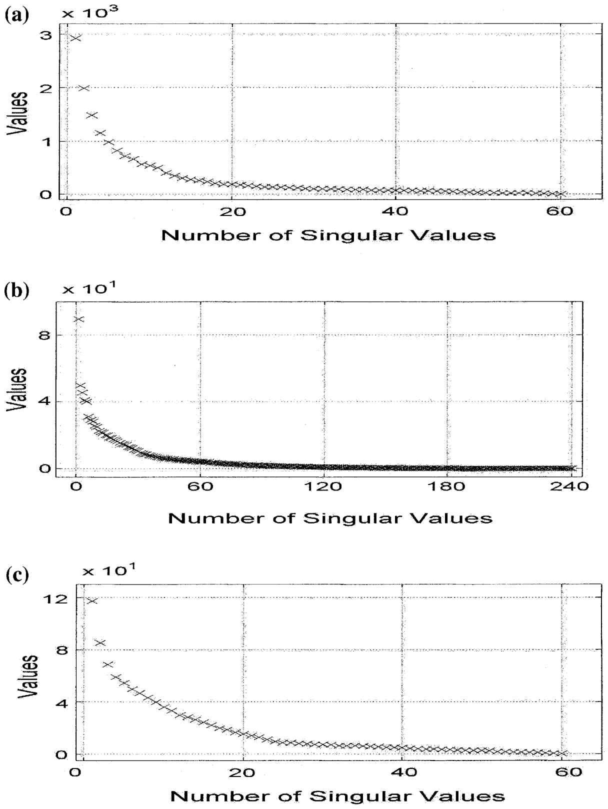

Figure 4. The singular values of the Hankel matrices constructed from the approximate one-dimensional components of the simulated noisy image set P_n (see Figure 1 and Section 4.3.) in Algorithm 2. The one-dimensional components are generated with 8 retained singular values from both singular value decompositions in Algorithm 1. A, B and C correspond to the first, second and third one-dimensional components, respectively.

gular values of the corresponding singular value decompositions. From Figure 2 it is seen only a few singular values of the noisy image set P_n are dominant and are close to the corresponding singular values of the noise-free image set P. The rest of the singular values of the noisy image set are very different from those of the noise-free image set because of the corruption from noise. Since only dominant singular values are important to the accuracy of the approximation we retained the eight largest singular values in each of the two singular value decompositions used to construct the one-dimensional components of the noisy image set. Following the notation introduced in Section 3.3 we represent three resultant one-dimensional components by $P^{52}_{n;1}$, $P^{52}_{n;2}$ and $P^{52}_{n;3}$, respectively, since in each of the two singular value decompositions 52 singular values are ignored. Figure 4 shows the singular values of the three Hankel matrices constructed from the one-dimensional components. Some of the singular values are relatively small and are therefore considered to correspond to insignificant parts of the dynamics. For the three one-dimensional realization problems we retain 53, 160 and 53 singular values, respectively. Since we drop 52, 52, 7, 80 and 7 singular values in both algorithms the smoothed estimate is then denoted by $P_n^{52,52;7,80,7}$ (see Section 3.3 for an explanation of the notation).

The sizes of the three one-dimensional approximate components $P_{n;1}^{52}$, $P_{n;2}^{52}$ and $P_{n;3}^{52}$ are $P_{n;1}^{52}(k_1) \in \mathbb{R}^{1\times 8}$, $k_1 = 1, 2, \ldots, 60$, $P_{n;2}^{52}(k_2) \in \mathbb{R}^{8\times 8}$, $k_2 = 1, 2, \ldots, 30$ and $P_{n;3}^{52}(k_3) \in \mathbb{R}^{8\times 1}$, $k_3 = 1, 2, \ldots, 60$. When a state space realization of the second component was calculated we dropped the 80 smallest singular values of the corresponding Hankel matrix. Following our notation in Section 3.3 we denote $(A_{n;2}^{52;80}, B_{n;2}^{52;80}, C_{n;2}^{52;80})$ as the approximate state space realization of the second component. The size of the matrix $A_{n;2}^{52;80}$ is 160 × 160. It is much smaller than $A_{n;2}^{0;0}$ (the A matrix for the corresponding exact realization). As a result, the memory requirement is reduced significantly.

