

# Sum-of-exponentials modeling and common dynamics estimation using Tensorlab

I. Markovsky\*, O. Debals\*\*, and L. De Lathauwer\*\*\*\*

\* *Department ELEC*  
*Vrije Universiteit Brussel (VUB)*  
*Building K, Pleinlaan 2, B-1050*  
*Brussels, Belgium*  
*Ivan.Markovsky@vub.ac.be*

\*\* *ESAT/STADIUS and iMinds Medical IT*  
*Katholieke Universiteit Leuven*  
*Kasteelpark Arenberg 10*  
*3001 Leuven, Belgium*  
*Otto.Debals@esat.kuleuven.be*

\*\*\* *Katholieke Universiteit Leuven Campus Kortrijk*  
*Group Science, Engineering and Technology*  
*E. Sabbelaan 53, 8500 Kortrijk Belgium*  
*Lieven.DeLathauwer@kuleuven-kulak.be*

---

**Abstract:** Fitting a signal to a sum-of-exponentials model is a basic problem in signal processing. It can be posed and solved as a Hankel structured low-rank matrix approximation problem. Subsequently, local optimization, subspace, and convex relaxation methods can be used for the numerical solution. In this paper, we show another approach, based on the recently developed concept of structured data fusion. Structured data fusion problems are solved in the Tensorlab toolbox by local optimization methods. The approach allows fitting of signals with missing samples and adding constraints on the model, such as fixed exponents and common dynamics in multi-channel estimation problems. These problems are non-trivial to solve by other existing methods. Tensorlab is publicly available and the results presented are reproducible.

*Keywords:* System identification; low-rank approximation; mosaic Hankel matrix; tensorlab; structured data fusion.

---

## 1. INTRODUCTION

First, we define the notation and the basic sum-of-exponentials modeling problem. Then, we list some generalization of the basic problem and state contribution of the paper in the context of the current state-of-the-art methods.

### 1.1 Sum-of-exponentials modeling

A *discrete-time exponential signal* is a sequence

$$(\dots, \exp_z(-1), \exp_z(0), \exp_z(1), \dots), \quad \exp_z(t) := z^t.$$

Given a set of  $n$  complex numbers  $z = (z_1, \dots, z_n)$ , the  $n$ th order *sum-of-exponentials model*  $\mathcal{B}$  is defined as the set (Willems, 1986, 1987; Polderman and Willems, 1998; Markovsky *et al.*, 2006)

$$\mathcal{B} = \mathcal{B}(z) = \left\{ \sum_{j=1}^n c_j \exp_{z_j} \mid c \in \mathbb{C}^n \right\}. \quad (1)$$

The model  $\mathcal{B}$  is real if all signals  $y \in \mathcal{B}$  are real valued. In what follows, we consider discrete-time real models.

The sum-of-exponentials model (1) is an autonomous linear time-invariant dynamical system (Luenberger, 1979; Kailath, 1981; Sontag, 1998; Markovsky *et al.*, 2006):

$$\mathcal{B} = \mathcal{B}(A, C) = \{y \mid \sigma x = Ax, y = Cx\}, \quad (2)$$

where  $\sigma$  is the shift operator

$$(\sigma x)(t) := x(t+1)$$

the model parameters are defined as

$$A = \text{diag}(z_1, \dots, z_n), \quad C = [1 \ \dots \ 1], \quad (3)$$

and the initial condition is

$$x(0) = [c_1 \ \dots \ c_n]^T.$$

Alternatively,  $y \in \mathcal{B}(z)$  is the impulse response of the linear time-invariant system

$$\begin{aligned} \mathcal{B} &= \mathcal{B}(A, B, C, D) \\ &= \{y \mid \sigma x = Ax + Bu, y = Cx + Du\}, \end{aligned}$$

where  $A$  and  $C$  are as defined in (3), and

$$B = [c_1 \ \dots \ c_n]^T, \quad D = y(0).$$

The restriction of the model  $\mathcal{B}$  on the interval  $[1, T]$  is denoted by  $\mathcal{B}|_T$ . In the modelling problem, we are given a trajectory

$$y = (y(1), \dots, y(T)) \in \mathcal{B}|_T.$$

It can be shown that

$$\mathcal{B}(z)|_T = \text{image}(V_T(z)),$$

where  $V_T(z)$  is the (transposed) Vandermonde matrix

$$V_T(z) = \begin{bmatrix} z_1^1 & \dots & z_n^1 \\ \vdots & & \vdots \\ z_1^T & \dots & z_n^T \end{bmatrix}.$$

