

# Tensor decomposition and forecasting for multivariate time series

Kirill Semkin<sup>1\*</sup> and Vadim Strijov<sup>2\*</sup>

<sup>1</sup>\*Forecsys® Moscow Russia.

<sup>2</sup>\*Forecsys® Moscow Russia.

\*Corresponding author(s). E-mail(s): [kirill.semkin32@mail.ru](mailto:kirill.semkin32@mail.ru);  
[strijov@ccas.ru](mailto:strijov@ccas.ru);

**Keywords:** time series, decomposition, forecast, SSA, CPD

Processing of multidimensional time series is associated with an additional task of determining dependencies between signals. Its inclusion in models boosts the quality of forecasts. On the other hand, taking this dependency into account makes models more complex and less interpretable. The paper proposes a non-parametric method based on tensor data representation and Singular Spectrum Analysis (SSA). It derives a time series decomposition technique and an explicit forecast model. Finally, it applies an elaborated theory to electricity consumption and inertial measurement unit datasets. It compares the obtained quality of forecast with mSSA, VAR, and RNN models.

# 1 Introduction

The main object of the paper is multivariate time series. It is a set of  $m$  time series  $\{x_i(t)\}_{i=1}^m$  observed in discrete time  $t \in 1, \dots, N$ . We state two problems concerning them. First, to make a *forecast* is to estimate future values  $\{x_i(T)\}_{i=1}^m$  of the time series at time  $T > N$ . Second, to build an additive *decomposition* is to represent each signal in a set as a sum of several components:  $x_i(t) = c_1(t) + \dots + c_s(t)$ ,  $\forall i \in 1, \dots, m$ .

For the decomposition in a single-variate time series case, the papers [1–3] introduce seasonal-trend-cycle techniques. For the forecast, the authors in [4] propose autoregressive methods as well as works [1, 5] suggest exponential smoothing, regression models, and neural networks. However, these methods can not be transferred to the multivariate time series straightforwardly if they are interdependent. The paper [6] models the size of prey-and-predator populations as a coupled system of differential equations. The dynamics of one population depends directly on the other's and vice-versa. Therefore, the forecast can not be made separately.

Some methods both take into account multivariate time series interdependence and make the forecast. First, *recurrent neural network* (RNN) [7] connects time series and their past values through several layers of nonlinear transformations. Information from the past values is encapsulated in the hidden state vector at each time step. It is used to forecast future values [8].

Second, the *vector autoregression* (VAR) [9, 10] is a linear stochastic model for multivariate time series. Denote vector  $\mathbf{x}_t = (x_1(t) \dots x_m(t))^\top$  as a time series realizations at time  $t$ , their further dynamics

$$\mathbf{x}_t = \boldsymbol{\mu} + \sum_{i=1}^p A_i \mathbf{x}_{t-i} + \mathbf{u}_t.$$

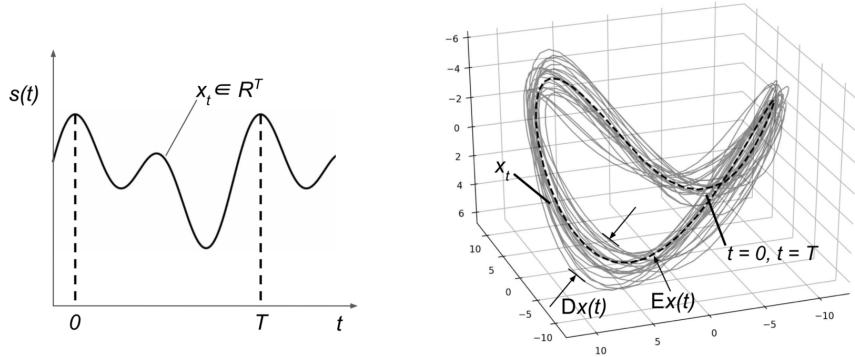
Here  $\mu$  is a constant vector,  $A_i$  are  $m \times m$  matrices,  $\mathbf{u}_t$  is a random vector (e.g. white noise  $WN(t)$ ). The time series interdependence is defined by  $A_i$  matrices, so each time series depends on each other and their past linearly.

The RNN and VAR methods are able to make the forecast, but, firstly, have a large number of learnable and structural parameters. That means their underlying models have an extensive complexity. Their architecture needs to be picked over and tuned. Secondly, the methods' structure does not contain an explicit way of building time series decomposition.

To rectify the listed disadvantages, we have developed a new method called the *tensor SSA* (tSSA). It has only two adjustable parameters. The tSSA extends the SSA method [11] for the multivariate time series and implies using dynamical systems theory. The Takens theorem [12] is the theoretical foundation of the tSSA. *Hankel* matrices and their properties are involved in the decomposition problem. It is a matrix with equal elements on each anti-diagonal. Such matrices are tightly connected with Toeplitz and circulant matrices.

The main task of the paper is to find the shared *phase subspace* [13, 14] for all observed time series. This is a low-dimensional linear subspace where all the system's trajectories can be mapped into isomorphically. Finding it enables us to obtain phase representation of the observed time series. On the Fig. 1 (left) we use single time series  $s(t)$  to build phase space and the specific trajectory  $\mathbf{x}_t$  of the dynamical system that generated the given time series, Fig. 1 (right). The forecast then is found upon the fact that the continuation of the trajectory lies in the very same phase space. Partitioning basis vectors of this space into several groups builds the decomposition of the time series. The paper develops named conceptions in the next sections.

Our method captures autocorrelation and multilinearity of the time series via tensor data representation. *Canonical polyadic decomposition* (CPD) is used to build the phase space of the time series. Another solution is to use matrix data representation



**Figure 1** Time series's phase space visualization. Single time series  $s(t)$  builds phase trajectory  $\mathbf{x}_t$  in a 3-dimensional phase space

and matrix decompositions. Such method is called matrix SSA (mSSA) [15]. It is the second extension of the SSA method for the forecast and the decomposition of the multivariate time series. The tSSA is compared with the mSSA in the computational experiment section.