Cross sections of the estimate $P_n^{52,52;7,80,7}$ of the noisy image set P_n are shown in Figure 5A1, A2, A3 and A4. The noise level is significantly reduced in the estimated image set. The root-mean-square error between the estimated image set and the simulated noise-free image set is 36.1, which is less than one fifth of the standard deviation of the Gaussian noise. It can, of course, not be expected that the noise-free image set P is fully recovered, since the result of the proposed method depends on the trade off between noise reduction on the one hand and the accuracy of the estimate on the other hand. Some errors are seen as low level ripples in regions of the images which have low signal levels. We have also observed that the best results are obtained by taking advantage of noise suppression in both Algorithms 1 and 2.

4.3.1. Performance of the Proposed Method Under Different Noise Levels

Table 1 shows the root-mean-square errors of the estimates obtained via the proposed method for different noise levels in the simulated data sets. The noise levels were varied by changing the standard deviation of the additive Gaussian noise during simulation. The table shows that the root-mean-square errors are less than 22% of the standard deviations of the noise in the corresponding noisy image set. This implies that noise is effectively suppressed. Clearly, by increasing the noise level in the simulated data set the accuracy of the estimates decreases. We obtain a corresponding increase in the root-mean-square error, since with increasing noise level

Table 1. The effectiveness of noise suppression via the proposed approach.

σ	50	100	200	300	400	500
RMSE	10.9	20.0	36.1	51.2	66.1	85.2

Root-mean-square errors between the smoothed image sets obtained via the proposed approach and the noise-free image set are calculated for different noise levels. The standard deviation of additive Gaussian noise is denoted as σ.

only the rough features of the image sets can be recovered. On the other hand, the ratios between the root-mean-square errors and the standard deviations of Gaussian noise decrease as the noise levels increase. This suggests that the proposed method suppresses noise effectively, even when the noise level is high.