Therefore, for any  $y \in \mathcal{B}(z)|_T$ , there exists a  $c \in \mathbb{C}^n$ , such that

$$y = V_T(z)c. \quad (4)$$

The distance between a signal  $y$  and a system  $\mathcal{B}$  is measured by the projection of  $y$  on  $\mathcal{B}$

$$\text{dist}(y, \mathcal{B}) := \min_{\hat{y} \in \mathcal{B}} \|y - \hat{y}\|_2, \quad (5)$$

Here  $\|\cdot\|_2$  is the Frobenius norm or 2-norm, defined as the square root of the sum-of-squares of the elements of the vector, matrix or tensor.

With this notation, the sum-of-exponentials modeling problem is defined as follows.

*Problem 1.* Given a time series  $y$  and order  $n$ , find a model  $\mathcal{B}(z)$  of order at most  $n$ , that is as close as possible to the data  $y$  in the 2-norm sense, *i.e.*,

$$\text{minimize } \text{dist}(y, \mathcal{B}(z)) \quad \text{over } z \in \mathbb{C}^n. \quad (6)$$

This problem can be viewed as an autonomous linear time-invariant system identification problem or else as the identification of a linear time-invariant system from impulse response data, called the realization problem (Kalman, 1979; Ho and Kalman, 1966; Silverman, 1971; Markovsky, 2013). More specifically, problem (6) defines a maximum likelihood estimator in the output error setting, assuming additive zero mean, stationary, white Gaussian noise (Markovsky, 2008; Pintelon and Schoukens, 2012).

## 1.2 Literature review and contributions of the paper

The basic sum-of-exponentials modeling problem (6) is defined in the literature, see, *e.g.*, (Kumaresan and Tufts, 1982; Stoica and Moses, 2005), for one scalar time series  $y$  (single channel sum-of-exponentials modeling). Recently, the following generalizations have been considered:

- (1) the data  $y$  is a vector valued time series (a multi channel sum-of-exponentials modeling),
- (2) the data  $y$  consists of  $N \geq 1$  time-series with possibly different lengths  $T_1, \dots, T_N$  (multiple experiments sum-of-exponentials modeling) (Markovsky and Pintelon, 2015),
- (3) the data  $y$  may contain missing samples, and

The subspace methods (Van Overschee and De Moor, 1996) which are based on the state space representation (2) are trivial to generalize to the multi channel case as well as the case of multiple time-series with equal lengths  $T_1 = \dots = T_N$ . However, these methods are non-trivial to generalize for multiple time-series with different lengths as well as to data with missing data. A subspace method for sum-of-exponentials modeling with missing data is proposed in (Markovsky, 2016).

Another approach for sum-of-exponentials modeling is based on reformulation of the problem as a Hankel structured low-rank approximation problem (Markovsky, 2008; Markovsky,

2012). The methods are based on local optimization (Marquardt, 1963; Nocedal and Wright, 1999; Golub and Pereyra, 2003) and allow efficient computation (linear computational complexity in the number of samples). These methods have been recently generalized to multiple time series (Markovsky and Pintelon, 2015) and missing data (Markovsky and Usevich, 2013).

The contribution of this paper is in reformulation of the sum-of-exponentials modeling problem as a structured data function problem. The resulting solution methods are also based on local optimization and allow solution of problems with multiple time series and missing data. In addition, the structured data fusion approach to sum-of-exponentials modeling problem allows a solution of a related problem of common dynamics modeling (Papy *et al.*, 2006), see Section 4.4.

## 2. SUM-OF-EXPONENTIALS MODELING USING TENSORLAB

### 2.1 Introduction of Tensorlab

Tensorlab is a MATLAB toolbox offering algorithms for tensor decompositions, complex least-squares optimization, global minimization and more (Sorber *et al.*, 2014). Note that tensors are higher-order generalizations of vectors and matrices, denoted in this paper by calligraphic letters. Structured data fusion is recently proposed as a framework within Tensorlab for rapid prototyping of (coupled) decompositions of (whether dense, incomplete or sparse) tensors with constraints on the factor matrices (Sorber, 2014; Sorber *et al.*, 2013). Structured data fusion is implemented in Tensorlab by using a domain specific language, enabling the user to choose from various factor structures and from three different tensor decompositions: the canonical polyadic decomposition, the block term decomposition (De Lathauwer, 2008) or the low-multilinear rank approximation.