The rest of the paper covers the theory and application of our method to real data. Firstly, the dynamical systems model is introduced and the problem of basis search in the phase space is stated and resolved. These results enable us to propose a way of decomposing the time series and to make the forecast. Simultaneously, features of obtained solutions are examined and two associated theorems are formulated. Finally, tSSA and mentioned methods are applied to two datasets: electricity consumption and accelerometer/gyroscope observations. We obtain their forecast and additive decompositions. For the latter, a special quality metric is introduced.

## 2 Problem statement

Suppose a *dynamical system* defined on  $X$ , with the evolution function  $f$  and the initial state  $\mathbf{y}_0$ . Time space is discrete

$$\mathbf{y}(t+1) = f(\mathbf{y}(t)), \quad t \in \mathbb{N},$$

$$\mathbf{y}(0) = \mathbf{y}_0.$$

Generally,  $X$  is a high-dimensional smooth manifold. Then unknown mapping  $\phi : X \rightarrow \mathbb{R}^m$  takes each trajectory point  $\mathbf{y}(t)$  to the multivariate time series realization  $\mathbf{x}_t$

$$\phi(\mathbf{y}(t)) = \mathbf{x}_t. \text{ Coordinateley: } \begin{cases} \phi_1(\mathbf{y}(t)) = x_1(t), \\ \dots \\ \phi_m(\mathbf{y}(t)) = x_m(t). \end{cases}$$

Assume that the trajectories  $\mathbf{y}(t)$  belong to the smaller dimension manifold  $M \subset X$ . Now the problem is to find an embedding of the  $M$  into  $\mathbb{R}^L$  for some  $L$ . The second problem is to build a basis in the image of the embedding. Having done that, the initial dynamical system will be described in terms of the standard linear space. The same will hold to all time series  $x_i(t)$ .

### 3 Single-variate time series case

The tSSA relies on the SSA method and the Takens theorem. The theorem provides the needed embedding in the single-variate time series case. Any point  $\mathbf{y}(t) \in M$  corresponds to the following vector:

$$[\phi \circ f^{t-L+1}(\mathbf{y}(t)), \dots, \phi \circ f(\mathbf{y}(t)), \phi \circ \mathbf{y}(t)]^\top = [x(t-L+1), \dots, x(t-1), x(t)]^\top.$$

It is called the *delay vector* at time  $t$  and is denoted as  $\overleftarrow{\mathbf{x}}_t$ . The vector's dimension  $L$  must satisfy condition  $L > 2 \cdot \dim(M)$ . The function  $\phi(\cdot)$  must satisfy several regularity conditions. They are omitted here and considered fulfilled.

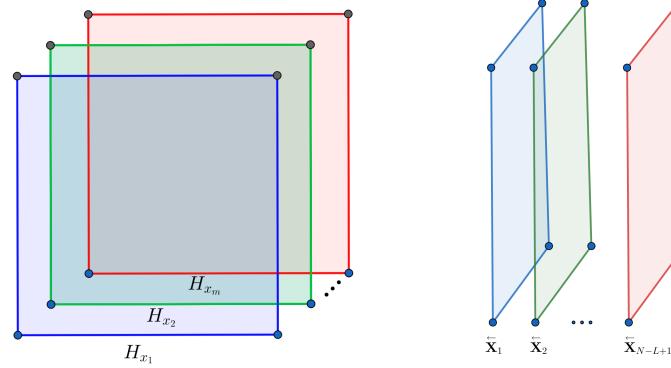
Now the time series  $x(t)$  with the length  $N$  gives  $N - L + 1$  delay vectors. Therefore image of the embedding is  $\text{Lin}(\{\overleftarrow{\mathbf{x}}_t\})$ . It represents the *individual* phase space of the

time series and must be low dimensional. This means  $\text{Lin}(\{\overleftarrow{\mathbf{x}}_t\}) \subset \mathbb{R}^L$ . Finally, the orthonormal basis is the  $U$ -component of the Singular value decomposition (SVD) on the *trajectory matrix*  $H_x$ . It is a stacking of the delay vectors

$$H_x = [\overleftarrow{\mathbf{x}}_1 \dots \overleftarrow{\mathbf{x}}_{N-L+1}].$$

## 4 Multivariate time series case and the tSSA method

Now there given multivariate time series. Let  $m$  be their number. Hence, the  $\phi(\cdot)$  is a multidimensional map. This means that instead of the single delay vector we have  $m$  of them at any time  $t$ . Let  $\overleftarrow{X}_t$  be the matrix  $[\overleftarrow{\mathbf{x}}_{1,t} \dots \overleftarrow{\mathbf{x}}_{m,t}]$  called a *delay matrix*. Then, the embedding image is a linear span of all  $\overleftarrow{X}_t$ . It is also called a *shared* phase space of the multivariate time series.



**Figure 2** Two views on trajectory tensor. The left is in terms of signals' trajectory matrices  $\{x_i(t)\}_{i=1}^m$ . The right is in terms of delay matrices

Similarly to the trajectory matrix introduce a *trajectory tensor* denoted by  $\mathbf{T}$ . It stacks delay matrices along the second index of the tensor, Fig. 2 (left). From the definition, it follows that each tensor slice along the second index is a trajectory matrix. Then, the  $\mathbf{T}$  is also the trajectory matrices  $H_{x_i}$  stacked along the third index, Fig. 2

(right). Therefore, the shared phase space of the time series is equal to a linear span of all  $H_{x_i}$ . As discussed in the Sec 3, each trajectory matrix also defines the individual phase space for the corresponding time series  $x_i(t)$ . Finally, we give

**Definition 1.** The multivariate time series are said to be *interdependent* if they share the common phase space with the same basis.