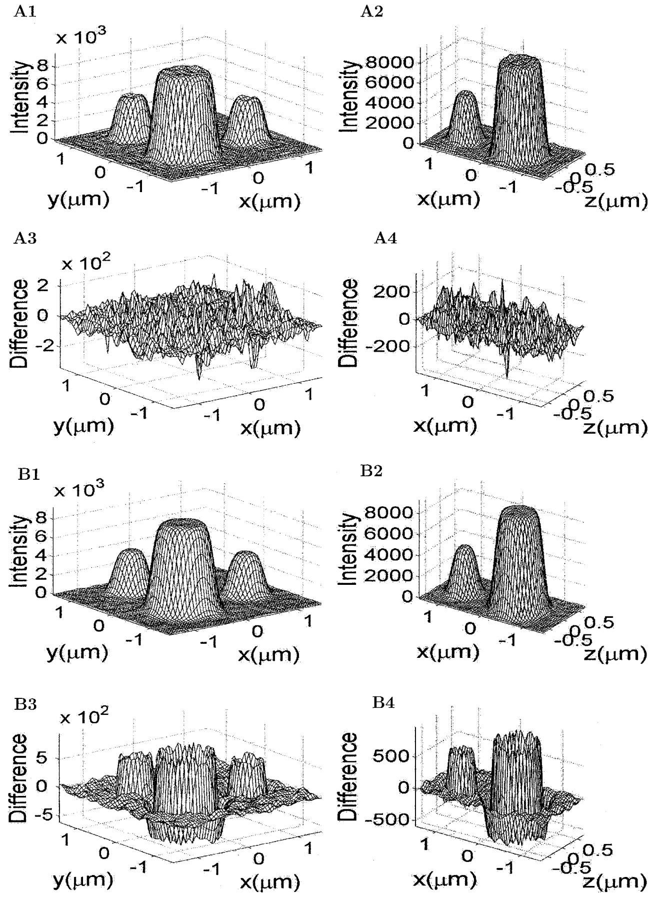

Figure 5. Cross sections of the estimates $P_n^{52,52;7,80,7}$ and P_n^g of the simulated noisy image set P_n, shown in Figure 3, by using the proposed method with noise suppression and a Gaussian filter with standard variance one, respectively. A1 and A3 show the 16th frame ($P_n^{52,52;7,80,7}(k_1, 16, k_3), k_1, k_3 = 1, \ldots, 60$) of the estimate obtained via the proposed method and its error, respectively. A2 and A4 show the cross section $P_n^{52,52;7,80,7}(22, k_2, k_3), k_2 = 1, \ldots, 30, k_3 = 1, \ldots, 60$ and its error, respectively. B1 and B3 show the 16th frame ($P_n^g(k_1, 16, k_3), k_1, k_3 = 1, \ldots, 60$) of the estimate obtained by using a Gaussian filter and its error, respectively. B2 and B4 show the cross section $P_n^g(22, k_2, k_3), k_2 = 1, \ldots, 30, k_3 = 1, \ldots, 60$ and its error, respectively. The coordinate system is defined as in Figure 3.

4.3.2. *Comparison Between the Results of the Proposed Method and the Smoothed Image sets Using a Gaussian Smoothing Filter*

A Gaussian filter of variance one is often used to suppress noise in fluorescence microscopy images (see e.g. [30], [31]). To compare the performance of the proposed method with the results of the Gaussian filter, we also smoothed the image set P_n with the Gaussian filter. Cross sections of the smoothed estimates of the noisy image set using both methods are shown in Figure 5. From these plots, it becomes evident that using a Gaussian filter results in much larger errors than the proposed approach. This is because the Gaussian filter calculates the weighted average of neighboring pixels and large errors occur at the edges of the balls. The root-mean-square error of the smoothed image set obtained by using the Gaussian filter is 151. This is more than four times the root-mean-square error (36.1) of the smoothed image set that is obtained via the proposed method.

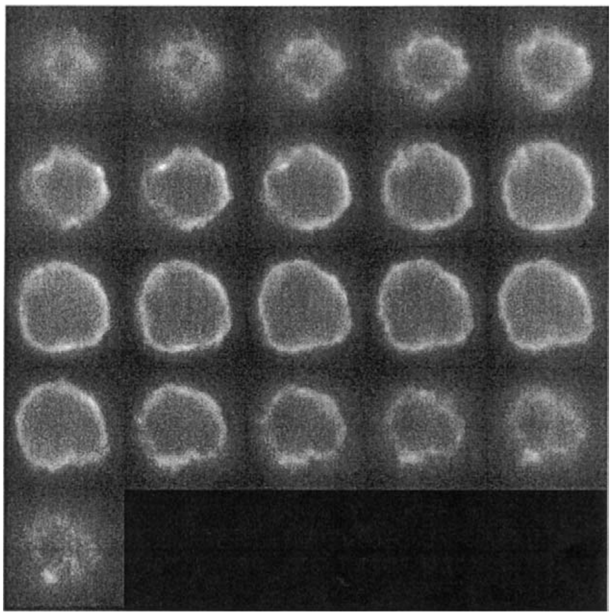

Figure 6. All two-dimensional images of a three-dimensional fluorescent microscopy image set Im_n of a T cell, showing the locations of FcRn. The images were acquired on a Zeiss inverted microscope with a 100× Plan-Aprochromat objective (N.A. = 1.4) using a highly sensitive Hamamatsu Orca 100 Peltier cooled 12 bit CCD camera. The image set consists of 21 images each being a 99 × 110 pixel array. The images are 300 nm apart from each other. Each pixel is 67 × 67 nm in size. Therefore, each frame has a size of 6.633 × 7.37 μm. The panel is arranged such that the frames are displayed sequentially from left to right and top to bottom.

4.4. *A Three-dimensional Fluorescent Microscopy Image Set*

We apply the proposed method to a three-dimensional fluorescent microscopy image set of a mouse T cell transfected with FcRn-GFP. The MHC Class I related receptor, FcRn, plays a role in the transfer of gammaglobulin (IgG) from mother to young and also regulates the serum levels of IgG, reviewed in [32]. The large number of molecules and the limited resolution capacity of the microscope imply that only the total accumulations of the signals are recognizable. The image set consists of 21 images, 300 nm apart from each other. Each image has 99 × 110 pixels and the size of a pixel is 67 × 67 nm in the object space of the imaging system. This means that the image set forms an array $Im_n(k_1, k_2, k_3), k_1 = 1, 2, \ldots, 99, k_2 = 1, 2, \ldots, 21, k_3 = 1, 2, \ldots, 110$. As in the previous data the second index of the image set refers to