In this paper, tensor decompositions are being used for solving the vector and matrix problems in equation (4). Indeed, the data sets need not necessarily be higher-order. Instead of finding (structured) factor matrices  $U^{(1)}$ ,  $U^{(2)}$  and  $U^{(3)}$  to decompose a tensor  $\mathcal{T}$ , we search factor matrices  $U^{(1)}$  and  $U^{(2)}$  to factorize a matrix  $T$  into  $U^{(1)}U^{(2)\top}$ . Likewise, we can search for a matrix  $U$  and vector  $c$  (both generally known as ‘factors’) to write a known vector  $t$  as  $t = Uc$ . Of course, multiple solutions exist for the latter two factorizations. Uniqueness only appears through additional constraints on the factors, unlike for tensor decompositions realizing uniqueness under mild conditions (De Lathauwer, 2011; Domanov and De Lathauwer, 2013a; Domanov and De Lathauwer, 2013b; Kruskal, 1977). These constraints can be enforced using structured data fusion of which the usefulness becomes apparent from the generalizations proposed in Section 4.

### 2.2 Structured data fusion

Constraints/structures on the factor matrices are typically (non)linear dependencies on some underlying variables  $z$ . These underlying variables can be scalars, vectors or matrices. They are transformed to factors through smooth mappings, *i.e.*, the partial derivatives with respect to the elements of the underlying variables and their complex conjugates should exist and be continuous up to all orders (continuous Wirtinger derivatives). Next, the factors are assigned to each dataset through a

chosen decomposition. Tensorlab implements a following least-squares optimization:

$$\min_z f = \sum_{d=1}^D \omega_d f^{(d)} \quad (7)$$

$$\text{with } f^{(d)} = \frac{1}{2} \left\| \underbrace{\mathcal{M}^{(d)}(\mathcal{X}(z)) - \mathcal{F}^{(d)}}_{\mathcal{F}^{(d)}} \right\|_{\mathcal{W}^{(d)}}^2,$$

where  $\|\mathcal{A}\|_{\mathcal{W}} := \|\mathcal{W} * \mathcal{A}\|$  with  $*$  the Hadamard (element-wise) product. In (7),  $z$  is the set of underlying variables,  $\mathcal{X}$  is the collection of the mappings (with the factors as the images),  $\mathcal{F}^{(d)}$  is the  $d$ th dataset, and  $\mathcal{M}^{(d)}$  is the model of the decomposition corresponding to each dataset  $\mathcal{F}^{(d)}$ . For example, regarding the canonical polyadic decomposition, we have the following model:

$$\mathcal{M}_{\text{CPD}}(U^{(1)}, \dots, U^{(N)}) := \sum_{r=1}^R u_r^{(1)} \otimes \dots \otimes u_r^{(N)}. \quad (8)$$

Without constraints, the matrices  $U^{(i)}$  belong to  $\mathcal{X}(z)$ . Missing elements from an incomplete tensor are associated with a zero in the corresponding position in the weighting tensor  $\mathcal{W}^{(d)}$ , then known as the observation tensor. Details about the algorithms used and their implementation are given in the next section.

### 3. SOLUTION METHOD

#### 3.1 Algorithms for structured data fusion

We only discuss the nonlinear least squares algorithm, which makes use of first-order derivative information but is able to achieve up to second-order convergence by approximating the second-order derivatives. With  $z$  we denote a vector collecting the underlying variables. By approximating the residual tensors  $\mathcal{F}^{(d)}$  from (7) with a linear model

$$m_k^{\mathcal{F}^{(d)}}(p) := \mathcal{F}^{(d)}(z_k) + J_k^{(d)} \cdot \overset{c}{p}, \quad \text{where } J_k^{(d)} := \partial \mathcal{F}^{(d)} / \partial \overset{c}{z}^\top,$$

one obtains a second-order model of the following form:

$$m_k^f(p) := f(z_k) + \overset{c}{p}^\top \cdot \frac{\partial f}{\partial \overset{c}{z}}(z_k) + \frac{1}{2} \overset{c}{p}^* \cdot B_k \cdot \overset{c}{p}. \quad (9)$$

Eq. (9) is a second-order complex Taylor series expansion of  $f$  from (7) around  $z_k$ , and it is minimized to obtain the steps  $p$  to the next iterate  $z_{k+1}$ . The critical work is to calculate the complex gradient  $\partial f / \partial \overset{c}{z}$  and the matrix  $B_k$  which is a positive semidefinite approximation of the complex Hessian described by  $\partial^2 f / (\partial \overset{c}{z} \partial \overset{c}{z}^\top)$ . In the complex gradient, both the cogradient  $\partial f / \partial z$  and conjugate cogradient  $\partial f / \partial \bar{z}$  are stacked. They are conjugates of each other as  $f$  is real. Typically, one only uses the conjugate cogradient.