Assuming the observed time series interdependence, it follows that each  $H_{x_i}$  has a matrix factorization with the same set of factors. Since the trajectory matrices constructs the trajectory tensor, it has a low-dimensional representation. Using these two corollaries, we apply the CP-decomposition to the  $\mathbf{T}$  and obtain

$$\mathbf{T} = \sum_{i=1}^r \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i. \quad (1)$$

The CPD is defined by the tensor rank  $r$  and a set of vectors. The vectors are usually composed in full-rank matrices

$$A = [\mathbf{a}_1 \dots \mathbf{a}_r], B = [\mathbf{b}_1 \dots \mathbf{b}_r], C = [\mathbf{c}_1 \dots \mathbf{c}_r]. \quad (2)$$

Denote by  $\boldsymbol{\sigma}_{x_k}$  the  $k$ -th row of the  $C$ . Finally, with respect to the third-dimension slices of the  $\mathbf{T}$ , CPD goes as

$$\begin{cases} H_{x_1} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_1}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top, \\ H_{x_2} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_2}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top, \\ \dots \\ H_{x_m} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_m}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top. \end{cases} \quad (3)$$

The CPD exists for any tensor. In case of the trajectory tensor three properties must hold.

**Theorem 1.** Suppose that the given multivariate time series are interdependent. Then

1. Any  $\sigma_{x_k}$  is a non-zero vector.
2. The tensor rank  $r \leq L$ . It is equal to the dimension of the shared phase space.
3. Transposition of the trajectory matrix is a trajectory matrix. The shared phase space for the transposed matrices is the  $\text{Lin}(\{\mathbf{b}_i\})$ .

*Proof.* 1. Suppose some vector has a zero element at the position  $j$  and the other is non-zero in this position. Then two trajectory matrices factorize with a different set of factors. That violates the definition of the time series interdependence. The other case is that all  $\sigma_{x_k}$  have a zero element at the position  $j$ . Then the matrix  $C$  has a zero row and is not full-ranked. It contradicts to the CPD definition.

2. Since delay vectors are  $L$ -dimensional, each time series has its phase space dimension  $\leq L$ . All delay vectors of every time series also lie in the shared phase space  $\text{Lin}(\{\mathbf{a}_i\})$ . Therefore, dimensionality of the shared space can not be greater than  $L$ . Then, the set of  $\{\mathbf{a}_i\}_{i=1}^r$  vectors is linear-independent as follows from the CPD definition. So the dimension of the shared phase space equals  $r$ .
3. Rows of the trajectory matrices are the delay vectors of size  $N - L + 1$ . Transpose all equalities in (3). Now it is clear that rows of any  $H_{x_i}$  belongs to the  $\text{Lin}(\{\mathbf{b}_i\})$ . This is the shared phase space for the new delay vectors.

□

## 5 Time series decomposition

The time series are decomposed into additive components using the following conception: *factorization of the trajectory matrix  $H_{x_k}$  defines the decomposition of the related time series*. This idea comes from the SSA method. Since all  $H_{x_k}$  have a similar factorization, each time series is decomposed separately.

Trajectory matrices of the time series are *hankel* by their definition. That means equal elements along each anti-diagonal of the matrix. It is trivial that any time series of length  $N$  bijectionally corresponds to a hankel matrix of size  $L \times (N - L + 1)$ .

Describe the decomposition algorithm. First, the factors of the  $H_{x_k}$  are arbitrary partitioned into  $s$  groups. The  $s$  is also a chosen number. Then all factors within each partition are summed up. As a result we obtain matrices  $C_1, \dots, C_s$ . If they are all hankel then each  $C$  matrix corresponds to the time series component of the  $x_i(t)$ . However, it is almost infeasible even with the trivial time series (see the [11]). Therefore every  $C$  matrix is additionally *hankelized*. This operator averages each matrix's anti-diagonals so the matrix becomes hankel. Denote it as  $\text{Hankel}(\cdot)$ . Finally, the algorithm can be written as a chain of identical transformations

$$H_{x_k} = \quad (4)$$

$$\sum_{i=1}^r \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top = \quad (5)$$

$$\sum_{i \in \mathbb{I}_1} \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top + \dots + \sum_{i \in \mathbb{I}_s} \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top = \quad (6)$$

$$C_1 + \dots + C_s = \quad (7)$$

$$\text{Hankel}(C_1) + \dots + \text{Hankel}(C_s).$$

Final expression corresponds to the  $x_k(t) = c_1(t) + \dots + c_s(t)$ .

Here  $\mathbb{I}_1 \sqcup \dots \sqcup \mathbb{I}_s = \{1, \dots, r\}$  are chosen groups of indices.  $c_1(t), \dots, c_s(t)$  are the time series components extracted from the  $\text{Hankel}(C_1), \dots, \text{Hankel}(C_s)$  (see the second paragraph of this section). The (7) step requires a proof. But first, consider a hankel operator's property:

**Lemma 1.** *The  $\text{Hankel}(\cdot)$  is a linear operator.*

*Proof.* Having matrices  $A$  and  $B$  with the same size, take their elements from any anti-diagonal. Denote them by  $a_1, \dots, a_n$  and  $b_1, \dots, b_n$ . Then this anti-diagonal in  $\text{Hankel}(A + B)$  is

$$\frac{1}{n} \sum_{i=1}^n (a_i + b_i) = \frac{1}{n} \sum_{i=1}^n a_i + \frac{1}{n} \sum_{i=1}^n b_i.$$

It exactly results in a sum of the  $\text{Hankel}(A)$ 's and  $\text{Hankel}(B)$ 's anti-diagonals.