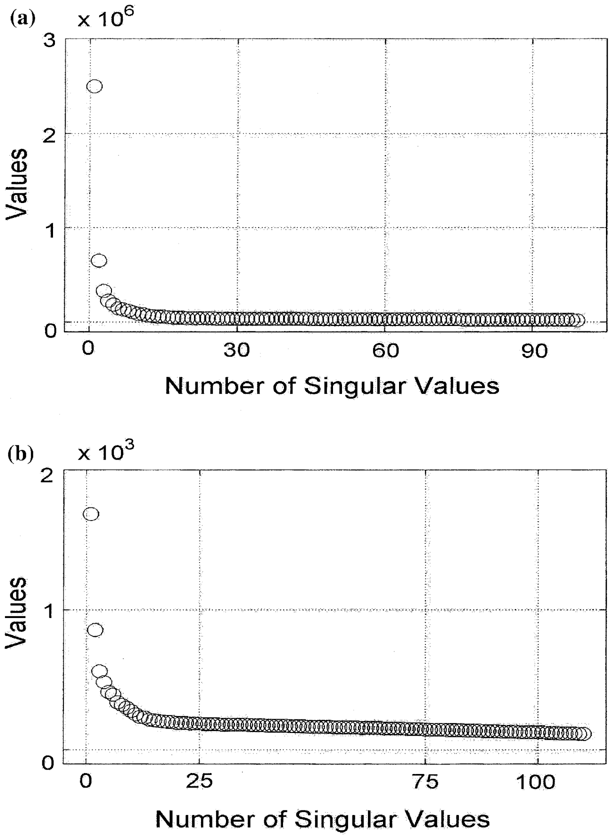

Figure 7. The singular values of the two singular value decompositions (A for the first singular value decomposition, B for the second singular value decomposition assuming all non-zero singular values are retained in the first decomposition) when the three-dimensional image set Im_n of a T cell, shown in Figure 6, is decomposed using Algorithm 1.

the optical axis of the microscope, i.e. the z-axis. All images of this three-dimensional image set are shown in Figure 6.

4.5. Exact Estimate of a Fluorescent Microscopy Image Set via State Space Realizations

Following the same procedure for the simulated data, we decomposed the image set Im_n into three one-dimensional components via Algorithm 1 and calculated an exact