We further subdivide these components to separate the decomposition models  $\mathcal{M}^{(d)}$  from the structure imposed by  $\mathcal{X}$ . The mappings  $\mathcal{X}$  are assumed to be analytical functions, i.e., they are not dependent on the conjugate of  $z$ . For non-analytic mappings  $\mathcal{X}$ , we direct the reader to (Sorber *et al.*, 2012; Sorber *et al.*, 2013). The chain rule on the conjugated cogradient can be applied to isolate the derivatives regarding the mappings (Sorber *et al.*, 2012):

$$\frac{\partial f}{\partial \bar{z}} = \sum_{d=1}^D w_d \left( \left( \frac{\partial \text{vec}(\mathcal{X})}{\partial \text{vec}(z)^\top} \right)^* \cdot \frac{\partial f^{(d)}}{\partial \text{vec}(\mathcal{X})} \right). \quad (10)$$

The left-hand factor is related to the mapping and the right-hand factor is related to the decompositions applied. Both can be calculated analytically beforehand, independent of each other.

For the Hessian's approximation, one can write

$$B_k = \left( \sum_{d=1}^D w_d \left( \frac{\partial \text{vec}(\mathcal{X})}{\partial \text{vec}(z)^\top} \right)^* \cdot J_k^{(d)*} J_k^{(d)} \right) \cdot \frac{\partial \text{vec}(\mathcal{X})}{\partial \text{vec}(z)^\top}, \quad (11)$$

with the Jacobian

$$J_k^{(d)} = \left( \partial \text{vec}(\mathcal{F})^{(d)} / \partial \text{vec}(\mathcal{X})^\top \right)$$

and

$$J_k^{(d)*} J_k^{(d)}$$

called the (Jacobian's) Gramian. Note that this Gramian can be computed very efficiently because of the highly-structured  $J_k^{(d)}$ ; hence, we omit the dot in between.

The first-order information of the mappings can be defined beforehand by deriving two matrix-vector products analytically:

$$\left( \frac{\partial \text{vec}(\mathcal{X})}{\partial \text{vec}(z)^\top} \right)^* \cdot l$$

and

$$\frac{\partial \text{vec}(\mathcal{X})}{\partial \text{vec}(z)^\top} \cdot r.$$

The latter can be used for equation (11) while the former can be used for both the equations (10) and (11).

#### 3.2 Implementation details

In this section, we elaborate on the implementation for the technique described in Section 3. To use custom mappings  $\mathcal{X}(z)$  in Tensorlab, only an evaluation method and the two previously described matrix-vector products have to be provided. We discuss the latter for the mappings used in this paper.

The first transformation used is a Vandermonde mapping

$$V_d(z) : z, d \rightarrow V, \quad \text{with } (V)_{ij} = z_i^{d_j}.$$

For general vectors  $z \in \mathbb{C}^Z, d \in \mathbb{R}^D$  and given vectors  $l \in \mathbb{C}^{ZD}, r \in \mathbb{C}^Z$  we have the following matrix-vector products while supposing  $1 \leq i \leq Z$  and  $1 \leq j \leq D$ :

$$\left( \left( \frac{\partial \text{vec}(V_d(z))}{\partial \text{vec}(z)^\top} \right)^* \cdot l \right)_i = \sum_{j=1}^D l_{i+j\tilde{Z}} \overline{d_j z_i^{d_j-1}},$$

$$\left( \frac{\partial \text{vec}(V_d(z))}{\partial \text{vec}(z)^\top} \cdot r \right)_{i+j\tilde{Z}} = r_i d_j z_i^{d_j-1},$$

with  $\tilde{Z} = Z - 1$ . The returned vectors are of sizes  $Z$  and  $ZD$ , respectively. In this paper, we have  $d = [1, 2, \dots, T]$ .