Secondly, having the scalar  $\alpha$ , it is trivial that

$$\frac{1}{n} \sum_{i=1}^n \alpha \cdot a_i = \alpha \frac{1}{n} \sum_{i=1}^n a_i.$$

Hence, we derive  $\text{Hankel}(\alpha A) = \alpha \text{Hankel}(A)$ .  $\square$

Now recall that the  $H_{x_k}$  is hankel. Combining it with the previous lemma, apply the  $\text{Hankel}(\cdot)$  to the  $H_{x_k} = C_1 + \dots + C_s$  from the algorithm. This proofs (7). Therefore we have shown the correctness of the decomposition algorithm.

## 6 Optimal decomposition problem

To eliminate ambiguity in the partitioning of the  $H_{x_k}$  factors, we restrict  $C_j$  matrices to be hankel. Thus, the (7) step of the decomposition algorithm becomes unnecessary. In addition, such partitioning completely fulfills the decomposition concept.

Formulate the decomposition problem as a system of matrix equations. Take (5) equality and apply the hankel operator to both sides. Recall that the  $H_{x_k}$  is hankel. Also take the very same equality as it is. We obtain

$$\begin{cases} H_{x_k} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top, \\ H_{x_k} = \sum_{i=1}^r \text{Hankel}(\boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top). \end{cases}$$

Then, subtract the second from the first and introduce

$$H_i = \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^T - \text{Hankel}(\boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^T) \quad (8)$$

It is called the *hankel residual matrix* of the  $i$ -th factor. Therefore we get

$$H_1 + \dots + H_r = 0. \quad \text{Or equally } H_r = - \sum_{j=1}^{r-1} H_j. \quad (9)$$

The condition on the  $C_i$  to be hankel is reformulated as the residual matrices sum up to zero within each partition group. The problem is to find the non-intersecting sets  $\mathbb{I}_1, \dots, \mathbb{I}_s$  such that

$$\sum_{k \in \mathbb{I}_i} H_k = 0, \quad i \in 1, \dots, s, \quad (10)$$

$$\mathbb{I}_1 \sqcup \dots \sqcup \mathbb{I}_s = \{1, \dots, r\}.$$

Without loss of generality, we set  $s$  equals two. Only  $\mathbb{I}_1, \mathbb{I}_2$  must be found. If ( 10 ) is solved in such setting, then  $\mathbb{I}_1$  and  $\mathbb{I}_2$  can be further partitioned separately. Hence, any number of the time series components can be obtained. Moreover, for any  $H_{x_k}$  its residual matrices are linearly dependent as follows from (9). Thus, the partition group for the  $H_r$  is uniquely defined if all other residuals are already partitioned. This matrix can be discarded. We also assume all  $H_j$  to be non-zero matrices. Otherwise, the zero residual matrix alone and all the rest make the desired partition on two groups. This assumption implies each of the groups to have at least two residual matrices. Finally, for any  $H_j$  matrix introduce an indicator-variable  $\beta_j \in \{0, 1\}$ . It shows what group the residual matrix belongs to. Denote by  $\boldsymbol{\beta}$  a vector  $[\beta_1 \dots \beta_{r-1}]^T$ . The decomposition

problem is to find  $\beta$  such that

$$\begin{cases} \sum_{j=1}^{r-1} \beta_j H_j = 0, \\ \beta_j \in \{0, 1\} \forall j \in 1 \dots r, \\ \sum_{i=1}^{r-1} \beta_i \geq 2. \end{cases} \quad (11)$$

The [11] shows that even for trivial time series such problem is infeasible. Now we transform it into a feasible one. First, vectorize each  $H_i$  and stack resulting vectors into the matrix denoted by  $\Lambda$ . Then the first equation in (11) is equivalent to the  $\Lambda\beta = 0$ . Second, change the problem of solving the system of equations to an optimization problem. Let's find such  $\beta$  that delivers a minimum to the norm of the  $\Lambda\beta$ . The changed problem is

$$\begin{cases} \|\Lambda\beta\| \rightarrow \min_{\beta}, \\ \beta_j \in \{0, 1\} \forall j \in 1 \dots r, \\ \sum_{i=1}^{r-1} \beta_i \geq 2. \end{cases} \quad (12)$$

Call solution of (12) an *optimal time series decomposition*. The problem itself is an Integer Least Squares (ILS) problem, which is proved to be NP-hard [16]. Therefore, *the optimal time series decomposition is an NP-hard problem*.

In spite of the complexity, methods and heuristics exist to reduce the ILS to computationally effective problems, see [17].

## 7 Time series forecasting

Using the build shared phase space each time series is forecast individually. We make a one-step forecast to estimate  $x(N+1)$ . Phase trajectory of the time series is a sequence of the delay vectors  $\{\overleftarrow{\mathbf{x}}_t\}$ . The whole trajectory belongs to the phase space  $\text{Lin}(\{\mathbf{a}_i\})$ . Continuation of the trajectory is the  $\overleftarrow{\mathbf{x}}_{N+1}$ . Therefore  $\overleftarrow{\mathbf{x}}_{N+1} \in \text{Lin}(\{\mathbf{a}_i\})$ . At the same

time  $x(N + 1)$  is the last component of the  $\overleftarrow{\mathbf{x}}_{N+1}$ . Now, recall the  $A$  matrix from (2).