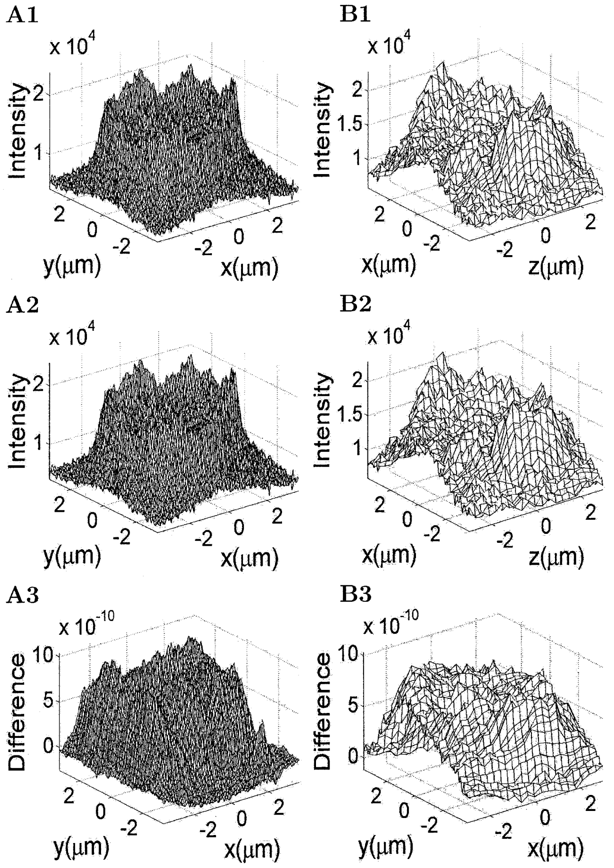

Figure 8. Cross sections of the fluorescent image set Im_n of a T cell, shown in Figure 6, and its estimate $Im_n^{0,0;0,0,0}$ calculated via the exact state space realizations. A1, A2 and A3 show the 12th frame $(Im_n(k_1, 12, k_3), k_1 = 1, \ldots, 99, k_3 = 1, \ldots, 110)$ of the image set Im_n, the 12th frame $(Im_n^{0,0;0,0,0}(k_1, 12, k_3), k_1 = 1, \ldots, 99, k_3 = 1, \ldots, 110)$ of the estimate calculated via the exact state space realizations and their difference, respectively. B1, B2 and B3 show the cross section $Im_n(60, k_2, k_3), k_2 = 1, \ldots, 21, k_3 = 1, \ldots, 110$, the cross section $Im_n^{0,0;0,0,0}(60, k_2, k_3), k_2 = 1, \ldots, 21, k_3 = 1, \ldots, 110$, and their difference, respectively. The coordinate system is defined as in Figure 3

balanced state space realization for each of the one-dimensional components. Figure 7 shows the plots of the singular values for the two singular value decompositions used in Algorithm 1. Similar to the simulated data, all singular values are non-zero due to the presence of noise. The singular values obtained in Algorithm 2 are also non-zero. To obtain the exact realizations we retained all singular values. The exact estimate of the image set is denoted as $Im_n^{0,0;0,0,0}$.

Figure 8 shows two cross sections of the image set Im_n and the corresponding cross sections in the estimated image set $Im_n^{0,0;0,0,0}$ calculated from the exact state space realizations. The root-mean-square error between the estimated image set and the acquired image set is 7.4×10^{-10}. This shows that up to insignificant numerical errors exact state space realizations were obtained. It is interesting to note that the significant background intensity level has also been accurately matched by the proposed method.

As in the case of the simulated image set, large memory allocations are necessary, if the exact state space realizations are calculated. Let $Im_{n,1}^0$, $Im_{n,2}^0$ and $Im_{n,3}^0$ be the three one-dimensional components of Im_n calculated using Algorithm 1. We have $Im_{n,1}^0(k_1) \in \mathbb{R}^{1\times 99}$, $k_1 = 1, 2, \ldots, 99$, $Im_{n,2}^0(k_2) \in \mathbb{R}^{99\times 110}$, $k_2 = 1, 2, \ldots, 21$, and $Im_{n,3}^0(k_3) \in \mathbb{R}^{110\times 1}$, $k_3 = 1, 2, \ldots, 110$. Let $(A_2^{0;0}, B_2^{0;0}, C_2^{0;0})$ be the exact state space realization obtained for the second one-dimensional component. The size of this system is considerable with $A_2^{0;0}$ being a 2079×2079 matrix.

4.6. Noise Suppression of a Fluorescent Microscopy Image Set

Image sets acquired in fluorescent microscopy are notoriously noisy, since the fluorescent signal is relatively weak. Noise sources include scattered photons, contamination in the samples, and noise sources in the cameras etc. Therefore noise suppression is an important aspect in the analysis of fluorescent microscopy image sets. Here, we will apply the proposed method to suppress the noise components in the image set Im_n.

From Figure 7 only a few singular values are dominant amongst the singular values of the two singular value decompositions that are used in Algorithm 1. We retain 15 and 30 singular values in the first and the second singular value decompositions, respectively. Figure 9 shows the singular values of the three Hankel matrices constructed from the corresponding one-dimensional components. For the calculation of the approximate realizations we retain $60, 300$ and 100 singular values, respectively. The other singular values are very small and judged not to be significant enough to include in the approximation. The numbers of the dropped singular values in the five singular value decompositions are $84, 80, 39, 15$ and 10. Following the notation introduced in Section 3.3, we denote the estimated image set as $Im_n^{84,80;39,15,10}$.

Figure 10 shows cross sections of the estimated image set $Im_n^{84,80;39,15,10}$. The difference between the smoothed image set and the original image set appears as random noise. Some small ripples appear to have been created during noise suppression and they are observed near the background area of the estimated images.