Another transformation used (in Section 4.4 for common dynamics) is the Hadamard product between a constant matrix and a matrix containing variables:  $H_A(Z) : Z, A \rightarrow A \odot Z$ . For general matrices  $Z \in \mathbb{C}^{I \times J}, A \in \mathbb{C}^{I \times J}$  and given vectors  $l \in \mathbb{C}^{IJ}, r \in \mathbb{C}^{IJ}$  we have:

$$\frac{\partial \text{vec}(H_A(Z))}{\partial \text{vec}(z)^\top} \cdot l = \text{vec}(\bar{A} \odot l),$$

$$\frac{\partial \text{vec}(H_A(Z))}{\partial \text{vec}(z)^\top} \cdot r = \text{vec}(A \odot r).$$

The initial approximation for  $z$  and  $C$  is obtained by Kung's algorithm (Kung, 1978) and a mask matrix  $M$  is used to enforce zero pattern of  $C$  in case of common poles estimation.

#### 4. GENERALIZATIONS OF THE SUM-OF-EXPONENTIALS MODELING PROBLEM

In Section 4.1 we generalize problem (6) to the multi-channel problem. Multiple trajectories are introduced in Section 4.2. Tensorlab can be used to incorporate missing data, as discussed in Section 4.3. Finally, Section 4.4 discusses the presence of common dynamics, which has a strong connection to blind signal separation (De Lathauwer, 2011).

##### 4.1 Multi-channel modeling

In the multi-channel sum-of-exponential modeling problem, the given data  $y$  is a  $p$ -dimensional vector time series. Each output measurement  $y_i$  is a trajectory of the same model  $\mathcal{B}(z)$ , so that, in general, the exponents  $z_1, \dots, z_n$  are common to all channels. Eq. (4) gives us a structured matrix factorization which can be seen as a structured data fusion problem:

$$[y_1 \cdots y_p] = V_T(z) \underbrace{[c_1 \cdots c_p]}_C. \quad (12)$$

##### 4.2 Modeling using multiple trajectories

If the data consists of  $N$  time series

$$y^i = (y^i(1), \dots, y^i(T_i))$$

generated from the same model  $\mathcal{B}(z)$ , the modeling problem is to approximate all time series in a least-squares sense:

$$\text{minimize } \sum_{i=1}^N \text{dist}^2(y^i, \mathcal{B}(z)) \text{ over } z \in \mathbb{C}^n. \quad (13)$$

Note that modeling using multiple trajectories is different from the multi-channel modeling because the lengths  $T_1, \dots, T_N$  of the time series may be different. In addition, it is possible to consider multi-channel modeling with multiple trajectories. Problem (13) is solved in Tensorlab using a coupled factorization with common parameters  $z$ :

$$y^i = V_T(z)c^i, \quad \text{for } i = 1, \dots, N.$$

##### 4.3 Missing data

Missing data values  $y(t)$ , for  $t \in \mathcal{I}_m$  are handled by excluding them from the calculation of the distance measure, *i.e.*, (5) is calculated only with respect to the given data. In Tensorlab missing values are specified by setting the corresponding elements to NaN. The initial approximation is computed by

- (1) replacing the missing data with the average of the given data, and
- (2) applying Kung's method on the complete data sequence. Another approach for computing initial approximation is to use the nuclear norm heuristic (Fazel, 2002).

##### 4.4 Common dynamics modeling

The common dynamics estimation problem in multi-channel signal processing (Papy *et al.*, 2006) is aiming to decompose the dynamics of the channels into "common dynamics", *i.e.*, joint exponents, and "individual dynamics", *i.e.*, exponents that appear in a specific channel only. A zero-pattern structure on the matrix  $C$  from (12) is imposed. To illustrate we have

$$C = \begin{bmatrix} C_{0,1} & \cdots & C_{0,p} \\ C_{1,1} & & \\ & \ddots & \\ & & C_{p,p} \end{bmatrix}, \quad (14)$$

with all missing values being zeros. Let  $n_0 := \text{rowdim}(C_{0,1})$  and  $n_i := \text{rowdim}(C_{i,1})$ . The first  $n_0$  exponents are common to all channels, the next  $n_1$  exponents are used in the first channel only, and so on. Such a constraint can be easily imposed in Tensorlab by a Hadamard product of a fully parameterized matrix variable and a mask matrix, *i.e.*, a matrix with ones at the position of the nonzero elements in (14) and zeros elsewhere. The implementation is discussed in Section 3.2.