The system of linear equation to find  $x(N + 1)$  is

$$\overleftarrow{\mathbf{x}}_{N+1} = A\boldsymbol{\lambda} \text{ or equally } \begin{cases} \mathbf{x} = \tilde{A}\boldsymbol{\lambda} \\ x(N + 1) = \mathbf{a}^\top \boldsymbol{\lambda} \end{cases}, \text{ where} \quad (13)$$

$$A = \begin{pmatrix} \tilde{A} \\ \mathbf{a}^\top \end{pmatrix}, \quad (14)$$

$$\overleftarrow{\mathbf{x}}_{N+1} = (\mathbf{x} \ x(N + 1))^\top. \quad (15)$$

The delay vector  $\overleftarrow{\mathbf{x}}_{N+1}$  and the matrix  $A$  are split into two blocks, see (14) and (15). The blocks are associated with the known and predicted part of the time series. The forecast comes from the second equation in (13). Therefore, the vector  $\boldsymbol{\lambda} \in \mathbb{R}^r$  is to be found from the first equation in (13). It is a system of linear equations with the matrix  $\tilde{A}$ . The matrix has a rank at least  $r - 1$  since  $A$  matrix is full-ranked. However, we assume  $\tilde{A}$  to have rank  $r$ , too. As experiments show this is not burdensome. In addition, the linear system is overdetermined since  $r \leq L$ , see Sec 4. Therefore its solution is given by the least-squares formula  $\boldsymbol{\lambda} = (\tilde{A}^\top \tilde{A})^{-1} \tilde{A}^\top \mathbf{x}$ . After replacing it in (13) we obtain the forecast

$$x(N + 1) = \mathbf{a}^\top (\tilde{A}^\top \tilde{A})^{-1} \tilde{A}^\top \mathbf{x}. \quad (16)$$

Denote by  $\mathbf{d}$  the vector  $\mathbf{a}^\top (\tilde{A}^\top \tilde{A})^{-1} \tilde{A}^\top$ . Once computed, it is used to make the forecast for further steps. Rewrite (16) in terms of  $x(t)$ :

$$x(t) = \sum_{i=1}^{L-1} d_i \cdot x(t - i)$$

This expression reveals the property of the forecast sequence  $\{x(N + i)\}$ .

**Theorem 2.** *The model of the tSSA's forecast is autoregressive AR( $L - 1$ ). The coefficients  $d_i$  found in (16), completely define behavior of the forecast sequence.*

## 8 Computational experiment

Analyze forecast and decomposition quality of the tSSA on sample multivariate time series. Also compare the tSSA with the other methods: the mSSA, the VAR, and the RNN. The RNN and VAR methods are able to make the forecast, but do not contain an explicit way of building time series decomposition.

To measure quality of the decomposition we introduce special metrics. They are motivated by the hankel residual matrices (8) introduced for the optimal decomposition problem, see Sec 5.

**Definition 2.** The *absolute hankel error* of the matrix  $M$  is

$$\text{AHE} = \|M - \text{Hankel}(M)\|_F.$$

**Definition 3.** The *relative hankel error* of the matrix  $M$  is

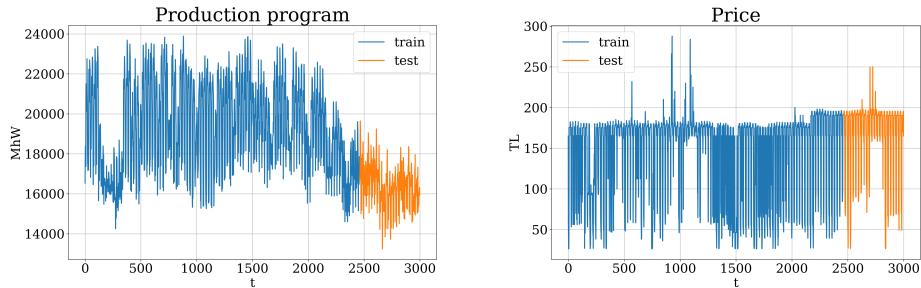
$$\text{RHE} = \frac{\text{AHE}}{\|M\|_F}.$$

The AHE is proportional to the total variance of matrix's anti-diagonals. The RHE is AHE's more interpretable modification. Recall the decomposition algorithm (5). It introduced the  $C_i$  matrices corresponded to the  $c_i(t)$  components of the time series  $x(t)$ . The (12) is the optimal decomposition problem. It is basically AHE minimizing for the  $C_i$  matrices. Hence, the AHE and the RHE are computed for the  $C_i$  in the further experiments. Denote by  $\overline{\text{RHE}}_{\text{ts}}$  the mean RHE across all  $C_i$  of the time series named "ts". Denote by  $\overline{\text{RHE}}$  the mean  $\overline{\text{RHE}}_{\text{ts}}$  across time series of this computational experiment.

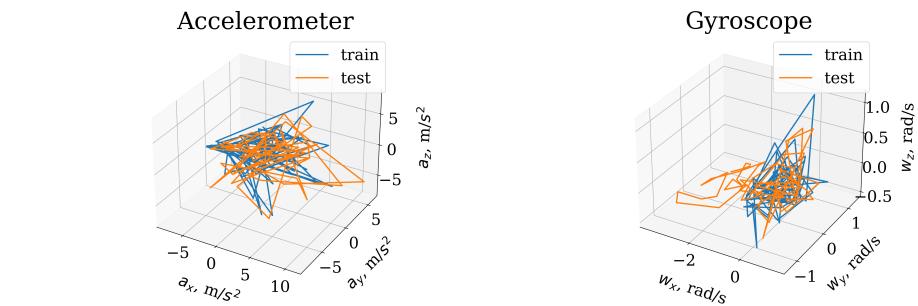
To measure forecast quality for the time series named "ts", use: the *mean squared error*  $\text{MSE}_{\text{ts}}$  and the *mean absolute percentage error*  $\text{MAPE}_{\text{ts}}$ . Denote by  $\overline{\text{MSE}}$

and  $\overline{\text{MAPE}}$  the mean  $\text{MSE}_{\text{ts}}$  and  $\text{MAPE}_{\text{ts}}$  across time series of this computational experiment.