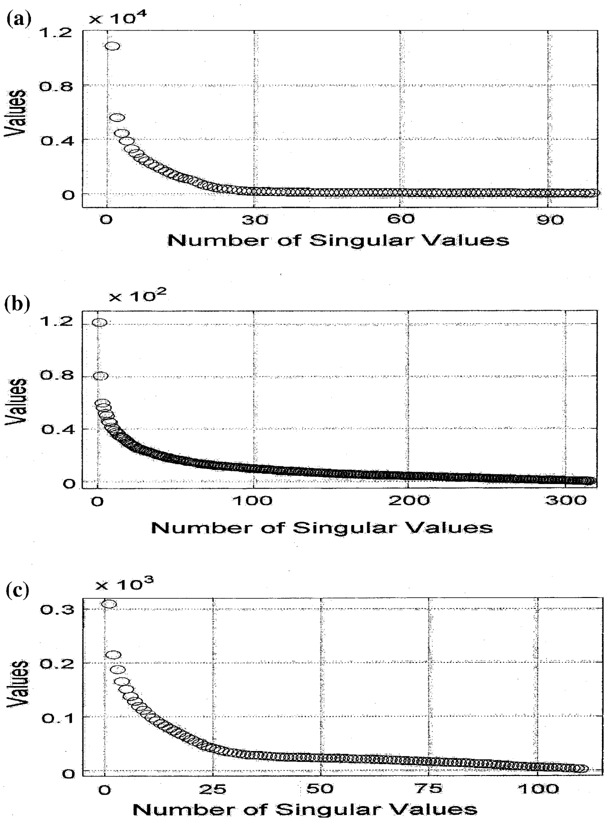

Figure 9. The singular values of the Hankel matrices constructed from the approximate one-dimensional components of the image set Im_n of a T cell (Figure 6) in Algorithm 2. The one-dimensional components are generated with 15 and 30 retained singular values from the first and second singular value decompositions in Algorithm 1. A, B and C correspond to the first, second and third one-dimensional components, respectively.

As expected the memory requirement for the approximate state space realizations is much smaller than that for the exact realizations. For example, the matrix $A_2^{80;15}$ for the second approximate one-dimensional component of Im_n is of size 300 × 300, which is considerably smaller than the 2079 × 2079 of the corresponding A matrix for the exact realization.

It should be pointed out that we found that it is of advantage to estimate the background level of the image set separately, if approximate realizations are to be obtained. This is particularly relevant in situations such as the present one when the background level is significant since relatively small errors in the estimation of the background could lead to relatively large errors in the estimation of the actual signal

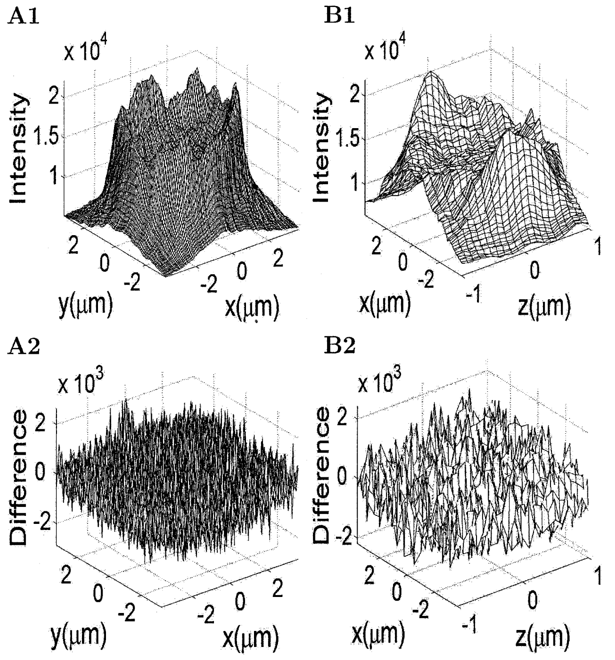

Figure 10. Cross sections of the smoothed estimate $Im_n^{84,80;39,15,10}$ of the fluorescent image set Im_n, shown in Figure 6. A1 and A2 show the 12th frame $(Im_n^{84,80;39,15,10}(k_1, 12, k_3), k_1 = 1, \ldots, 99, k_3 = 1, \ldots, 110)$ of the estimate and the difference between it and the 12th frame of Im_n, respectively. B1 and B2 show the cross section $Im_n^{84,80;39,15,10}(60, k_2, k_3), k_2 = 1, \ldots, 21, k_3 = 1, \ldots, 110$, and the difference between it and the corresponding cross section of Im_n, respectively.

that is of interest. To avoid this potential error we reduced the background level of the image set by subtracting an estimate of the background level and then applied the proposed method to the modified image set. The estimated background level was formed by the mean of the pixels at the edges of the image set and was added back to the resulting smoothed image set.