## 5. NUMERICAL RESULTS

In this section we compare the structured data fusion method for sum-of-exponentials modeling (implemented in the function `sem_tensorlab`) with a system identification method (implemented in the function `ident`) based on a formulation of the problem as a structured low-rank approximation problem, see (Markovsky, 2013; Markovsky and Usevich, 2014). The data is generated by a second order system with poles  $0.5 \pm 0.5i$ , referred to as the *true system*  $\mathcal{B}$ . Zero mean, white noise with variance  $s^2$  is added to a true output signal  $\bar{y} \in \mathcal{B}$ ,

$$y = \bar{y} + \tilde{y}, \quad \text{where } \tilde{y} \sim \mathcal{N}(0, s^2).$$

The noise variance  $s^2$  is chosen so that the signal-to-noise ratio is equal to 0.25.

We repeat the identification experiment multiple times with different noise realizations and report the mean estimation errors for the compared methods. The obtained models  $\hat{\mathcal{B}}^i$  in the  $i$ th repetition are compared with respect to the following criteria:

- distance from the data

$$\text{dist}(y, \hat{\mathcal{B}}^i);$$

- parameter error

$$\|\bar{\theta} - \hat{\theta}^i\|,$$

where  $\bar{\theta}$  is the normalized true model's  $\mathcal{B} = \mathcal{B}(\bar{z})$  parameter vector

$$\bar{\theta} = [\bar{\theta}_0 \ \bar{\theta}_1 \ \cdots \ \bar{\theta}_{n-1}]$$

$$(z - \bar{z}_1) \cdots (z - \bar{z}^n) =$$

$$\bar{\theta}_0 + \bar{\theta}_1 z + \cdots + \bar{\theta}_{n-1} z^{n-1} + z^n$$

and  $\hat{\theta}^i$  are defined similarly for the identified models  $\hat{\mathcal{B}}^i$ ;

- execution time.

In addition, in case of missing data estimation, the

- missing data estimation error  $\|\bar{y}|_{\mathcal{I}_m} - \hat{y}^i|_{\mathcal{I}_m}\|_2$

criterion is used.

In what follows we present results obtained for the following experiments:

- Single-channel modeling, see Table 1.
- Multi-channel modeling, see Table 2.
- Multiple trajectories, see Table 3.
- Missing data, see Tables 4 and 5.
- Common dynamics, see Table 6.
- Fixed exponents, see Table 7.

In the single-channel, multi-channel, and multiple trajectories problems the results obtained by the `ident` and `tensorlab` methods coincide (up to the convergence tolerance of the optimization solvers). This is the desired empirical confirmation that the two methods solve the sample problem. Note, however, that due to the nonconvexity of the problem it is not guaranteed that the different local optimization solvers will converge to the same locally optimal solution. This, however, is the case in our simulation examples.

The computation time for the `ident` method is smaller than the one for the `tensorlab`. The reason for this is the `ident` is based on the structured low-rank approximation package which is implemented in C while Tensor lab is entirely written in MATLAB and version 2.0, used in this empirical study, is not optimized for structured data fusion.

The discrepancy in the results obtained with `ident` and `tensorlab` in the common dynamics and fixed exponents simulations is due to the fact that the prior information of common dynamics and fixed exponents, which enters as constraints in the optimization problem is not taken into account by the `ident` method. Thus, the fitting error  $\text{dist}(y, \hat{\mathcal{B}})$  obtained by `ident` is smaller but the parameter error  $\|\hat{\theta} - \hat{\theta}\|$  is smaller for the `tensorlab` method. Of course, we are ultimately of interest in the error  $\|\hat{\theta} - \hat{\theta}\|$  of estimating the true parameter value. Therefore, the simulation results confirm the value of using relevant prior knowledge.