Two sample multivariate time series are involved in the experiment. First, the electricity consumption and price, Fig. 3. Second, the inertial unit measurements [18]: three time series are from an accelerometer and three from a gyroscope, Fig. 4. The measurements represent walking movements. We assume the multivariate time series have shared phase space, see Def. 1. For the electricity time series it is related to the law of demand and supply. For the inertial unit time series it is related to coupled dynamical system describing the motion. The sample time series have the length  $3 \cdot 10^3$  time ticks. About 20% of them are deferred test samples for the forecast quality evaluation.



**Figure 3** Time series for the electricity consumption and price



**Figure 4** Time series for the inertial unit measurements

The tSSA's parameter  $L$  is set to 500 for the electricity time series and to 1000 for the inertial unit time series. This is also a parameter for the other models. For the RNN it is the minimum length of time series. For the VAR it is a maximal lag number. For the mSSA it is similar to the tSSA.

## 9 Data availability

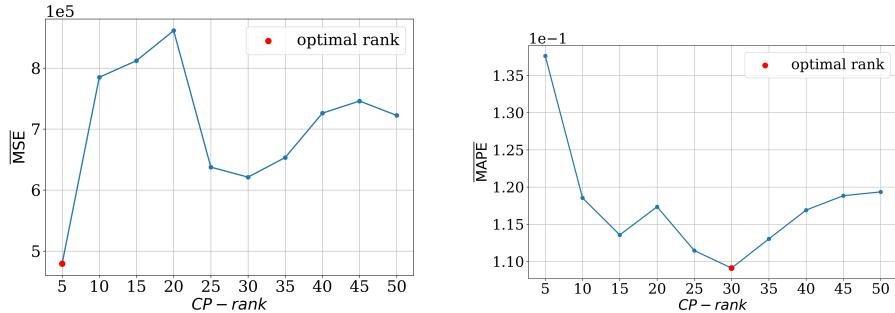
The electricity consumption data is available at <https://sourceforge.net/p/mvr/code/HEAD/tree/data/TurkElectricityConsumption.csv>. The inertial unit data is available at <https://data.mendeley.com/datasets/45f952y38r/5>.

## 10 Results and discussion

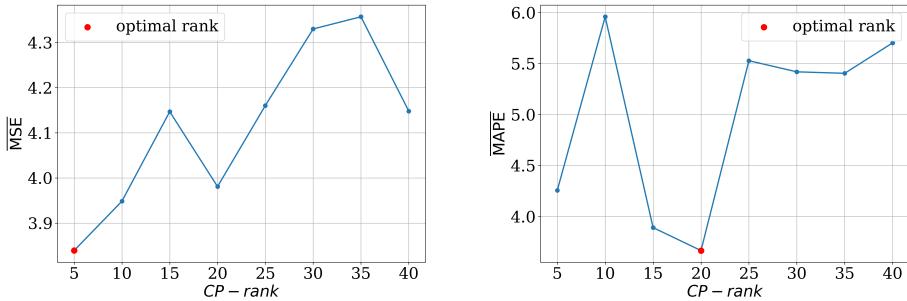
### *The forecast*

The trajectory tensor rank is equal to the forecast model's order, see Th. 2. Since efficient computation of tensor rank is impossible, the rank becomes tSSA's parameter. Fig. 5 and 6 show the forecast quality decreases with the greater ranks for the MAPE and MSE. Optimal rank value is 30. Nonetheless, the CPD computation error decreases monotonously with greater ranks, see Fig. 7. This error is an estimation of how close the computed CPD is to the true one (1). As a result, the small-order forecast model delivers the best quality.

Tab. 1 and 2 contain the MAPE and MSE metrics for all models. Our method showed the best results in most cases. Close values were obtained with the mSSA. The VAR model appeared unstable for the test samples. The RNN learned only the constant function from the electricity samples. But with the greater  $L$  for the inertial unit samples, the RNN performed better.



**Figure 5**  $\overline{\text{MSE}}$  and  $\overline{\text{MAPE}}$  metrics for the tSSA forecast depending on the CPD rank. An optimal point is marked with red. Electricity data



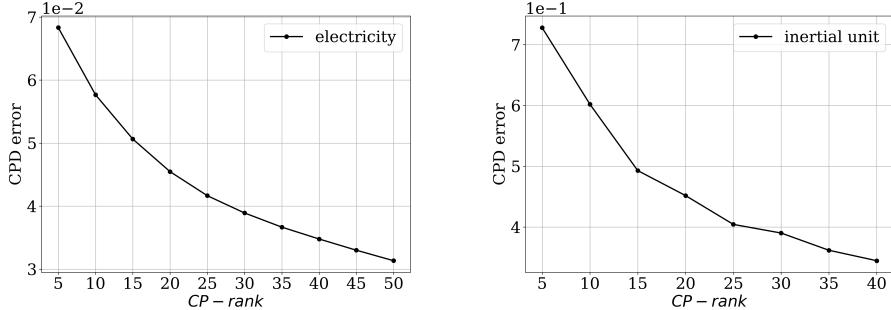
**Figure 6**  $\overline{\text{MSE}}$  and  $\overline{\text{MAPE}}$  metrics for the tSSA forecast depending on the CPD rank. An optimal point is marked with red. Inertial unit data

**Table 1** Forecast quality of the models on the electricity data

Metric \ Method	<i>tSSA</i>	<i>mSSA</i>	<i>VAR</i>	<i>RNN</i>
$\overline{\text{MSE}}_{\text{Production}}, 10^6$	1.24	1.51	7.81	2.70
$\overline{\text{MSE}}_{\text{Price}}, 10^3$	0.88	1.03	4.85	30.0
$\overline{\text{MSE}}, 10^6$	<b>0.62</b>	0.75	3.91	135.00
$\overline{\text{MAPE}}_{\text{Production}}$	0.054	0.060	0.137	0.999
$\overline{\text{MAPE}}_{\text{Price}}$	0.164	0.170	0.360	1.004
$\overline{\text{MAPE}}$	<b>0.109</b>	0.115	0.249	1.002