5. Conclusions

We developed a method to calculate state space realizations of a three-dimensional image set by decomposing the three-dimensional image set into three one-dimensional components and constructing a state space realization for each of these components. Besides the capability to obtain exact state space realizations of the image set the method allows for the incorporation of approximation steps that can be used for noise reduction in the image set.

We tested the approach with both simulated data and a three-dimensional fluorescent microscopy image set. In both cases, the estimates obtained via the exact state space realizations were very accurate. However, the proposed method is memory-intensive, when the exact state space realizations are calculated. It has been shown that the proposed method suppressed the noise components of the image sets effectively and compared favorably to the use of a Gaussian filter, especially with respect to the preservation of sharp features in the image set.

Acknowledgements

The authors thank Carlos Vaccaro for preparing the biological sample. This work was supported in part by grants by the National Institutes of Health (R01 A150747 and R01 A142949).

References

1. S. Inoue and K.R. Spring, *Video Microscopy: The Fundamentals,* Plenum Pub Corp., 1997.
2. D.A. Agard, "Optical Sectioning Microscopy: Cellular Architecture in Three Dimensions," *Annual Review of Biomedical Engineering,* 1984, vol. 13, pp. 191–219.
3. D.A. Agard, Y. Hiraoka, and J.W. Sedat, "Three-dimensional Microscopy: Image Processing for High Resolution Subcellular Imaging," *SPIE,* vol. 61, 1989, pp. 24–30.
4. G.M.P. van Kempen and L.J. van Vliet, "The Influence of the Regularization Parameter and the First Estimate on the Performance of Tikhonov Regularized Non-linear Image Restoration Algorithms," *Journal of Microscopy,* vol. 198, no. 1, 2000, pp. 63–75.
5. G.M.P. van Kempen, L.J. van Vliet, P.J. Verveer, and H.T.M. van der Voort, "A Quantitative Comparison of Image Restoration Methods for Confocal Microscopy," *Journal of Microscopy,* vol. 185, 1997, pp. 354–365.
6. B. Roysam, A.K. Bhattacharjya, C. Srinivas, D.H. Szarowski, and J.N. Turner, "Unsupervised Noise Removal Algorithms for Three-dimensional Confocal Fluorescence Microscopy," *Micron and Microscopica Acta,* vol. 23, no. 4, 1992, 447–461.
7. A. Doi and T. Hinamoto, "A Spatial-domain Technique for the Design of 3-D Separable-denominator State-space Digital Filters," *Multidimensional Systems and Signal Processing,* vol. 12, 2001, pp. 89–98.
8. T. Hinamoto and A. Doi, "Design of Multidimensional Separable-denominator Digital Filters in the Spatial Domain," in *Proceedings of IEEE Asia Pacific Conference on Circuits and Systems,* Seoul, Korea, 1996, pp. 219–222.
9. T. Hinamoto, A. Doi, and K. Harada, "Design of Multidimensional Separable-denominator Digital Filters in the Spatial Domain," *Multidimensional Systems and Signal Processing,* vol. 8, 1997, pp. 273–293.
10. K. Hirano, M. Sakane, and M.Z. Mulk, "Design of Three-dimensional Recursive Digital Filters," *IEEE Transactions on Circuits and Systems,* vol. CAS-31, no. 6, 1984, pp. 550–561.
11. B. Lashgari, L.M. Silverman, and J.F. Abramatic, "Approximation of 2-D Separable in Denominator Filters," *IEEE Transactions on Circuits and Systems,* vol. CAS-30, no.2, 1983, pp. 107–121.
12. R.J. Ober, "Balanced Canonical Forms," *in Identification; Adaptation; Learning,* eds. S. Bittanti and G. Picci, Berlin: Springer-Verlag, 1996, pp. 120–179.