Table 1. Single-channel modeling

	h2ss	ident	tensorlab
$\text{dist}(y, \hat{\mathcal{B}})$	0.1593	0.15882	0.15882
$\ \hat{\theta} - \hat{\theta}\ $	0.067918	0.081972	0.081972
time, sec	0.021258	0.00703	0.39417

Table 2. Multi-channel modeling

	h2ss	ident	tensorlab
$\text{dist}(y, \hat{\mathcal{B}})$	0.24562	0.24398	0.24398
$\ \hat{\theta} - \hat{\theta}\ $	0.032581	0.082582	0.082643
time, sec	0.18418	0.073227	0.34834

Table 3. Multiple trajectories

	h2ss	ident	tensorlab
$\text{dist}(y, \hat{\mathcal{B}})$	0.5537	0.53823	0.53823
$\ \hat{\theta} - \hat{\theta}\ $	0.05024	0.010513	0.010685
time, sec	0.16975	0.072516	3.6245

Table 4. Missing data with a failing sensor.

	h2ss	ident	tensorlab
$\text{dist}(y, \hat{\mathcal{B}})$	0.48181	0.43176	0.4295
$\ \hat{y}_{\mathcal{I}_m} - \hat{y}_{\mathcal{I}_m}\ _2$	0.54959	2.6637	2.8703
$\ \hat{\theta} - \hat{\theta}\ $	0.068502	0.0046522	0.0051387
time, sec	0.23586	5.5531	129.04

Table 5. Missing data with every 3rd sample missing.

	h2ss	ident	tensorlab
$\text{dist}(y, \hat{\mathcal{B}})$	0.1544	0.11186	0.086098
$\ \hat{y}_{\mathcal{I}_m} - \hat{y}_{\mathcal{I}_m}\ _2$	0.186	0.14223	0.28437
$\ \hat{\theta} - \hat{\theta}\ $	0.30408	0.13358	0.26602
time, sec	0.22507	0.096979	7.5459

Table 6. Common dynamics

	h2ss	ident	tensorlab
$\text{dist}(y, \hat{\mathcal{B}})$	0.010412	0.010411	0.01167
$\ \hat{\theta} - \hat{\theta}\ $	0.0071362	0.0071068	0.0051559
time, sec	0.24239	0.081743	0.42571

Table 7. Fixed exponents

	h2ss	ident	tensorlab
$\text{dist}(y, \hat{\mathcal{B}})$	0.0070297	0.0068503	0.0070182
$\ \hat{\theta} - \hat{\theta}\ $	2.2719	0.53264	0.0041439
time, sec	0.084715	0.036707	0.18954

## 6. CONCLUSIONS

In this paper, we considered the basic sum-of-exponentials modeling problem and its generalizations to multi-channel modeling, multiple time series, missing data, modeling with known exponents, common dynamics identification, and fixed poles. All these problems were solved by the structured data fusion functionality of the Tensor toolbox. The solution approach using structured data fusion is based on local optimization methods and requires user specified initial approximation. Simulation results comparing the Tensor toolbox with the variable projections method for structured low-rank approximation approach show equivalent results in terms of accuracy. The current implementation of the structured data fusion in the Tensor toolbox is simple to generalize to new problems but is slower than the alternative methods.

## ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement number 258581 "Structured low-rank approximation: Theory, algorithms, and applications" and ERC Advanced Grant agreement number 339804 "BIOTENSORS". It is also funded by a Ph.D. grant of the Agency for Innovation by Science and Technology (IWT), by the Research Council KU Leuven: CoE PFV/10/002 (OPTEC), by F.W.O.: projects G.0830.14N and G.0881.14N, G028015N "Decoupling multivariate polynomials in nonlinear system identification"; G090117N "Block-oriented nonlinear identification using Volterra series"; and by the Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO II, Dynamical systems, control and optimization, 2012–2017).

## REFERENCES

- De Lathauwer, L. (2008). Decompositions of a higher-order tensor in block terms — Part II: Definitions and uniqueness. *SIAM Journal on Matrix Analysis and Appl.* **30**(3), 1033–1066.
- De Lathauwer, L. (2011). Blind separation of exponential polynomials and the decomposition of a tensor in rank- $(L_r, L_r, 1)$  terms. *SIAM Journal on Matrix Analysis and Appl.* **32**(4), 1451–1474.
- Domanov, I. and L. De Lathauwer (2013a). On the uniqueness of the canonical polyadic decomposition of third-order tensors — Part I: Basic results and uniqueness of one factor matrix. *SIAM Journal on Matrix Analysis and Applications* **34**(3), 855–875.
- Domanov, I. and L. De Lathauwer (2013b). On the uniqueness of the canonical polyadic decomposition of third-order