### The decomposition

Fig. 8 illustrates the RHE reaches a minimum for the tensor rank = 20. As the rank goes greater, the RHE does not change or even increase. The forecast quality had the same pattern. Therefore, small tensor ranks deliver the best quality for the



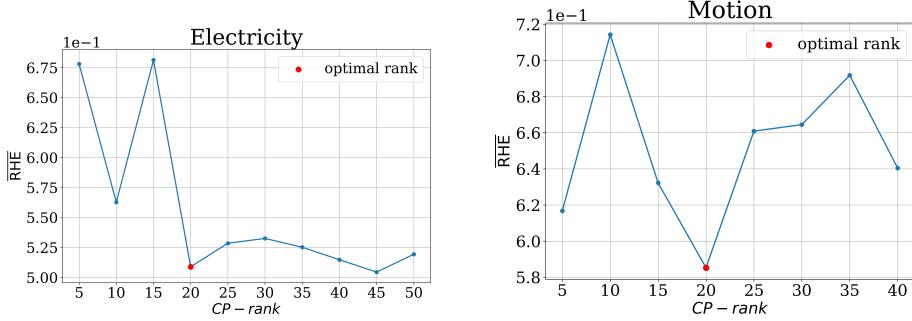
**Figure 7** Relative CPD computation error depending on the CPD rank. The left is for the electricity data, the right is for the inertial unit data

**Table 2** Forecast quality of the models on the inertial unit data

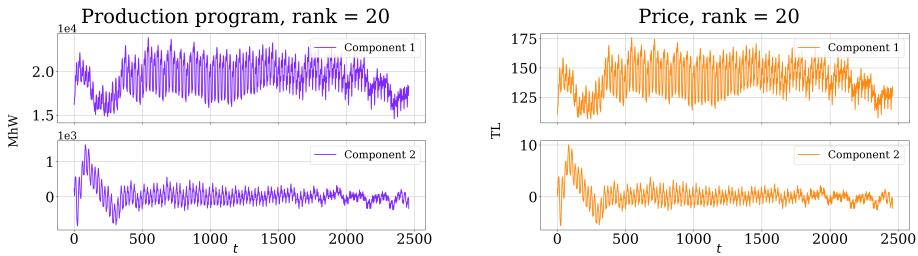
Metric \ Method	<i>tSSA</i>	<i>mSSA</i>	<i>VAR</i>	<i>RNN</i>
$\overline{\text{MSE}}_{\text{Accel}}$	7.351	6.980	8.108	6.604
$\overline{\text{MSE}}_{\text{Gyro}}$	0.610	0.636	0.631	0.639
$\overline{\text{MSE}}$	3.981	3.808	4.370	<b>3.622</b>
$\overline{\text{MAPE}}_{\text{Accel}}$	3.558	3.516	3.370	1.747
$\overline{\text{MAPE}}_{\text{Gyro}}$	3.773	3.943	10.427	5.641
$\overline{\text{MAPE}}$	<b>3.666</b>	3.730	6.899	3.694

decomposition and forecast problems. At the same time, the rank is a value of the shared phase space dimension, see Th. 1. The delay vectors define phase trajectory and have dimension equals  $L$ . Recall that  $L$  is set to 500 for the electricity samples and to 1000 for the inertial unit samples. As a result, the tSSA has significantly reduced dimensionality of the sample time series.

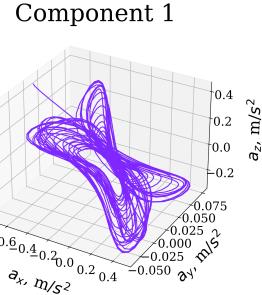
Fig. 9 illustrates the decomposition on two components for the electricity consumption and price. The components are almost identical to the bias and scale factors. This is attributed to the *shared* phase space between the time series. Fig. 10 shows two-component decomposition for the accelerometer and Fig. 11 shows the same for the gyroscope. The second components are identical to the rotation and scale. Moreover, all components here coincide with the phase trajectories.



**Figure 8** The  $\overline{\text{RHE}}$  metrics depending on the CPD rank. The left is for the electricity data, the right is for the inertial unit data. An optimal point is marked with red

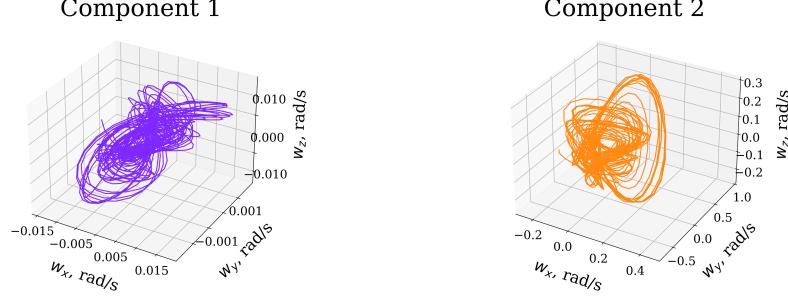


**Figure 9** tSSA decomposition for the electricity data. CPD rank = 20



**Figure 10** tSSA decomposition for the accelerometer data. CPD rank = 10

Tab. 3 and 4 shows the tSSA have lower decomposition quality than the mSSA. It is attributed to the two factors. First, the mSSA decomposition is connected with the factors partitioning like the tSSA. Several heuristics exist to obtain usually high-RHE components. On the other hand, the tSSA optimal decomposition problem (12) is the Integer Least Squares (ILS) problem, see the end of Sec. 6. It can not be solved



**Figure 11** tSSA decomposition for the gyroscope data. CPD rank = 10

efficiently, only approximate methods are available. Moreover, the matrix  $\Lambda$  involved in the ILS has a large row dimension. To shorten computation time, only several hundred rows were left. It is still much greater than the size of the sought parameter vector in the ILS. But such approximation is not equal to the initial problem.