13. M. Green and D.J.N. Limebeer, *Linear Robust Control,* New Jersey: Prentice-Hall Inc., 1995.
14. K. Zhou, J.C. Doyle, and K. Glover, *Robust Optimal Control,* New Jersey: Prentice-Hall Inc., 1996.
15. W.S. Lu and A. Antoniou, *Two-dimensional Digital Filters,* New York: Marcel Dekker Inc., 1992.
16. P.S.K. Hansen, P.C. Hansen, S.D. Hansen, and J.A. Sorensen, "Experimental Comparison of Signal Subspace Based Noise Reduction Methods," in *Proceedings of International Conference on Acoustics Speech and Signal Processing,* vol. 1, 1999, pp. 101–104.
17. T. Lin, M. Kawamata, and T. Higuchi, "Design of 2-D Separable-denominator Digital Filters Based on the Reduced-dimensional Decomposition," *IEEE Transactions on Circuits and Systems,* vol. CAS-34, no. 8 1987, pp. 934–941.
18. T. Lin, M. Kawamata, and T. Higuchi, "Decomposition of 2-D Separable-denominator Systems: Existence Uniqueness and Applications," *IEEE Transactions on Circuits and Systems,* vol. CAS-34, no. 3, 1987, pp. 292–296.
19. T. Kailath, *Linear Systems,* New Jersey: Prentice-Hall Inc., 1980.
20. B. Moore, "Principal Component Analysis in Linear Systems: Controllability Observability and Model Reduction," *IEEE Transactions on Automatic Control,* vol. AC-26, no. 1, 1981, pp. 17–32.
21. S.Y. Kung, "A New Identification and Model Reduction Algorithm via Singular Value Decompositions," in *Proceedings of the 12th Asilomar Conference on Signals, Systems and Computers,* 1978, pp. 705–714.
22. H.P. Zeiger and A.J. McEwen, "Approximate Linear Realizations of Given Dimension via Ho's Algorithm," *IEEE Transactions on Automatic Control,* vol. AC-19, no. 153, 1974.
23. J.M. Maciejowski, "Guaranteed Stability with Subspace Methods," *Systems and Control Letters,* vol. 26, 1995, pp. 153–156.
24. T. Kailath, A.H. Sayed, B. Hassibi, *Linear Estimation,* New Jersey: Prentice-Hall Inc., 2000.
25. L. Pernebo and L.M. Silverman, "Model Reduction via Balanced State Space Representations," *IEEE Transactions on Automatic Control,* vol. Ac-27, no. 2, 1982 pp. 382–387.
26. W.W.F. Pijnappel, A. van den Boogaart, R. de Beer, and D. van Ormondt, "SVD-Based Quantification of Magnetic Resonance Signals," *Journal of Magnetic Resonance,* vol. 97, 1992, pp. 122–134.
27. R.J. Ober and E.S. Ward, "Correcting for Phase Distortion of NMR Spectra Analyzed Using Singular-value Decomposition of Hankel Matrices," *Journal of Magnetic Resonance,* vol. A 114, 1995, pp. 120–123.
28. R.J. Ober, J. Caves, and E.S. Ward, "Analysis of Exponential Data Using a Noniterative Technique: Application to Surface Plasmon Experiments," *Analytical Biochemistry,* vol. 312, 2003, pp. 57–65.
29. L.J. van Vliet, "*Grey-Scale Measurements in Multi-Dimensional Digitized Images,*" Ph.D. thesis, University of Delft, 1993.
30. H. Chen, J.R. Swedlow, M. Grote, and J.W. Sedat, and D.A. Agard, "The Collection Processing and Display of Digital Three-Dimensional Images of Biological Specimens," in *Handbook of Biological Confocal Microscopy,* ed. J. B. Pawley, 2nd edition, New York: Plenum Press 1995. 197–210.
31. G.M.P. van Kempen and L.J. van Vliet, "Improving the restoration of textured objects with prefiltering," in *Proceedings of 3rd Annual Conference of the Advanced School for Computing and Imaging (ASCI'97)* 1997, pp. 174–179.
32. V. Ghetie and E.S. Ward, "Multiple Roles for the Major Histocompatibility Complex Class I Related Receptor FcRn," *Annual Review of Immunology,* vol. 18, 2000, pp. 739–766.