- tensors — Part II: Uniqueness of the overall decomposition. *SIAM Journal on Matrix Analysis and Applications* **34**(3), 876–903.
- Fazel, M. (2002). Matrix rank minimization with applications. PhD thesis. Elec. Eng. Dept., Stanford University.
- Golub, G. and V. Pereyra (2003). Separable nonlinear least squares: the variable projection method and its applications. *Institute of Physics, Inverse Problems* **19**, 1–26.
- Ho, B. L. and R. E. Kalman (1966). Effective construction of linear state-variable models from input/output functions. *Regelungstechnik* **14**(12), 545–592.
- Kailath, T. (1981). *Linear Systems*. Prentice Hall.
- Kalman, R. E. (1979). On partial realizations, transfer functions, and canonical forms. *Acta Polytechnica Scandinavica* **31**, 9–32.
- Kruskal, J. B. (1977). Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Appl.* **18**(2), 95–138.
- Kumaresan, R. and D. Tufts (1982). Estimating the parameters of exponentially damped sinusoids and pole-zero modeling in noise. *IEEE Trans. Acoust., Speech, Signal Process.* **30**(6), 833–840.
- Kung, S. (1978). A new identification method and model reduction algorithm via singular value decomposition. In: *Proc. 12th Asilomar Conf. Circuits, Systems, Comp.* pp. 705–714.
- Luenberger, D. G. (1979). *Introduction to Dynamical Systems: Theory, Models and Applications*. John Wiley.
- Markovsky, I. (2008). Structured low-rank approximation and its applications. *Automatica* **44**(4), 891–909.
- Markovsky, I. (2012). *Low Rank Approximation: Algorithms, Implementation, Applications*. Springer.
- Markovsky, I. (2013). A software package for system identification in the behavioral setting. *Control Eng. Practice* **21**(10), 1422–1436.
- Markovsky, I. (2016). On the most powerful unfalsified model for data with missing values. *Systems & Control Lett.* **95**, 53–61.
- Markovsky, I. and K. Usevich (2013). Structured low-rank approximation with missing data. *SIAM J. Matrix Anal. Appl.* **34**(2), 814–830.
- Markovsky, I. and K. Usevich (2014). Software for weighted structured low-rank approximation. *J. Comput. Appl. Math.* **256**, 278–292.
- Markovsky, I. and R. Pintelon (2015). Identification of linear time-invariant systems from multiple experiments. *IEEE Trans. Signal Process.* **63**(13), 3549–3554.
- Markovsky, I., J. C. Willems, S. Van Huffel and B. De Moor (2006). *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. SIAM.
- Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.* **11**, 431–441.
- Nocedal, J. and S. Wright (1999). *Numerical optimization*. Springer-Verlag.
- Papy, J.-M., L. De Lathauwer and S. Van Huffel (2006). Common pole estimation in multi-channel exponential data modeling. *Signal Processing* **86**(4), 846–858.
- Pintelon, R. and J. Schoukens (2012). *System Identification: A Frequency Domain Approach*. second ed.. IEEE Press. Piscataway, NJ.
- Polderman, J. and J. C. Willems (1998). *Introduction to Mathematical Systems Theory*. Springer-Verlag. New York.
- Silverman, L. M. (1971). Realization of linear dynamical systems. *IEEE Trans. Automat. Contr.* **16**, 554–567.
- Sontag, E. D. (1998). *Mathematical control theory: Deterministic finite dimensional systems*. Springer-Verlag.
- Sorber, L. (2014). Data fusion: Tensor factorizations by complex optimization. PhD thesis. Faculty of Engineering, KU Leuven. Leuven, Belgium.
- Sorber, L., M. Barel and L. Lathauwer (2012). Unconstrained optimization of real functions in complex variables. *SIAM Journal on Optimization* **22**(3), 879–898.
- Sorber, L., M. Van Barel and L. De Lathauwer (2013). Structured data fusion. Internal report 13–177, ESAT-SISTA, KU Leuven (Leuven, Belgium).
- Sorber, L., M. Van Barel and L. De Lathauwer (2014). Tensorlab v2.0. Available online. URL: <http://www.tensorlab.net>.
- Stoica, P. and R. Moses (2005). *Spectral Analysis of Signals*. Prentice Hall.
- Van Overschee, P. and B. De Moor (1996). *Subspace identification for linear systems: Theory, implementation, applications*. Kluwer. Boston.
- Willems, J. C. (1986, 1987). From time series to linear system—Part I. Finite dimensional linear time invariant systems, Part II. Exact modelling, Part III. Approximate modelling. *Automatica* **22**, **23**, 561–580, 675–694, 87–115.