**Table 3** Decomposition quality of the models for the electricity data

Metric \ Method	tSSA	mSSA
$\overline{\text{RHE}}_{\text{Production}}$	0.507	0.308
$\overline{\text{RHE}}_{\text{Price}}$	0.511	0.310
$\overline{\text{RHE}}$	0.509	<b>0.309</b>

**Table 4** Decomposition quality of the models for the inertial unit data

Metric \ Method	tSSA	mSSA
$\overline{\text{RHE}}_{\text{Accel}}$	0.438	0.394
$\overline{\text{RHE}}_{\text{Gyro}}$	0.732	0.468
$\overline{\text{RHE}}$	0.585	<b>0.431</b>

## 11 Conclusion

The tSSA method is devoted to the forecast and the additive decomposition of the interdependent multivariate time series. It is based on dynamical systems theory and

explicitly build the shared phase space of the time series. It also exploits multilinearity of the data via tensor data representation. The method has only two adjustable parameters and require the CPD computation of the trajectory tensor. The tSSA forecast is vector's scalar product. The tSSA decomposition is matrix factorization and factor's partitioning. Finally, the computational experiment showed tSSA's high forecast quality. For the electricity data, the tSSA has the best forecast. The MSE metric is 21% less than the second best forecast. The MAPE metric is 35% less than the other methods on average. The tSSA also reduced dimensionality of the electricity data in 16 times.

Future research will be devoted to a new way of building the shared phase space. Then, the current decomposition conception results in a computationally complex problem. Finding new concepts to build additive components is the other direction for the method's improvement.

## References

- [1] Enders, W.: Applied Econometric Time Series. Wiley series in probability and statistics. Wiley, New York (2010)
- [2] Dagum, E., Bianconcini, S.: Seasonal Adjustment Methods and Real Time Trend-Cycle Estimation, p. 283. Springer, Switzerland (2016)
- [3] Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: Stl: A seasonal-trend decomposition procedure based on loess (with discussion). *Journal of Official Statistics* **6**, 3–73 (1990)
- [4] Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day series in time series analysis and digital processing. Holden-Day, Minnesota (1970)

- [5] Hyndman, R.J., Athanasopoulos, G.: *Forecasting: Principles and Practice*. OTexts, Australia (2021)
- [6] Volterra, V., Brelot, M.: *Leçons sur la Théorie Mathématique de la Lutte pour la vie*. Cahiers Scientifiques. Fasc Series. Gauthier-Villars et cie., Michigan (1931)
- [7] Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Computation* **9**(8), 1735–1780 (1997) <https://doi.org/10.1162/neco.1997.9.8.1735>
- [8] Tealab, A.: Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal* **3**(2), 334–340 (2018) <https://doi.org/10.1016/j.fcij.2018.10.003>
- [9] Asteriou, D., Hall, S.: Vector Autoregressive (VAR) Models and Causality Tests, pp. 333–346 (2016). [https://doi.org/10.1057/978-1-37-41547-9\\_15](https://doi.org/10.1057/978-1-37-41547-9_15)
- [10] Zellner, A.: An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American Statistical Association* **57**(298), 348–368 (1962) <https://doi.org/10.1080/01621459.1962.10480664>
- [11] Stepanov, D., Golyandina, N.: SSA-based approaches to analysis and forecast of multidimensional time series, pp. 293–298 (2005)
- [12] Takens, F.: Detecting Strange Attractors in Turbulence. In: *Dynamical Systems and Turbulence*, Warwick 1980. Lecture Notes in Mathematics, vol. 898, pp. 366–381. Springer, Berlin (1981). Chap. 21. <https://doi.org/10.1007/bfb0091924>
- [13] Krim, H., Viberg, M.: Two decades of array signal processing research: The parametric approach. *Signal Processing Magazine, IEEE* **13**, 67–94 (1996) <https://doi.org/10.1109/79.526899>
- [14] Ignatov, A.D., Strijov, V.V.: Human activity recognition using quasiperiodic

- time series collected from a single tri-axial accelerometer. *Multimedia tools and applications* **75**, 7257–7270 (2016) <https://doi.org/10.1007/s11042-015-2643-0>
- [15] Hassani, H., Mahmoudvand, R.: Multivariate singular spectrum analysis: A general view and new vector forecasting approach. *International Journal of Energy and Statistics* **01** (2013) <https://doi.org/10.1142/S2335680413500051>
- [16] Emde-Boas, P.: Another NP-complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice. Report. Department of Mathematics. University of Amsterdam. Department, Univ., Amsterdam (1981)
- [17] Grafarend, E., Zwanzig, S., Awange, J.: Integer Least Squares, pp. 579–603. Springer, Cham (2022)
- [18] Nahid, A.-A., Sikder, N., Rafi, I.: KU-HAR: An Open Dataset for Human Activity Recognition. Mendeley Data (2021). <https://doi.org/10.17632/45f952y38r.5>
- [19] Håstad, J.: Tensor rank is np-complete. *Journal of Algorithms* **11**(4), 644–654 (1990) [https://doi.org/10.1016/0196-6774\(90\)90014-6](https://doi.org/10.1016/0196-6774(90)90014-6)
- [20] Kolda, T., Bader, B.: Tensor decompositions and applications. *SIAM Review* **51**, 455–500 (2009) <https://doi.org/10.1137/07070111X>